2005.08.10 (Fundamenta Informaticae 68 (1-2) 2005, 175-206)

TOWARDS A FRAMEWORK FOR MODELLING SYSTEMS WITH RICH INTERNAL STRUCTURES OF STATES AND PROCESSES

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

Abstract

The paper is concerned with modelling distributed systems by specifying their states and processes. Processes are defined as activities in a universe of objects, each object with a set of possible internal states, each activity changing states of some objects and establishing or destroying relations among objects. Partial operations of composing processes sequenially and in parallel are defined. It is shown that certain sets of processes of form together with these operations categories with additional structures and special properties, that processes of a system can be represented as morphisms of such categories, and that independence of processes can be characterized in a natural, purely algebraic way.

Key words

Distributed systems, states, processes, structures, sequential composition, parallel composition, category, independence.

1 Introduction

In this paper we propose a framework for modelling distributed systems. We think of a framework such that:

- systems are modelled by specifying their states and runs from a state to a state, called processes,
- states and processes of a system are represented together with their components and internal structures,
- processes of a system that consist of many immediate transitions from a state to a state are represented,
- specification of processes reflects how processes consist of processes,
- dependence of processes on contexts is reflected,
- independence of state components and processes is reflected,
- processes in which only a part of a system is involved can be represented as local to this part and to its extensions,

- processes are defined in a way that does not exclude a possibility of representing runs of continuous system.

In order to develop such a framework we formulate a general, system independent definition of processes, define partial operations of composing processes sequentially and in parallel, define categories of processes with the composition given by the sequential composition of processes, and propose to characterize systems and their behaviours as subsets of such categories.

When speaking of processes we have in mind only nonbranching processes. Though our intention is to define processes as general as possible, in the present paper we think mainly of processes like those considered for Petri nets of various kinds and for graph grammars (cf. [Petri 77], [Wink 80], [RT 86], [BD 87], [DMM 89], [MR 95], [MMS 96], [CMR 96]). We define categories of processes and present their basic properties. In particular, we show that independence of processes is equivalent to the existence in these categories of suitable bicartesian squares. This implies that such categories are members of an axiomatically defined class of categories with axioms allowing to define independence of morphisms. We show that by reducing categories from this class, called discrete process categories, to their objects and atomic morphisms, and by endowing the results of reduction with the existing information on independence, we obtain structures close to transition systems with independence of [WN 95]. Finally, we show that our transition systems with independence generate freely discrete process categories.

Our definition of processes is formulated with the idea of reflecting the internal structure of processes such that the sequential and the parallel composition of processes can be defined and the information essential for defining independence of processes is also reflected. To this end processes are represented as activities in a universe of objects, each object with a set of possible internal states and instances corresponding to these states, each activity changing states of some objects and relations among objects, where changes are viewed as replacements of existing occurrences of active objects by new occurrences. Activities of a system may consist of atomic activities or be continuous and infinitely divisible. In order to define operations on processes isomorphic activities are identified. The choice of the universe of objects depends on the nature of systems to be represented.

1.1. Example. Processes of a Place/Transition Petri net can be regarded as activities in the universe $U = (W, V, ob)$, where $V = \{v_1, v_2, ...\}$ is an infinite set of objects which may become tokens in places of the net, each object v with the possible states *passive*, *active*, *terminated*, and the respective instances v^-, v^+, $v^\bullet$, and where W is the set of instances of objects from V and $ob : W \to V$ is the mapping that assigns the respective object to its instances, i.e., $ob(v^-) = ob(v^+) = ob(v^\bullet) = v$.

Consider for example the net in figure 1.1 and its markings M and M', where M consists of a token t_1 in p, a token t_2 in q, and two tokens t_3 and t_4 in r and M' consists of a token t'_1 in p' and a token t'_2 in q', and consider the transformation of M into M' shown in figure 1.2.

This transformation is a process P. It can be regarded as one of the activities in the universe U and represented by the partially ordered set X of occurrences of objects from U that is shown in figure 1.3.

The partial order of occurrences of objects is represented by arrows. It reflects the flow order in the set of occurrences of objects, that is how occurrences arise from occurrences.

The correspondence between occurrences of objects and the respective object instances is represented by a function, in this case the function given by
$f(x_1) = v_1^+$, $f(x_3) = v_3^+$, $f(x_5) = v_5^-$, $f(x_2) = v_2^+$, $f(x_4) = v_4^+$,$f(x_6) = v_6^-$, $f(x_1') = v_1^\bullet$, $f(x_3') = v_3^\bullet$, $f(x_5') = v_5^+$, $f(x_2') = v_2^\bullet$, $f(x_4') = v_4^\bullet$, $f(x_6') = v_6^+$, $f(x_7) = v_7^-$, $f(x_8) = v_8^-$,... .

