ALGEBRAS OF PROCESSES OF TIMED PETRI NETS [1]

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

Abstract

Processes of timed Petri nets are represented by labelled partial orders with some extra features. These features reflect the execution times of processes and allow to combine processes sequentially and in parallel. The processes can be represented either without specifying when particular situations appear (free time-consuming processes), or together with the respective appearance times (timed time-consuming processes). The processes of the latter type determine the possible firing sequences of the respective nets.

Key words: concatenable weighted pomset, symmetry, table, interchange, sequential composition, parallel composition, monoidal category, timed Petri net, process, free time-consuming process, timed time-consuming process, delay table, firing sequence.

1 Motivation and introduction

Petri nets are a widely accepted model of concurrent systems. Originally they were invented for modelling those aspects of system behaviours which can be expressed in terms of causality and choice. Recently a growing interest can be observed in modelling real-time systems, which implies a need of a representation of the lapse of time. To meet this need various solutions has been proposed known as timed Petri nets.

For the usual Petri nets there exist precise characterisations of behaviours. Among them occurrence nets as described in [Wns 87] and in [MMS 92], and algebras of processes as in [DMM 89] seem to be most adequate.

In the case of timed Petri nets the situation is less advanced since the existing semantics either do not reflect properly concurrency (cf. [GMMP 89] for a review) or they oversimplify the representation of the lapse of time (as in [BG 92]). Besides, the presence of the concept of time in the model gives rise to a variety of problems as those of performance evaluation, and this creates a need of new formal tools.

In this paper we try to build the needed tools by representing the behaviour of a timed net by an algebra of structures called concatenable weighted pomsets. These structures correspond to concatenable processes of [DMM 89] with some extra information about the lapse of time. If the lapse of time is represented only in terms of delays between situations then we call such structures free time-consuming processes. If also the time instants at which situations arise are given then we call them timed time-consuming processes.

[1] This work has been supported partially by KBN - the Polish State Committee for Scientific Research (Grant No. 2 2047 92 03). Its version has been presented at CONCUR'94 and published in vol. 836 of Springer Lecture Notes in Computer Science.

There are natural homomorphisms from the algebra of timed time-consuming processes of a net to the algebra of its free time-consuming processes, and from the algebra of free time-consuming processes to an algebra whose elements reflect how much time the respective processes take. More precisely, to each free time-consuming process there corresponds a table of least possible delays between its data and results (a delay table) such that the tables corresponding to the results of operations on processes (a sequential and a parallel composition) can be obtained by composing properly the tables corresponding to components.

The delay tables which correspond to processes generalize in a sense the concept of execution time. The representation of execution time by a number is not adequate enough when we have to do with processes consisting of independent components. For example, for a process $\alpha;\beta \otimes \gamma;\delta$, where $\alpha;\beta$ and $\gamma;\delta$ are independent components, $\alpha;\beta$ stands for α followed by β, and $\gamma;\delta$ stands for γ followed by δ, the execution time cannot be represented by a number since it may vary depending on when the components $\alpha;\beta$ and $\gamma;\delta$ start. At the same time, the table of least possible delays between causally related initial and terminal situations is unique.

We want the items of delay tables to represent the least possible delays between the respective situations. This corresponds to executing the respective processes as fast as possible.

An important property of free time-consuming processes and their delay tables is that they do not depend on when the respective data appear. Due to this property one can compute how a process of this type proceeds in time for any given combination of appearance times of its data. The combination which is given plays here the role of a marking. This marking is timed in the sense that not only the presences of tokens in places are presented, but also the respective appearance times which need not be the same.

The possibility of computing how a free time-consuming process applies to a given timed marking allows us to find the corresponding timed process. The possibility of finding timed processes of a net allows us to see which of them can be chosen and to define the possible firing sequences of the net.

The present paper exploits some ideas of [Wi 80] and [Wi 92]. It is an improved version of [Wi 93]. Results are presented in it without proofs. The respective proofs will be given in a more complete paper.

2 Concatenable weighted pomsets

Processes of timed nets will be represented by partially ordered multisets (pomsets in the terminology of [Pra 86]) with some extra arrangements of minimal and maximal elements (similar to those in concatenable processes of [DMM 89]), and with some extra features (weights) and properties.

Let V be a set of labels.

2.1. Definition. A *concatenable weighted labelled partial order* (or a *cwlp-order*) over V is $\mathcal{A} = (X, \leq, d, e, s, t)$, where:

(1) $(X, \leq)$ is a finite partially ordered set with a subset X_{min} of minimal elements and a subset X_{max} of maximal elements such that each maximal chain has an element in each maximal antichain,

(2) $d : X \times X \to \{-\infty\} \cup [0, +\infty)$ is a *weight function* such that $d(x, y) = -\infty$ iff $x \leq y$ does not hold, $d(x, x) = 0$, and $d(x, y)$ is the maximum of sums $d(x, x_1)+...+d(x_n, y)$ over all maximal chains $x \leq x_1 \leq ... \leq x_n \leq y$ from x to y whenever $x \leq y$,

(3) $e : X \to V$ is a *labelling function*,

(4) $s = (s(v) : v \in V)$ is a family of enumerations of the sets $e^{-1}(v) \cap X_{min}$ (an *arrangement of minimal elements*),

(5) $t = (t(v) : v \in V)$ is a family of enumerations of the sets $e^{-1} \cap X_{max}$ (an *arrangement of maximal elements*). □

By an enumeration of a set we mean here a sequence of elements of this set in which each element of this set occurs exactly once.

The property of the order in (1) is known in the theory of Petri nets as K-density. It is assumed in order to guarantee that each maximal antichain of a cwlp-order $\mathcal{A}$ defines a decomposition of $\mathcal{A}$ into components such that $\mathcal{A}$ could uniquely be reconstructed from these components (cf. 2.5). The property (2) implies that for $x \leq y$ there exists a maximal chain $x \leq x_1 \leq ... \leq x_n \leq y$ from x to y which is *critical* in the sense that $d(x, y) = d(x, x_1) + ... + d(x_n, y)$. Together with (1) it implies also that the weight function is determined uniquely by specifying its values for x, y such that y is an immediate successor of x.

The interpretation of a cwlp-order as a representation of a process will be given in section 4.

2.2. Definition. Two cwlp-orders $\mathcal{A} = (X, \leq, d, e, s, t)$ and $\mathcal{A}' = (X', \leq', d', e', s', t')$ are said to be *isomorphic* if there exists an isomorphism from $\mathcal{A}$ to $\mathcal{A}'$, that is a bijection $b : X \to X'$ such that $x \leq y$ iff $b(x) \leq' b(y)$, $d'(b(x), b(y)) = d(x, y)$, $e'(b(x)) = e(x)$, $s'(v) = b(s(v))$, and $t'(v) = b(t(v))$, where $b(x_1...x_n)$ denotes $b(x_1)...b(x_n)$, for all $x, y \in X$ and $v \in V$. □

2.3. Definition. A *concatenable weighted pomset* (or a *cw-pomset*) over V is an isomorphism class of cwlp-orders over V. □

The cw-pomset which is the isomorphism class of a given cwlp-order $\mathcal{A} = (X, \leq, d, e, s, t)$ is written as $[\mathcal{A}]$ and $\mathcal{A}$ is called its *instance*. The restriction of $\mathcal{A}$ to X_{min} with t replaced by s and that to X_{max} with s replaced by t are instances of cw-pomsets written respectively as $\partial_0([\mathcal{A}])$ and $\partial_1([\mathcal{A}])$ and called the *source* and the *target* of $[\mathcal{A}]$. If $X = X_{min} = X_{max}$ then $\leq$ reduces to the identity and we call $[\mathcal{A}]$ a *symmetry*. If also $t = s$ then $[\mathcal{A}] = \partial_0([\mathcal{A}]) = \partial_1([\mathcal{A}])$ and $[\mathcal{A}]$ becomes a *trivial symmetry* or, equivalently, a *multiset* of elements of V with the multiplicity of each $v \in V$ given by $cardinality(e^{-1}(v) \cap X)$. By $cwpomsets(V)$, $symmetries(V)$, $trivsym(V)$ we denote respectively the set of cw-pomsets, the set of symmetries, and the set of trivial symmetries over V.

Examples of cw-pomsets of which one is a symmetry are shown in figure 1. In the graphical representation we omit arrows which follow from the transitivity of order and the respective weights which follow from the assumed properties of weight function. The arrangements of minimal and maximal elements are shown by endowing the labels of minimal elements with subscripts and the labels of maximal elements with superscripts which indicate the positions of the considered elements in the respective sequences.

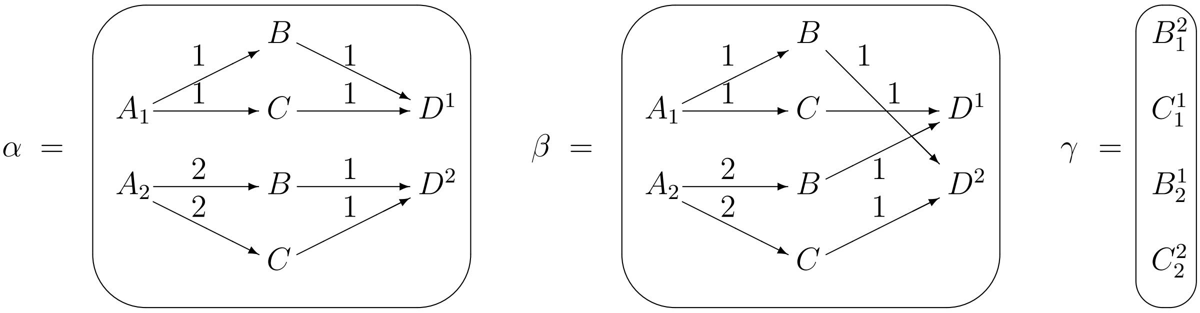

Figure 1

To each permutation p of a sequence $1, ..., n$ there corresponds an operation I_p, called after [DMM 89] an *interchange*, which to each n-tuple $(a_1, ..., a_n)$ of trivial symmetries $a_i = [(X_i, \leq_i, d_i, e_i, s_i, t_i)]$ assigns the symmetry $I_p = [(X, \leq, d, e, s, t)]$, where X is a disjoint union of (suitable copies of) $X_1, ...X_n$, $\leq$ is the identity on X, $e(x) = e_i(x)$ for $x \in X_i$, $s(v) = s_1(v)...s_n(v)$ (the concatenation of $s_1(v), ..., s_n(v)$), and $t(v) = s_{p(1)}(v)...s_{p(n)}(n)$ (the concatenation of $s_{p(1)}(v) = t_{p(1)}(v), ..., s_{p(n)}(v) = t_{p(n)}(v)$). By $*$ and I_* we denote respectively the permutation $1 \mapsto 2, 2 \mapsto 1$ and the corresponding interchange.

Let $\mathcal{A} = (X, \leq, d, e, s, t)$ be a cwlp-order and let $cuts(\mathcal{A})$ denote the set of all maximal antichains of $\mathcal{A}$. Each maximal antichain $Y \in cuts(\mathcal{A})$ defines the subsets

$$\downarrow Y = \{x \in X : x \leq y \text{ for some } y \in Y\} \quad \text{and} \quad \uparrow Y = \{x \in X : y \leq x \text{ for some } y \in Y\}.$$

Given two maximal antichains Y and Y' in $cuts(\mathcal{A})$, we write $Y \sqsubseteq Y'$ if $\downarrow Y \subseteq \downarrow Y'$.