The elements x_5, x_6, x_7, x_8,... are occurrences of passive objects v_5^-, v_6^-, v_7^-, v_8^-,..., where each passive object can become a token. The elements x_1', x_2', x_3', x_4' are occurrences of terminated objects $v_1^\bullet$, $v_2^\bullet$, $v_3^\bullet$, $v_4^\bullet$, where each terminated object has already been a token and cannot be a token anymore. The elements x_1, x_2, x_3, x_4, x_5', x_6' are occurrences of active objects v_1^+, v_2^+, v_3^+, v_4^+, v_5^+, v_6^+, where each active object plays the role of a token residing in a place. They form the subsets $X_p = \{x_1\}$, $X_q = \{x_2\}$, $X_r = \{x_3, x_4\}$, $X_{p'} = \{x_5'\}$, $X_{q'} = \{x_6'\}$ of the set X, each subset corresponding to a place.

Thus the considered activity can be regarded as $P = (X, \leq, f, S)$, where $\leq$ is the causal relation, and S is the structure on X given by the mutually disjoint subsets X_p, X_q, X_r, $X_{p'}$, $X_{q'}$, that is $S = (X_p, X_q, X_r, X_{p'}, X_{q'})$.

The construct P represents a concrete process of the considered system. In order to define operations on processes we must consider such constructs up to isomorphisms. Consequently, the abstract process represented by the construct P should be regarded as the corresponding isomorphism class π of such constructs, that is as the isomorphism class $[P]$ that contains P. We represent it graphically in figure 1.4.

Note that the process π is the result of composing sequentially processes π_1 and π_2 in figure 1.5 in the sense that it consists of π_1 followed by π_2.

Similarly, π is the result of composing in parallel processes ρ_1 and ρ_2 in figure 1.6.

Note also that due to relating occurrences of each object in a process to an instance of the object, and to the object itself, we are able to trace the history of each object and thus to define independence of processes as lack of conflicts at shared objects. $\square$

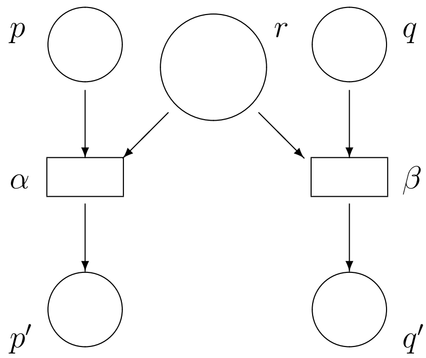


Figure 1.1: A Place/Transition Petri net N

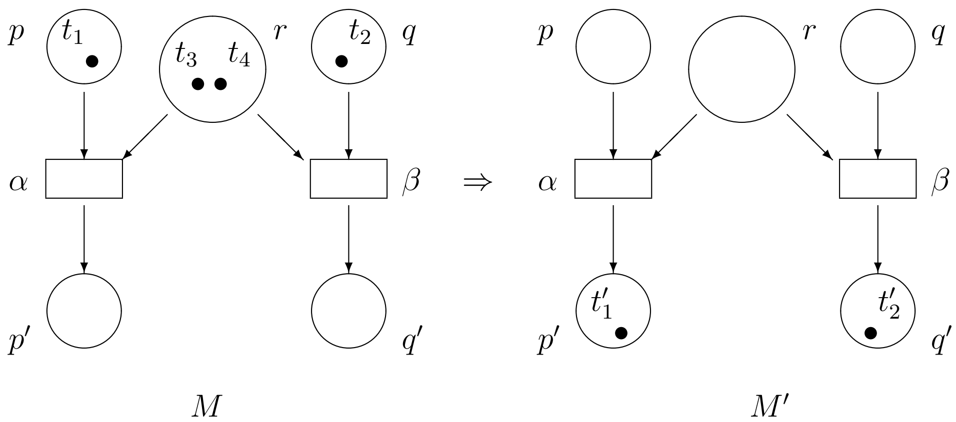


Figure 1.2: Transformation of the marking M into the marking M'