2.4. Proposition.; The relation $\sqsubseteq$ is a partial order on the set $cuts(\mathcal{A})$ such that $cuts(\mathcal{A})$ with this order is a lattice. $\square$

2.5. Proposition. For each $Y \in cuts(\mathcal{A})$ the order $\leq$ is the transitive closure of the union of its restrictions to the subsets $\downarrow Y$ and $\uparrow Y$. $\square$

2.6. Proposition. For each $Y \in cuts(\mathcal{A})$ and for all $x \in \downarrow Y$ and $y \in \uparrow Y$ the weight $d(x, y)$ is given by the formula $d(x, y) = max(d(x, z) + d(z, y) : z \in Y)$. $\square$

2.7. Proposition. For each $Y \in cuts(\mathcal{A})$ the restrictions of $\mathcal{A}$ to $\downarrow Y$ and $\uparrow Y$ with a family $r = (r(v) : v \in V)$ of enumerations of the sets $e^{-1}(v) \cap Y$ playing the role of arrangement of maximal elements of $\downarrow Y$ and of arrangement of minimal elements of $\uparrow Y$ are cwlp-orders, written respectively as $head_{Y,r}(\mathcal{A})$ and $tail_{Y,r}(\mathcal{A})$. $\square$

The cw-pomset $[\mathcal{A}]$ is said to consist of the cw-pomset $[head_{Y,r}(\mathcal{A})]$ followed by the cw-pomset $[tail_{Y,r}(\mathcal{A})]$.

2.8. Proposition. For every two cw-pomsets α and β with $\partial_0(\beta) = \partial_1(\alpha)$ there exists a unique cw-pomset $\alpha;\beta$ which consists of α followed by β. This cw-pomset is a symmetry whenever α and β are symmetries. $\square$

The operation $(\alpha, \beta) \mapsto \alpha;\beta$ is called the *sequential composition* of cw-pomsets. Examples of application of this operation are shown in figures 2 and 3.

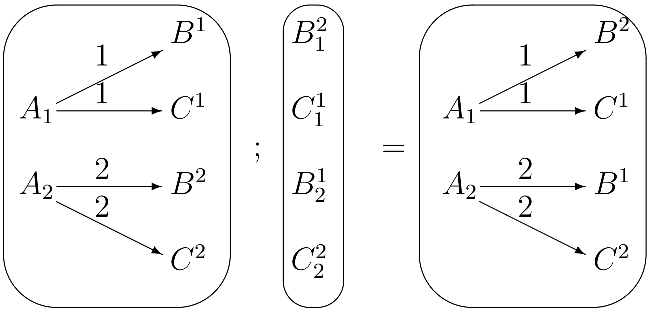

Figure 2

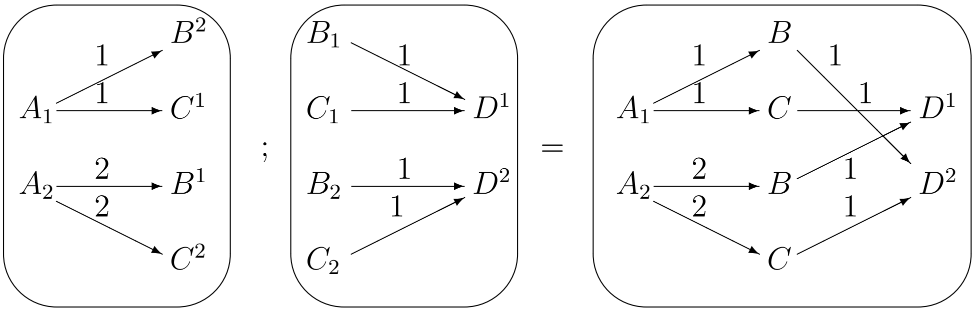

Figure 3

2.9. Proposition. The sequential composition is defined for all pairs (α, β) of cw-pomsets with $\partial_0(\beta) = \partial_1(\alpha)$, it is associative and such that $\partial_0(\alpha;\beta) = \partial_0(\alpha)$, $\partial_1(\alpha;\beta) = \partial_1(\beta)$, and $\partial_0(\alpha);\alpha = \alpha;\partial_1(\alpha) = \alpha$ for all cw-pomsets α,β. $\square$

Another operation on cw-pomsets can be introduced with the aid of *splittings*, where a splitting of a cwlp-order $\mathcal{A} = (X, \leq, d, e, s, t)$ is a partition $p = (X', X'')$ of X into two disjoint subsets X', X'' which are *independent* in the sense that x', x'' are incomparable whenever $x' \in X'$ and $x'' \in X''$, each $s(v)$ is $(s(v)|X')(s(v)|X'')$, the concatenation of the restrictions of $s(v)$ to X' and X'', and each $t(v)$ is $(t(v)|X')(t(v)|X'')$, the concatenation of the restrictions of $t(v)$ to X' and X''.

Let $\mathcal{A} = (X, \leq, d, e, s, t)$ be a cwlp-order and let $splittings(\mathcal{A})$ denote the set of splittings of $\mathcal{A}$.

2.10. Proposition. For each $p = (X', X'') \in splittings(\mathcal{A})$ the restrictions of $\mathcal{A}$ to X' and X'' with arrangements of minimal elements given respectively by $s|X' = (s(v)|X' : v \in V)$ and $s|X'' = (s(v)|X'' : v \in V)$, and arrangements of maximal elements given respectively by $t|X' = (t(v)|X' : v \in V)$ and $t|X'' = (t(v)|X'' : v \in V)$, are cwlp-orders, written respectively as $left_p(\mathcal{A})$ and $right_p(\mathcal{A})$. $\square$

The cw-pomset $[\mathcal{A}]$ is said to consist of the cw-pomset $[left_p(\mathcal{A})]$ accompanied by the cw-pomset $[right_p(\mathcal{A})]$.

2.11. Proposition. For every two cw-pomsets α and β there exists a unique cw-pomset $\alpha \otimes \beta$ which consists of α accompanied by β. This cw-pomset is a symmetry whenever α and β are symmetries. $\square$

The operation $(\alpha, \beta) \mapsto \alpha \otimes \beta$ is called the *parallel composition* of cw-pomsets. An example of application of this operation is shown in figure 4.