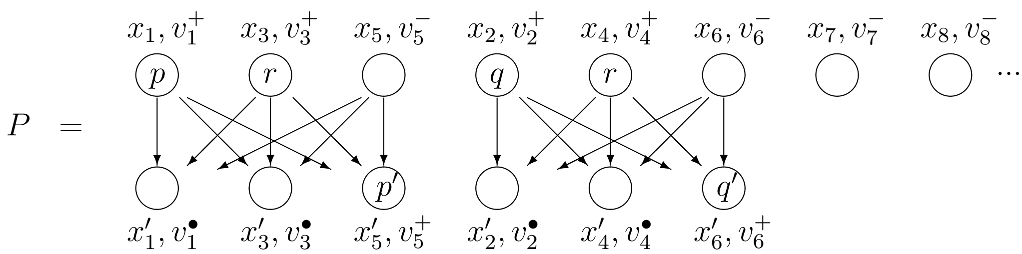


Figure 1.3: The representation of P as an activity in U

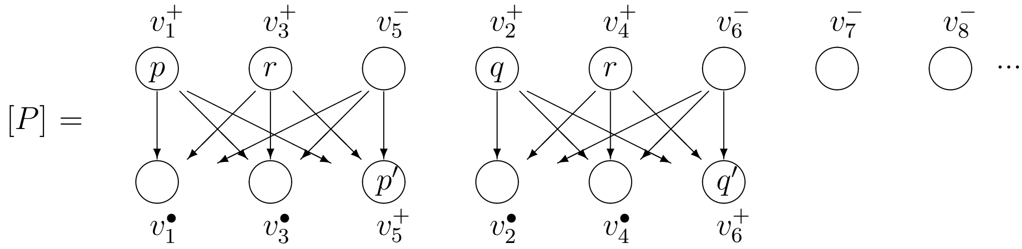


Figure 1.4: The representation of the abstract process $[P]$

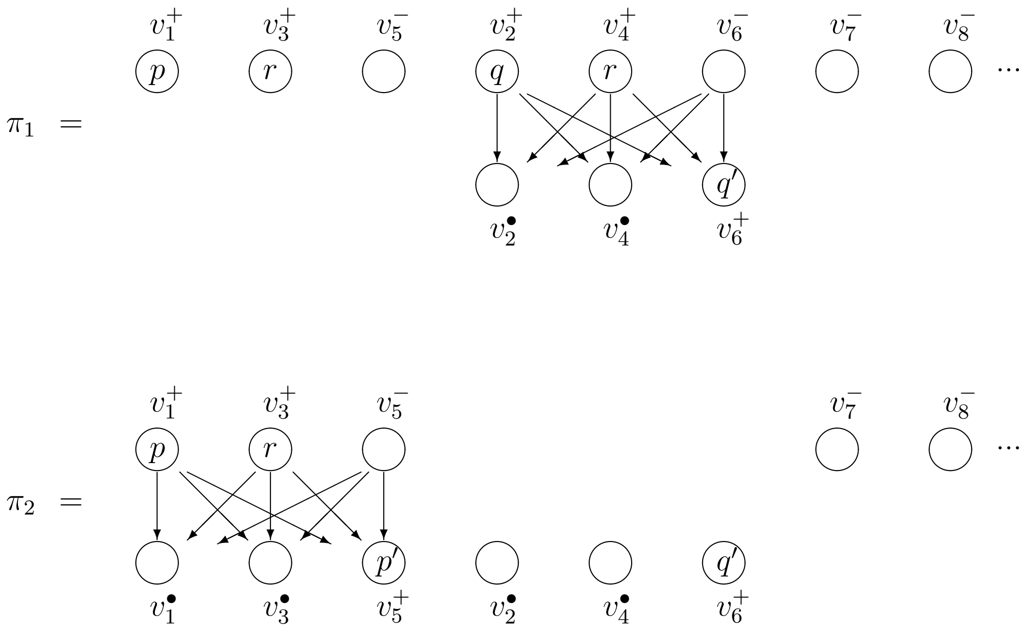
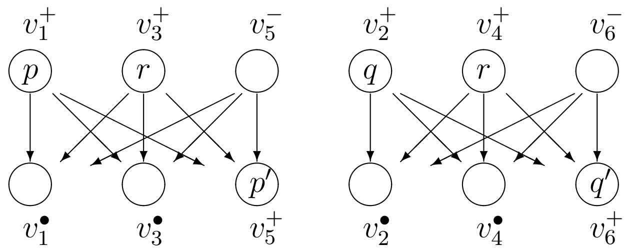


Figure 1.5: Processes π_1 and π_2

$\rho_1 \;=$ 

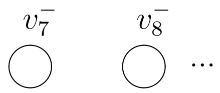
$\rho_2 \;=$ 

Figure 1.6: Processes ρ_1 and ρ_2

In the example just described the structure imposed on the set of object occurrences is very simple. Moreover, the flow order is the only mechanism defining the possible stages of process development. In general, we may need more complicated structures and, moreover, we may have to consider not only the flow order, but also reflect the fact that some object occurrences may play the role of a context for other object occurrences.

1.2. Example. Imagine that two robots A and B are supposed to transport two objects G and H through a corridor C and paint the floor of the corridor. Assume that the robots may transport objects provided that the corridor is dry, that is not painted yet, and that they may do it concurrently. States of this system can be represented in terms of occurrences of objects A, B, C, G, H and the relations *free*, *dry*, *painted*, *at*, *behind*, in the set of such occurrences, where $\mathit{free}(x)$, $\mathit{dry}(y)$, $\mathit{painted}(y)$, $\mathit{at}(z, y)$, $\mathit{behind}(z, y)$, stand respectively for "robot x is free", "corridor y is dry", "corridor y is painted", "object z is waiting at corridor y", and "object z is behind corridor y".

Consider a process P' in which A transports G and concurrently B transports H, and next A paints C. This process can be regarded as an activity in the universe $U' = (W', V', ob')$, where $W' = V' = \{A, B, C, G, H\}$ and $ob'(w) = w$, and it can be represented by the partially ordered set X' of occurrences A_1, A_2, A_3, B_1, B_2, C_1, C_2, G_1, G_2, H_1, H_2 of the objects A, B, C, G, H equipped with the relation cxt' of contextual dependence and with the structure S', where the partial order and the contextual dependence are represented in figure 1.7 respectively by arrows and dotted arrows, and where S' consists of the following relations:
$\mathit{free} = \{A_1, A_2, A_3, B_1, B_2\}$, $\mathit{dry} = \{C_1\}$, $\mathit{painted} = \{C_2\}$,
$\mathit{at} = \{(G_1, C_1), (H_1, C_1)\}$, $\mathit{behind} = \{(G_2, C_1), (H_2, C_1), (G_2, C_2), (H_2, C_2)\}$.

Note that in the process the robots occur only as free. However, the situations of each robot before and after its action are represented by different occurrences of this robot and this reflects participation of the robot in the action.

The relation cxt' specifies some of object occurrences as possible only in the context of presence of some other object occurrences. Consequently, it extends the flow order of object occurrences, $\leq'$, by enforcing x to precede y if $(z, x) \in cxt'$ and $z \leq' y$ for some z, or $x \leq' z$ and $(z, y) \in cxt'$ for some z. For example, B_2 must precede C_2 since $(C_1, B_2) \in cxt'$ and $C_1 \leq C_2$. Similarly, H_2 must precede C_2 since $(C_1, B_2) \in cxt'$ and $C_1 \leq' C_2$. This reflects the fact that transporting of the object H by the robot B must precede painting of the corridor C.

Thus P' can be regarded as $(X', \leq', cxt', f', S')$, where
$X' = \{A_1, A_2, A_3, B_1, B_2, C_1, C_2, G_1, G_2, H_1, H_2\}$,
$\leq'$ is the partial order represented in figure 1.7 by arrows,
$cxt' = \{(C_1, G_2), (C_1, A_2), (C_1, B_2), (C_1, H_2)\}$,
$f'(A_1) = f'(A_2) = f'(A_3) = A$, $f'(B_1) = f'(B_2) = B$, $f'(C_1) = f'(C_2) = C$, $f'(G_1) = f'(G_2) = G$, $f'(H_1) = f'(H_2) = H$,
and $S' = (\mathit{free}, \mathit{dry}, \mathit{painted}, \mathit{at}, \mathit{behind})$. $\square$

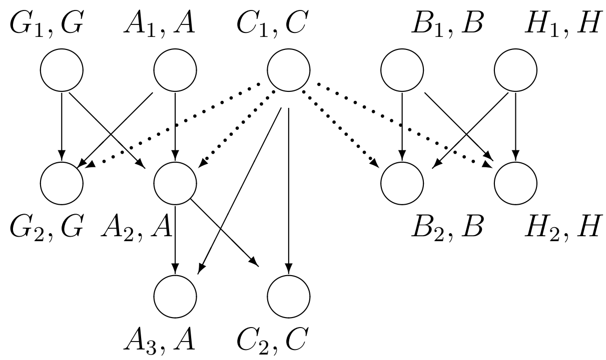


Figure 1.7: The underlying occurrence structure of P'

It is important to realize that the proposed way of representing processes is very general and it applies not only to discrete processes.

1.3. Example. A two-dimensional variant of the four-dimensional space-time world of relativity theory with Minkowski metric can be regarded as an activity in the universe $U'' = (W'', V'', ob'')$, where V'' is the set of real numbers, *Real*, each number v representing a particle, $W'' = V'' \times \mathit{Real}$, where each pair $(v, p) \in W''$ represents the particle v in the position p, and ob'' is the mapping defined by $ob''(v, p) = v$ (cf. [Carn 58]). We can represent such an activity as $P'' = (X'', \leq'', f'', S'')$, where

X'' is the set of occurrences of objects from U'',
$f'' : X'' \to W''$,

$S'' = (position : X'' \to Real, delay : (X'')^2 \to Real, distance : (X'')^2 \to Complex)$, is the structure that consists of:

a function *position* such that $f''(x) = (ob(f''(x)), position(x))$,

a function $delay : (X'')^2 \to Real$ such that

$delay(x, x') = delay(x, x'') + delay(x'', x')$,

the function $distance : (X'')^2 \to Complex$ with *Complex* denoting the set of complex numbers and