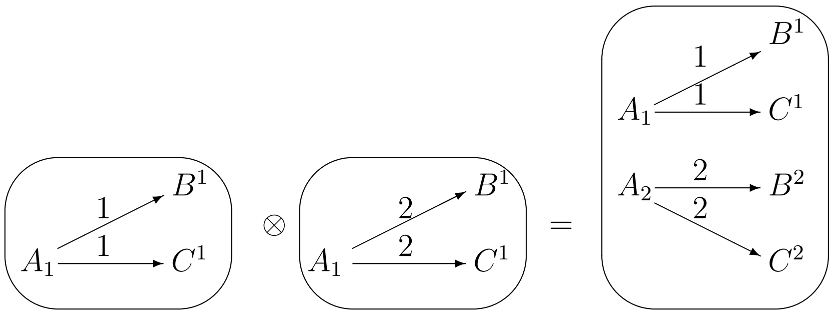

Figure 4

2.12. Proposition. The parallel composition is defined for all pairs (α, β) of cw-pomsets, it is associative, and has a neutral element *nil*, where *nil* is the unique cw-pomset with the empty instance. $\square$

2.13. Proposition. The parallel composition is *functorial* in the sense that

$$\alpha; \beta \otimes \gamma; \delta = (\alpha \otimes \gamma); (\beta \otimes \delta)$$

whenever $\alpha; \beta$ and $\gamma; \delta$ are defined. $\square$

2.14. Proposition. The parallel composition is *coherent* in the sense that

$$I_p(u_1, ..., u_n); \alpha_{p(1)} \otimes ... \otimes \alpha_{p(n)} = \alpha_1 \otimes ... \otimes \alpha_n; I_p(v_1, ..., v_n)$$

for all $\alpha_1, ..., \alpha_n \in cwpomsets(V)$ with $\partial_0(\alpha_i) = u_i$ and $\partial_1(\alpha_i) = v_i$, and for each permutation p of the sequence $1, ..., n$. $\square$

2.15. Proposition. The subset of symmetries is closed w.r. to the compositions and interchanges. $\square$

2.16. Theorem. The structure

$$CWPOMSETS(V) = (cwpomsets(V), \partial_0, \partial_1, ;, \otimes, nil, I_*)$$

is a symmetric strict monoidal category (the *monoidal category of cw-pomsets* over V) with cw-pomsets playing the role of morphisms, trivial symmetries playing the role of objects, and I_* playing the role of a natural transformation from $(\alpha, \beta) \mapsto \alpha \otimes \beta$ to $(\alpha, \beta) \mapsto \beta \otimes \alpha$. It contains $SYMMETRIES(V)$, the subcategory of symmetries with the members of $symmetries(V)$ playing the role of morphisms. $\square$

2.17. Proposition. Each cw-pomset α can be obtained with the aid of interchanges and compositions from *atomic* cw-pomsets of the following two types:

(1) *one-element* cw-pomsets, one for each $v \in V$, written also as v, namely the one-element cw-pomset with v being the label of the only element of its instance,

(2) *prime* cw-pomsets $\pi = [\mathcal{P}]$ for some $\mathcal{P} = (X_\mathcal{P}, \leq_\mathcal{P}, d_\mathcal{P}, e_\mathcal{P}, s_\mathcal{P}, t_\mathcal{P})$ such that $X_\mathcal{P} = (X_\mathcal{P})_{min} \cup (X_\mathcal{P})_{max}$, where $(X_\mathcal{P})_{min}$ and $(X_\mathcal{P})_{max}$ are nonempty and disjoint and each element of $(X_\mathcal{P})_{min}$ is comparable with each element of $(X_\mathcal{P})_{max}$.

In order to obtain α one needs always the same number $|\alpha|(\pi)$ of copies of each prime cw-pomset π. Each atomic cw-pomset ϱ is *symmetrical* in the sense that $\sigma; \varrho = \varrho; \sigma' = \varrho$ for all symmetries σ, σ' such that the respective compositions are defined. $\square$

By $|\alpha|$ we denote the multiset of prime processes which are needed to construct a cw-pomset α. By $atomic(V)$, $oneelement(V)$, $prime(V)$ we denote respectively the set of atomic, one-element, and prime cw-pomsets over V. For each subset P of cw-pomsets over V by $closure(P)$ we denote the least subset of $cwpomsets(V)$ that contains P and is closed w.r. to interchanges and compositions.

2.18. Theorem. The monoidal category $CWPOMSETS(V)$ and its subcategory $SYMMETRIES(V)$ are generated respectively by the set $atomic(V)$ of symmetrical atomic cw-pomsets and the subset $oneelement(V)$ of symmetrical one-element cw-pomsets in the sense that

$$cwpomsets(V) = closure(atomic(V))$$

$$symmetries(V) = closure(oneelement(V)) \quad \square$$

Following the line of [DMM 89] it is possible to show that the properties formulated here of the monoidal category of cw-pomsets characterize this category up to isomorphism.

3 Tables

Tables of delays between situations of processes (delay tables) are matrix-like objects with a special indexing of rows and columns which depends on a set of labels.

3.1. Definition. A *table* over a set V of labels is a triple $F = (I, J, f)$, where $I, J : V \to \{0, 1, ...\}$ and f is a function which assigns a weight $f(m, n) \in \{-\infty\} \cup [0, +\infty)$ to each pair of indices m, n such that $m = (u, i)$ and $n = (v, j)$ with $u, v \in V$, $1 \leq i \leq I(u)$ and $1 \leq j \leq J(v)$. $\square$

Functions I, J can be regarded as multisets of elements of V. For each table $F = (I, J, f)$ we have two tables $\partial_0'(F) = (I, I, \delta(I))$ and $\partial_1'(F) = (J, J, \delta(J))$, where $\delta(K)$ denotes the function with $\delta(K)(m, n) = 0$ for $n = m$ and $\delta(K)(m, n) = -\infty$ otherwise. If $J = I$ and there exist permutations φ_v of $1, ..., I(v)$ such that $f(m, n) = 0$ for $m = (u, i)$ and $n = (v, j)$ with $v = u$ and $j = \varphi_u(i))$, and $f(m, n) = -\infty$ for the remaining m, n, then we call F a *table symmetry*. In particular, if all φ_v are identities then we say that F is a *trivial* table symmetry and identify it with the multiset with the multiplicity $I(v) = J(v)$ of each $v \in V$. For $I = J = 0$ we have the *empty* table *nil'*. By *tables*(V), *tsymmetries*(V), and *trivtsym*(V) we denote respectively the set of tables, the set of table symmetries, and the set of trivial table symmetries over V. An example of a table is shown in figure 5. This table corresponds to the tc-process α in figure 1.

$$
tab(\alpha) \quad = \qquad
\begin{array}{c|cc}
 & (D,1) & (D,2) \\
\hline
(A,1) & 2 & -\infty \\
(A,2) & -\infty & 3
\end{array}
$$

Figure 5

For each permutation p of $1, ..., n$ we have an *interchange* I_p' which assigns to each n-tuple $(F_1, ..., F_n)$ of trivial symmetries $F_k = (I_k, I_k, \delta(I_k))$ a symmetry $F = (I, I, f)$, where $I(v) = I_1(v) + ... + I_n(v)$ and $f(m, n) = -\infty$ except for $m = (u, i)$ and $n = (u, p(i))$, where $f(m, n) = 0$. In particular, for $*$ denoting the permutation $1 \mapsto 2, 2 \mapsto 1$ we have the interchange I_*'.

For each pair (F, F') of tables $F = (I, J, f)$ and $F' = (I', J', f')$ such that $\partial_0'(F') = \partial_1'(F)$ we have the unique table $F;' F' = (I, J', g)$ with $g(m, n)$ denoting the maximum of the sums $f(m, k) + f'(k, n)$ over all $k = (v, j)$ with $v \in V$ and $1 \leq j \leq J(v)$.

The operation $(F, F') \mapsto F;' F'$ is called the *sequential composition* of tables. An example of application of this operation is shown in figure 6.

$$
\begin{array}{c|cccc}
 & (B,1) & (B,2) & (C,1) & (C,2) \\
\hline
(A,1) & 1 & -\infty & 1 & -\infty \\
(A,2) & -\infty & 2 & -\infty & 2
\end{array}
\;;'\;
\begin{array}{c|cc}
 & (D,1) & (D,2) \\
\hline
(B,1) & 1 & -\infty \\
(B,2) & -\infty & 1 \\
(C,1) & 1 & -\infty \\
(C,2) & -\infty & 1
\end{array}
\;=\;
\begin{array}{c|cc}
 & (D,1) & (D,2) \\
\hline
(A,1) & 2 & -\infty \\
(A,2) & -\infty & 3
\end{array}
$$

Figure 6

For each pair (F, F') of tables $F = (I, J, f)$ and $F' = (I', J', f')$ we have the unique table $F \otimes' F' = (I + I', J + J', h)$ with $h(m, n) = f(m, n)$ for $m = (u, i)$ and $n = (v, j)$ such that $u, v \in V$ and $1 \leq i \leq I(v)$ and $1 \leq j \leq J(v)$, $h(m,n) = f'(m',n')$ for $m = (v, i+I(v))$ and $n = (v, j+J(v))$ and $m' = (v,i)$ and $n' = (v,j)$ such that $u, v \in V$ and $1 \leq i \leq I'(v)$ and $1 \leq j \leq J'(v)$, and $h(m,n) = -\infty$ for the remaining m, n.

The operation $(F, F') \mapsto F \otimes' F'$ is called the *parallel composition* of tables. An example of application of this operation is shown in figure 7.

$$\begin{array}{c|cc} & (B,1) & (C,1) \\ \hline (A,1) & 1 & 1 \end{array} \quad \otimes' \quad \begin{array}{c|cc} & (B,1) & (C,1) \\ \hline (A,1) & 2 & 2 \end{array} \quad = \quad \begin{array}{c|cccc} & (B,1) & (B,2) & (C,1) & (C,2) \\ \hline (A,1) & 1 & -\infty & 1 & -\infty \\ (A,2) & -\infty & 2 & -\infty & 2 \end{array}$$

Figure 7

3.2. Theorem. The structure

$$TABLES(V) = (tables(V), \partial'_0, \partial'_1, ;', \otimes', nil', I'_*)$$

is a symmetric strict monoidal category (the *monoidal category of tables* over V) with tables playing the role of morphisms, trivial table symmetries playing the role of objects, and I'_* playing the role of a natural transformation from $(\alpha, \beta) \mapsto \alpha \otimes' \beta$ to $(\alpha, \beta) \mapsto \beta \otimes' \alpha$. It contains $TSYMMETRIES(V)$, the subcategory of table symmetries with the members of $tsymmetries(V)$ playing the role of morphisms. □

To each cw-pomset $\alpha = [(X, \leq, d, e, s, t)]$ over V there corresponds the unique table $tab(\alpha) = (I, J, f)$ with

$$I(v) = length(s(v)) = cardinality(e^{-1}(v) \cap X_{min})$$

$$J(v) = length(t(v)) = cardinality(e^{-1}(v) \cap X_{max})$$

for all $v \in V$, and $f(m,n) = d((s(u))(i), (t(v))(j))$ for $m = (u,i)$ and $n = (v,j)$ such that $u, v \in V$ and $1 \leq i \leq I(u)$ and $1 \leq j \leq J(v)$.

3.3. Theorem. The correspondence $\alpha \mapsto tab(\alpha) : CWPOMSETS(V) \to TABLES(V)$ is a homomorphism. The restriction of this homomorphism to the subcategory of symmetries is an isomorphism from this subcategory to the subcategory of table symmetries. □

4 Processes and their delay tables

In general, by a process we mean here a finite partially ordered complex of time-consuming acts which transform some entities into some other entities. A process of this type, called a *time-consuming process*, can be represented by a cw-pomset $[(X, \leq, d, e, s, t)]$, where:

- elements of X represent the entities which take part in the process,

- the partial order $\leq$ specifies the causal succession of entities, i.e. how the entities cause each other,
- the weight function d specifies the least possible delays with which entities appear after their causal predecessors,
- the labelling function e specifies the meanings of entities,
- s and t are respectively an arrangement of entities which the process receives from its environment and an arrangement of entities which the process delivers to its environment.