$distance(x, x') = ((c^2(delay(x, x'))^2 - (position(x) - position(x'))^2)^{1/2}$,

$\leq''$ is the partial order on X'' defined by:

$x \leq x'$ iff $0 \leq delay(x, x')$ and $|position(x) - position(x')| \leq c\ delay(x, x')$,

where c is the speed of light. □

The above examples suggest that it is possible to elaborate a universal model for a broad class of processes. In the rest of the paper we offer a candidate for such a model and, for a subclass of processes, we define operations which allow one to construct processes from processes, define the respective categories, and describe how to define for each system from a class the category of its processes.

The paper is organized as follows. In section 2 we recall the notion of a structure. In section 3 we introduce a general notion of a process. In section 4 we define operations on processes and categories of processes. In section 5 we introduce two notions of independence of processes similar to those considered in [Wink 03] for processes of Petri nets, and we characterize these notions in algebraic terms. In section 6 we describe the algebraic properties of categories of processes and the relation between such categories and transition systems with independence. Finally, in section 7 we discuss our results and describe how they are related to other work.

Some of the results described in sections 5 and 6 are obtained in essentially the same way as the similar results in [Wink 03] for Petri nets. Nevertheless, in order to make the presentation complete, we sketch the corresponding proofs also in the present paper.

An early version of the present paper was published in March 2004 as the report 973 of the Institute of Computer Science of the Polish Academy of Sciences.

2 Structures

By structures we mean slightly modified versions of structures in the sense of Bourbaki's Elements. We define them as follows.

Let $\times$ and $\mathcal{P}$ denote respectively the symbol of the cartesian product and the symbol of the operation that assigns to each set the set of all its subsets, that is its powerset.

2.1. Definition. A *structure form* with a variable x is either the variable x, or a constant, or an expression of the form $s_1 \times ... \times s_n$, where s_1,...,s_n are structure forms with the variable x, or an expression $\mathcal{P}(s)$, where s is a structure form with the variable x. □

In the sequel a structure form s with a variable x and constants a_1,...,a_m is writen as $s(x; a_1, ..., a_m)$.

2.2. Definition. Given a structure form $s(x; a_1, ..., a_m)$ with a_1,...,a_m denoting certain *auxiliary sets* A_1,...,A_m, a *structure* of this form on a set X is an element S of $s(X; A_1, ..., A_m)$, that is of the value of the expression $s(x; a_1, ..., a_m)$ for the substitution of the set X for the variable x and the auxiliary sets A_1,...,A_m for the constants a_1,...,a_m. The set X is called the *carrier* of such a structure. $\square$

The following examples of structures illustrate the notion.

The family $S = (X_p, X_q, X_r, X_{p'}, X_{q'})$ of mutually disjoint subsets of the set X of object occurrences of the process P in example 1.1 is a structure of the form $s(x; a) = \mathcal{P}(a \times \mathcal{P}(x))$ on X with a denoting the auxiliary set $\mathit{Places} = \{p, q, r, p', q'\}$.

The process P itself is the set X equiped with the structure $S_1 = (\leq, f, S)$ of the form $s_1(x; a, b) = \mathcal{P}(x \times x) \times \mathcal{P}(x \times b) \times \mathcal{P}(a \times \mathcal{P}(x))$, where a and b denote respectively the auxiliary sets *Places* and W.

The system $S' = (\mathit{free}, \mathit{dry}, \mathit{painted}, \mathit{at}, \mathit{behind})$ of the relations *free*, *dry*, *painted*, *at*, *behind* on the set X' of object occurrences of the process P' from example 1.2 is a structure of the form $s'(x) = \mathcal{P}(x) \times \mathcal{P}(x) \times \mathcal{P}(x) \times \mathcal{P}(x \times x) \times \mathcal{P}(x \times x)$ on X'.

The system $S'' = (\mathit{position}, \mathit{delay}, \mathit{distance})$ of the functions *position*, *delay*, *distance* in example 1.3 is a structure of the form $s''(x; r) = \mathcal{P}(x \times r) \times \mathcal{P}(x \times x \times r) \times \mathcal{P}(x \times x \times c)$ on the set X'' with r and c denoting respectively the auxiliary sets *Real* and *Complex*.

A graph with a set V of vertices, a set E of edges, a source function $s : E \to V$, and a target function $t : E \to V$, is the structure $G = (V, E, s, t)$ of the form $g(x) = \mathcal{P}(x) \times \mathcal{P}(x) \times \mathcal{P}(x \times x) \times \mathcal{P}(x \times x)$ on $X = V \cup E$.