It may be given either without specifying when its entities appear (a free time-consuming process), or together with the respective appearance times (a timed time-consuming process). In the first case the labelling function e specifies only the proper meaning of each entity from a given set V of meanings. In the second case e specifies an extended meaning which consists of the proper meaning and of the respective appearance time.

Let V be a set of meanings.

4.1. Definition. A *free time-consuming process* (or a *free tc-process*) over V is (a process which can be represented by) a cw-pomset over V. □

By $ftcprocesses(V)$ we denote the set of free tc-processes over V. Being identical with $cwpomsets(V)$ this set defines $FTCPROCESSES(V)$, the *monoidal category of free tc-processes* over V. According to 3.3, to each free tc-process α in this set there corresponds the table $tab(\alpha) \in tables(V)$, called the *delay table* of α.

4.2. Proposition. The correspondence $\alpha \mapsto tab(\alpha) : ftcprocesses(V) \to tables(V)$ is a homomorphism. □

4.3. Definition. A *timed time-consuming process* (or a *timed tc-process*) over V is (a process which can be represented by) a cw-pomset $\alpha = [(X, \leq, d, e, s, t)]$ over $V \times (-\infty, +\infty)$ such that $e = e_{proper} \times e_{time} : X \to V \times (-\infty, +\infty)$, i.e. $e(x) = (e_{proper}(x), e_{time}(x))$ with $e_{proper}(x) \in V$ and $e_{time}(x) \in (-\infty, +\infty)$ for $x \in X$, where

$$e_{time}(x) = max(e_{time}(y) + d(y, x) : y \leq x, y \neq x)$$

for all $x \in X - X_{min}$. □

By $ttcprocesses(V)$ we denote the set of timed tc-processes over V.

4.4. Proposition. The set $ttcprocesses(V)$ is closed w.r. to the compositions and interchanges. □

Being a closed subset of $cwpomsets(V \times (-\infty, +\infty))$ the set $ttcprocesses(V)$ defines a subcategory $TTCPROCESSES(V)$ of the monoidal category $CWPOMSETS(V \times (-\infty, +\infty))$, called the *monoidal category of timed tc-processes* over V. To each timed tc-process $\alpha = [(X, \leq, d, e, s, t)]$ in this set there corresponds the free tc-process $free(\alpha) = [(X, \leq, d, e_{proper}, s, t)]$ in $ftcprocesses(V)$ and the delay table $tab(free(\alpha))$ in $tables(V)$.

4.5. Proposition. The correspondence

$$\alpha \mapsto free(\alpha) : ttcprocesses(V) \to ftcprocesses(V)$$

is a homomorphism. $\square$

Timed tc-processes can be obtained by applying free tc-processes to *families of time sequences* of the form $M = (M(v) : v \in V)$, where each $M(v)$ is a finite sequence of time instants. Each such a family M represents the fact that entities with the respective meanings appear at specified time instants and it defines $timed(M) = (occ(t, M(v)) : v \in V)$, a *multiset of timed meanings*, where $occ(t, M(v))$ denotes the number of occurrences of the time instant t in the sequence $M(v)$, and it defines $st(M) = (length(M(v)) : v \in V)$, a *multiset of meanings*.

4.6. Proposition. Let $M = (M(v) : v \in V)$ be a family of time sequences and $\alpha = [(X, \leq, d, e, s, t)]$ a free tc-process over V with $\partial_0(\alpha)$ compatible with $st(M)$ in the sense that $length(s(v)) = length(M(v))$ for all $v \in V$. Let $\mathcal{B} = (X, \leq, d, e', s, t)$, where $e'(x) = (e'_{proper}(x), e'_{time}(x))$ with $e'_{proper}(x) = e(x)$ and $e'_{time}(x) = (M(v))(i)$ (the i-th element of $M(v)$) for $x \in X_{min}$ and $x = (s(v))(i)$ (the i-th element of $s(v)$), and $e'_{proper}(x) = e(x)$ and $e'_{time}(x) = max(e'_{time}(y) + d(y, x) : y \leq x, y \neq x)$ for $x \in X - X_{min}$. Then $\mathcal{B}$ is an instance of a timed tc-process $timed(M, \alpha)$ over V such that $\partial_0(timed(M, \alpha))$ is compatible with $timed(M)$ in the sense that they define the same multisets. The correspondence $(M, \alpha) \mapsto timed(M, \alpha)$ is surjective in the sense that each timed tc-process over V is of the form $timed(M, \alpha)$ for some M and α. $\square$

5 Processes of timed nets

Let $N = (Pl, Tr, pre, post, D)$ be a timed place/transition Petri net with a set Pl of places of infinite capacities, a set Tr of transitions, input and output functions $pre, post : Tr \to Pl^+$, where Pl^+ denotes the set of multisets of places, and with a duration function $D : Tr \to [0, +\infty)$. The multiset $pre(\tau)$ represents a collection of tokens, $pre(\tau, p)$ tokens in each place p, which must be consumed in order to execute a transition τ. The multiset $post(\tau)$ represents a collection of tokens, $post(\tau, p)$ tokens in each place p, which is produced by executing τ. The non-negative real number $D(\tau)$ represents the duration of each execution of τ. We assume that $pre(\tau) \neq 0$, $post(\tau) \neq 0$, $D(\tau) \neq 0$ for all transitions τ, and that $pre(\tau)$, $post(\tau)$, $D(\tau)$ determine τ uniquely.

A distribution of tokens in places is represented by a marking $\mu \in Pl^+$, where $\mu(p)$, the multiplicity of p in μ, represents the number of tokens in p. If many executions of transitions are possible for the current marking but there is too few tokens to start all these executions then a conflict which thus arises is resolved in an indeterministic manner. We assume that it takes no time to resolve conflicts: when an execution of a transition can start, it starts immediately, or it is disabled immediately. Finally, we admit many concurrent nonconflicting executions of the same transition.

The behaviour of N can be described by characterizing the possible processes of N, where a process is either an execution of a transition, or a presence of a token in a place, or a combination of such processes. Formal definitions are as follows.

5.1. Proposition. For $\tau \in Tr$ there exists a unique prime free tc-process $fproc(\tau)$ over Pl such that $fproc(\tau) = [(X, \leq, d, e, s, t)]$, where

(1) $X = X_{min} \cup X_{max}$ with X_{min} and X_{max} disjoint and such that $cardinality(e^{-1}(p) \cap X_{min}) = pre(\tau, p)$ and $cardinality(e^{-1}(p) \cap X_{max}) = post(\tau, p)$ for all $p \in Pl$,

(2) $d(x, x') = D(\tau)$ for all $x \in X_{min}$ and $x' \in X_{max}$. $\square$

5.2. Definition. A *free tc-process* of N is a member of

$$closure(oneelement(Pl) \cup fproc(Tr))$$

where $fproc(Tr)$ denotes the set of all free tc-processes $fproc(\tau)$ with $\tau \in Tr$. $\square$

By $fbeh(N)$ we denote the set of free tc-processes of N. Being closed w.r. to the compositions and interchanges this set defines a subcategory $FBEH(N)$ of the monoidal category $FTCPROCESSES(Pl)$. We call this subcategory the *algebra of free tc-processes* of N.

5.3. Proposition. For each $\tau \in Tr$ with $fproc(\tau) = [(X, \leq, d, e, s, t)]$ as in 5.1 there exist timed tc-processes, called *timed copies* of $fproc(\tau)$, which are of the form $\alpha = [(X, \leq, d, e', s, t)]$, where

(1) $e'_{proper} = e$,

(2) $e'_{time}(x') = max(e'_{time}(x) + D(\tau) : x \in X_{min})$ for all $x' \in X_{max}$. $\square$

5.4. Definition. A *timed tc-process* of N is a member of

$$closure(oneelement(Pl) \cup tproc(Tr))$$

where $tproc(Tr)$ denotes the set of all timed copies of free tc-processes $fproc(\tau)$ with $\tau \in Tr$. $\square$

By $tbeh(N)$ we denote the set of timed tc-processes of N. Being closed w.r. to the compositions and interchanges this set defines a subcategory $TBEH(N)$ of the monoidal category $TTCPROCESSES(Pl)$. We call this subcategory the *algebra of timed tc-processes* of N.

From 4.6 it follows that, being relatively small, the algebra of free tc-processes of N determines uniquely the much larger algebra of timed tc-processes of N. Nevertheless, we cannot avoid completely dealing with timed tc-processes since they are needed in order to formulate important concepts and problems. In particular, only in the case of timed tc-processes we can express that a process excludes another process due an earlier enabling of a transition, and only from timed tc-processes we are able to reconstruct classical firing sequences.

To be more precise, we start with an observation.

5.5. Proposition. Let α be any timed tc-process, let $\mathcal{A} = (X, \leq, d, e, s, t)$ be any instance of this process, and let u be an instant of time. Let $X(u)$ be the set of $x \in X$ such that either $e_{time}(x) \leq u$ or $e_{time}(y) \leq u$ for all y which are direct predecessors of x (which implies $X_{min} \subseteq X(u)$). Let $Y(u)$ be the subset of those elements of $X(u)$ which are maximal in $X(u)$. Then $Y(u)$ is a maximal antichain. □

The set $X(u)$ represents the entities which appear not later than at u or are results of prime components of α which start not later than at u (for instance, in the example in figure 8 the set $X(u)$ consists of A_1at3, $Bat4$, $Cat4$, A_2at1, $Bat3$, $Cat3$, D^2at4). The maximal antichain $Y(u)$ represents the entities which are present at u or are produced due to prime components of α which start not later than at u. Thus we obtain a set $\alpha|u$ of timed tc-processes of the form $[head_{Y(u),r}(\mathcal{A})]$ such that $\alpha = \alpha'; \alpha''$ with a unique α'' for each $\alpha' \in \alpha|u$, a multiset $\mu_{\alpha,u}$ of timed meanings, where $\mu_{\alpha,u}(v) = cardinality(e_{proper}^{-1}(v) \cap Y(u))$, and a multiset $\Theta_{\alpha,u}$ of prime free tc-processes, where $\Theta_{\alpha,u}(\pi) = |free(\alpha')|(\pi)$, the number of copies of π in $free(\alpha')$ for any $\alpha' \in \alpha|u$ (cf. 2.17). Moreover, the set $\alpha|u$ and the multisets $\mu_{\alpha,u}$, $\Theta_{\alpha,u}$ do not depend on the choice of instance of α, and we may regard $\Theta_{\alpha,u}$ as a multiset of transitions rather than of prime free tc-processes corresponding to transitions.

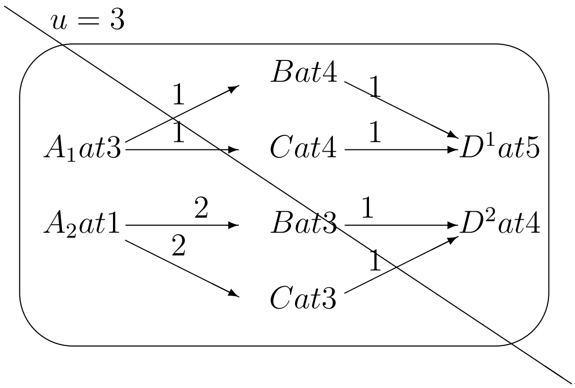

Figure 8

The phenomenon of exclusion of a timed tc-process by another such a process due to an earlier enabling of a prime component can be described with the aid of concepts of dominance and admissibility.