A topology $\mathcal{T}$ on a set X is a structure of the form $\tau(x) = \mathcal{P}(\mathcal{P}(x))$ on X, etc.

Given a structure form $s(x; a_1, ..., a_m)$ with a_1,...,a_m denoting auxiliary sets A_1,...,A_m, each mapping $f : X \to X'$ induces a mapping from $s(X; A_1, ..., A_m)$ to $s(X'; A_1, ..., A_m)$ according to the following rules:

(1) $f'(z) = z$ if $z \in s(X; A_1, ..., A_m)$
and $s(X; A_1, ..., A_m)$ is one of the auxiliary sets A_1,...,A_m,

(2) $f'(z_1, z_2) = (f_1(z_1), f_2(z_2))$ if $(z_1, z_2) \in s(X, A_1, ..., A_m)$
and $s(X; A_1, ..., A_m) = s_1(X; A_1, ..., A_m) \times s_2(X; A_1, ..., A_m)$
and $f_1 : s_1(X; A_1, ..., A_m) \to s_1(X'; A_1, ..., A_m)$
and $f_2 : s_2(X; A_1, ..., A_m) \to s_2(X'; A_1, ..., A_m)$,

(3) $f'(z) = \{f(u) : u \in z\}$ if $z \in s(X; A_1, ..., A_m)$
and $s(X; A_1, ..., A_m) = \mathcal{P}(s'(X; A_1, ..., A_m))$
and $f : s'(X; A_1, ..., A_m) \to s'(X'; A_1, ..., A_m)$.

2.3. Definition. An *isomorphism* from a structure $S \in s(X; A_1, ..., A_m)$ of the form $s(x; a_1, ..., a_m)$ on X to a structure $S' \in s(X'; A_1, ..., A_m)$ of the same form on X' is a bijection $f : X \to X'$ such that S' is the image of S under the mapping induced by f. □

2.4. Definition. A *structure type* is an axiomatically defined class T of structures of the same form $s(x; a_1, ..., a_m)$ such that if a structure $S \in s(X; A_1, ..., A_m)$ belongs to T and there is an isomorphism from this structure to a structure $S' \in s(X'; A_1, ..., A_m)$ then also the latter structure belongs to T. □

For example, the type of the structure S from example 1.1 is given by the axiom declaring S as a mapping from the set *Places* to the set of subsets of the carrier X such that the subsets which correspond to different elements are disjoint. The type of graphs is given by the axiom declaring that the set of vertices and the set of edges are disjoint and that the source function and the target function are mappings from the set of edges to the set of vertices.

What we have said starting from the syntactic definition of structure forms can be expressed semantically in the language of category theory.

Let *Ens* and *BijEns* denote respectively the category of sets and mappings and the category of sets and bijective mappings. A structure form can be defined as a functor $F : \mathit{Ens} \to \mathit{Ens}$ that can be built from the identity and constant functors using the powerset functor $\mathcal{P} : \mathit{Ens} \to \mathit{Ens}$ and the bifunctor $\times : \mathit{Ens} \times \mathit{Ens} \to \mathit{Ens}$ of cartesian product. A structure type of structures of the form F can be defined a functor $T : \mathit{BijEns} \to \mathit{BijEns}$ such that $T(b) = F(b)$ for each bijection $b : X \to X'$ and $T(X) \subseteq F(X)$ for each set X (cf. [BuDe 68]).

For convenience we assume the existence of a distinguished element *none* that represents the lack of a structure satisfying given conditions on a given set and is regarded as a structure of each possible type.

Note that each structure of a form $s(x; a_1, ..., a_m)$ on a subset X' of a set X can be regarded a structure of the same form on entire X. We exploit this fact and admit in the carriers of structures elements which do not occur in the respective structures. For example, we admit in the carrier of a graph elements which are neither vertices nor edges, we admit in a topological structure elements which do not belong to any open set of this structure, etc. This simplifies some constructions on structures by creating the possibility of transporting the respective structures to a common carrier. In particular, we can think of restrictions of structures to subsets of their carriers and of common extensions of structures satisfying appropriate compatibility conditions. The respective concepts can be introduced as follows.

Given a structure form $s(x; a_1, ..., a_m)$, such a form has a unique *normal representation* $y_1 \times ... \times y_k$ where each y_i is an expression of one of the forms x, a_j, $\mathcal{P}(z)$. Consequently, for a_1,...,a_m denoting auxiliary sets A_1,...,A_m, and for each set X, the set of all structures of the form $s(x; a_1, ..., a_m)$ on X is partially ordered by the relation $\ll_X$, where $S \ll_X S'$ iff $S = (S_1, ..., S_k)$ and $S' = (S'_1, ..., S'_k)$, $S_i = S'_i$ for y_i of the form x or a_j, and $S_i \subseteq S'_i$ for y_i of the form $\mathcal{P}(z)$.

In the language of category theory the normal representation of a structure form given by a functor $F : \mathit{Ens} \to \mathit{Ens}$ can be defined as the representation with $F(X) = Y_1 \times ... \times Y_k$, where $Y_i = X$, or $Y_i = A_j$ with A_j being the value of a constant functor, or $Y_i = \mathcal{P}(Z)$ with some Z, and we can define $S \ll_X S'$ iff $S = (S_1, ..., S_k)$ and $S' = (S'_1, ..., S'_k)$ with $S_i = S'_i$ for $Y_i = X$ or $Y_i = A_j$, and with $S_i \subseteq S'_i$ for $Y_i = \mathcal{P}(Z)$.

The existence of the partial order $\ll_X$ makes clear what is the greatest or the least structure on X that satisfies given conditions. In particular, we can formulate the following definitions which plays an important role in our approach to combining processes.

2.5. Definition. Given a structure form $s(x; a_1, ..., a_m)$, a structure type T of structures of this form, and a structure $S \in s(X; A_1, ..., A_m)$ of type T on a set X, the *restriction* of such a structure to a subset Y of X is the structure R of the same form $s(x; a_1, ..., a_m)$ and type T on Y such that R is the greatest structure of type T on Y whose image under the mapping from $s(Y; A_1, ..., A_m)$ to $s(X; A_1, ..., A_m)$ induced by the inclusion $Y \subseteq X$ is less than S. If such a restriction exists then we write R as $S|Y$. $\square$

For example, the restriction of a graph $G = (V, E, s, t)$ to a subset Y of $X = V \cup E$ such that $s(e) \in Y$ and $t(e) \in Y$ whenever $e \in Y$ is the structure $F = (V_Y, E_Y, s_Y, t_Y)$, where $V_Y = V \cap Y$, $E_Y = E \cap Y$, $s_Y = \{(e, v) \in E \times V : (e, v) \in s \text{ and } e, v \in Y\}$ and $t_Y = \{(e, v) \in E \times V : (e, v) \in t \text{ and } e, v \in Y\}$.

The restriction of a structure to a subset of its carrier is unique if it exists.

Note that there may be structures that have no restrictions to some subsets of their carrier. For example, such are structures consisting of a single element of their carrier.

Note also that our concept of restriction of a structure to a subset of its carrier coincides with the standard concept of restriction for relational structures of a form, but for topological structures and algebras this is true only when the respective subset is closed or a subalgebra.

2.6. Definition. Given a structure type T and two structures of this type: a structure P on a set X and a structure Q on a set Y, such that there exist the restrictions $P|X \cap Y$ and $Q|X \cap Y$ of these structures of type T, and $P|X \cap Y = Q|X \cap Y$, we say that the structures P and Q are *compatible* and define their *common extension* as a structure R of type T on $X \cup Y$ such that $P = R|X$ and $Q = R|Y$, if such a structure exists. $\square$

2.7. Definition. A structure type T is said to be *admitting the least common extensions of its compatible structures* if every two compatible structures of this type have the least common extension. $\square$

3 Processes

We start with some preliminaries.

Given a partial order $\leq$ on a set X, we call $\mathcal{X} = (X, \leq)$ a *partially ordered set*, by the *strict partial order* corresponding to $\leq$ we mean $<$, where $x < y$ iff $x \leq y$ and $x \neq y$, by a *cross-section* of $\mathcal{X}$ we mean a maximal antichain of $\mathcal{X}$ which has an element in each maximal chain of $\mathcal{X}$, for a cross-section Z we define $X^-(Z) = \{x \in X : x \leq z \text{ for some } z \in Z\}$ and $X^+(Z) = \{x \in X : z \leq x \text{ for some } z \in Z\}$, we say that a cross-section Z' *precedes* a cross-secton Z'' and write $Z' \sqsubseteq Z''$ if $X^-(Z') \sqsubseteq X^-(Z'')$, and for cross-sections Z' and Z'' such that $Z' \sqsubseteq Z''$ we define a *segment* of $\mathcal{X}$ from Z' to Z'' as $[Z', Z''] = X^+(Z') \cap X^-(Z'')$. We say that $\mathcal{X} = (X, \leq)$ (and $\leq$) is K-*dense* if all maximal antichains of $\mathcal{X}$ are cross-sections (cf. [Petri 77]).

Processes in a universe of objects can be defined as follows.

Let T be a structure type. Let $U = (W, V, ob)$ be a universe of objects, where V is a set of *objects*, W is a set of *instances* of objects from V, and $ob : W \to V$ is the mapping that assigns the respective object to each of its instances. A subset $W' \subseteq W$ of object instances is said to be *consistent* if it contains at most one instance of each object. Such a subset is said to be *complete* if it contains exactly one instance of each object.