Given two timed tc-processes α and β, we say that β *dominates* α if there exists a time instant u_0 such that $\Theta_{\alpha,u} = \Theta_{\beta,u}$ and $\Theta_{\alpha,u} = \Theta_{\alpha,u_0}$ and $\Theta_{\beta,u} < \Theta_{\beta,u_0}$ for $u < u_0$. Given any set P of timed tc-processes, a member α of P is said to be *admissible* in this set if there is no $\beta \in P$ which dominates α. Thus P determines a subset $admissible(P)$ of its admissible members.

With these concepts we are able to say which timed tc-processes of the considered net N are realizable and to describe how they define firing sequences of N.

Namely, to each $\alpha \in tbeh(N)$ a sequence $-\infty = u_0 < u_1 < ... < u_n < u_{n+1} = +\infty$ there corresponds such that $\alpha|u$, $\mu_{\alpha,u}$, $\Theta_{\alpha,u}$ are constant and respectively equal to some α_i,μ_i,Θ_i on each interval $[u_i, u_{i+1})$. The multisets μ_i of timed meanings can be regarded as *timed markings* whose items represent appearances of tokens in specified places at specified time instants. In this manner to α a sequence $fs(\alpha) = \mu_0[\Theta_1)\mu_1...[\Theta_n)\mu_n$ there corresponds which may be regarded as a candidate for a possible firing sequence of N.

Whether indeed α can be realized and thus $fs(\alpha)$ is a possible firing sequence depends on whether the process α can be excluded by another process due to an earlier enabling of a transition and it can be reflected with the aid of notions of dominance and admissibility.

Thus we obtain a set $admissible(tbeh(N))$ of admissible timed tc-processes of N such that only members of this set can be realized in N, and firing sequences of N can be defined as $fs(\alpha)$ for admissible α. This is justified by the following fact.

5.6. Theorem. If $fs(\alpha) = \mu_0[\Theta_1)\mu_1...[\Theta_n)\mu_n$ for some $\alpha \in admissible(tbeh(N))$ then for each $i = 1, ..., n$ there exists a time instant u_i such that

(1) u_i is the earliest instant of time such that, for all $p \in Pl$,

$$\Sigma(\mu_{i-1}(p, u) : u \leq u_i) \geq pre(\tau, p),$$

(2) Θ_i is a maximal multiset of transitions such that, for all $p \in Pl$,

$$\Sigma(\mu_{i-1}(p, u) : u \leq u_i) \geq \Sigma(\Theta_i(\tau)pre(\tau, p) : \tau \in Tr),$$

(3) for all $u > u_i$ and all $p \in Pl$ we have

$$\mu_i(p, u) = \mu_{i-1}(p, u) + \Sigma(\Theta_i(\tau)post(\tau, p) : \tau \in Tr, u_i + D(\tau) = u)$$

and

$$\Sigma(\mu_i(p, u) : u \leq u_i) = \Sigma(\mu_{i-1}(p, u) : u \leq u_i) - \Sigma(\Theta_i(\tau)pre(\tau, p) : \tau \in Tr).$$

Conversely, each sequence $\mu_0[\Theta_1)\mu_1...[\Theta_n)\mu_n$, where $\mu_0, \mu_1, ..., \mu_n$ are timed markings and $\Theta_1, ..., \Theta_n$ are multisets of transitions, such that for each $i = 1, ..., n$ there exists a time instant u_i such that the conditions (1) - (3) are satisfied is of the form $fs(\alpha)$ for some $\alpha \in admissible(tbeh(n))$. $\square$

6 Closing remarks

The representation of the behaviours of timed Petri nets in terms of processes and their delay tables seems to be conceptually simple due to its algebraic nature. In this representation nets can be viewed as sets of atomic generators of their behaviours considered as subalgebras of a monoidal category. Processes which constitute such behaviours determine in a natural way their execution times in the form of delay tables rather than of single numbers. This seems to be adequate for many applications and allows the parallel composition of processes to be a bifunctor.

The descriptions of behaviours of timed Petri nets in terms of processes have this advantage over descriptions in terms of firing sequences that the behaviours of large nets can be obtained by combining the behaviours of their components. This follows from the simple observation that such descriptions are compositional in the sense that

$$fbeh(N) = closure(fbeh(N_1) \cup ... \cup fbeh(N_k))$$

and

$$tbeh(N) = closure(tbeh(N_1) \cup ... \cup tbeh(N_k))$$

whenever N consists of subnets $N_1, ..., N_k$ which possibly share places, but have mutually disjoint sets of transitions.

Moreover, the descriptions in terms of processes are more economical than the descriptions in terms of firing sequences since one process can represent a set of firing sequences.

References

[BG 92] Brown, C., Gurr, D., *Timing Petri Nets Categorically* , Springer LNCS 623, Proc. of ICALP'92, 1992, pp.571-582

[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

[GMMP 89] Ghezzi, C., Mandrioli, D., Morasca, S., Pezze, M., *A General Way to Put Time in Petri Nets*, Proc. of the 5th Int. Workshop on Software Specifications and Design, Pittsburgh, May 1989, IEEE-CS Press

[MMS 92] Meseguer, J., Montanari, U., Sassone, V., *On the Semantics of Petri Nets*, Springer LNCS 630, Proc. of CONCUR'92, 1992, pp.286-301

[Pra 86] Pratt, V., *Modelling Concurrency with Partial Orders*, International Journal of Parallel Programming, Vol.15, No.1, 1986, pp.33-71

[Wi 80] Winkowski, J., *Behaviours of Concurrent Systems*, Theoretical Computer Science 12, 1980, pp.39-60

[Wi 92] Winkowski, J., *An Algebra of Time-Consuming Computations*, Institute of Computer Science of the Polish Academy of Sciences, Technical Report 722, December 1992

[Wi 93] Winkowski, J., *A Representation of Processes of Timed Petri Nets*, Institute of Computer Science of the Polish Academy of Sciences, Technical Report 728, August 1993

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