3.1. Definition. A *concrete process* of the type T over U is $P = (X, \leq, cxt, in, str)$, where

(1) X is a set of *object occurrences*,

(2) $in : X \to W$ is a mapping that assigns an object instance to each object occurrence,

(3) $\leq$ is a partial order on X (the *flow order*) such that each element of X belongs to a cross-section of $(X, \leq)$ and, for each object $v \in V$, the set

$\{x \in X : ob(in(x)) = v\}$ is either a maximal chain or it is empty,

(4) cxt is an acyclic binary relation on X (the *context relation*) such that, for all elements of X, $(x, y) \in cxt$ excludes both $x \leq y$ and $y \leq x$, and the reflexive and transitive closure of the following relation R, where cxt^+ denotes the transitive closure of cxt, is a partial order $\preceq$:

$$\begin{aligned}(x, y) \in R \text{ iff } & x \leq y \text{ or } (x < z \text{ and } (z, y) \in cxt^+ \text{ for some } z) \\ & \text{or } (x < t \text{ and } z < y \text{ and } (z, t) \in cxt \text{ for some } z \text{ and } t),\end{aligned}$$

(5) str is a structure of type T on X such that, for each segment $[Z', Z'']$ of $(X, \leq)$ from Z' to Z'', where the cross-sections Z' and Z'' are antichains of $(X, \preceq)$, there exists the restriction $str|[Z', Z'']$ of str to this segment and this restriction is a structure of type T. $\square$

We use subscripts, X_P, $\leq_P$, cxt_P, in_P, str_P, $\preceq_P$, when necessary.

Condition (4) guarantees that an object occurrence cannot end before object occurrences for which it is a context. Condition (5) guarantees that object ocurrences belonging to a potentially observable segment of P form a structure of the declared type.

The definition can be illustrated by examples.

The process P from example 1.1 is a concrete process with $X_P = X$, $\leq_P = \leq$, $cxt_P = \emptyset$, $in_P = f$, and $str_P = S$. Its type is the type of S.

The process P' from example 1.2 is a concrete process with $X_{P'} = X'$, $\leq_{P'} = \leq'$, $cxt_{P'} = cxt'$, $in_{P'} = f'$, and $str_{P'} = S'$.

The system P'' from example 1.3 is a concrete process with $X_{P''} = X''$, $\leq_{P''} = \leq''$, $cxt_{P''} = \emptyset$, $in_{P''} = f''$, and $str_{P''} = S''$.

3.2. Definition. Given a concrete process P, by a *cross-section* of P we mean a cross-section of $(X_P, \leq_P)$, and by a *cut* of P we mean a cross-section of $(X_P, \leq_P)$ which is an antichain of $(X_P, \preceq_P)$ with respect to $\preceq_P$. By $crosssections(P)$ and $cuts(P)$ we denote respectively the set of cross-sections and the set of cuts of P. $\square$

3.3. Proposition. For each cut c of a concrete process P, the restrictions of P to the subsets $X_P^-(c) = \{x \in X_P : x \leq_P z \text{ for some } z \in c\}$ and $X_P^+(c) = \{x \in X_P : z \leq_P x \text{ for some } z \in c\}$ are concrete processes, called respectively the *head* and the *tail* of P with respect to c, and written respectively as $head(P, c)$ and $tail(P, c)$. $\square$

A proof is straightforward.

3.4. Definition. Given a concrete process P, by a *splitting* of P we mean a pair $s = (X_P^L, X_P^R)$ of two disjoint subsets X_P^L and X_P^R of X_P such that $X_P^L \cup X_P^R = X_P$, $x' \leq_P x''$ only if x' and x'' are both in one of these subsets, $x' cxt_P x''$ only if x' and x'' are both in one of these subsets, there exist restrictions of str_P to X_P^L and X_P^R, these restrictions are of the same type as P and str_P is their least common extension. $\square$

3.5. Proposition. For each splitting $s = (X_P^L, X_P^R)$ of a concrete process P, the restrictions of P to the subsets X_P^L and X_P^R are concrete processes, called respectively the *left part* and the *right part* of P with respect to s, and written respectively as $left(P, s)$ and $right(P, s)$. $\square$

A proof is straightforward.

3.6. Proposition. The set of cross-sections of a concrete process P is partially ordered by the relation $\sqsubseteq_P$, where $Z' \sqsubseteq_P Z''$ iff for every $z' \in Z'$ there exists $z'' \in Z''$ such that $z' \leq_P z''$. $\square$

A proof is straightforward.

3.7. Proposition. For each concrete process P, the set $crosssections(P)$ with the partial order $\sqsubseteq_P$ is a lattice. The set $cuts(P)$ with the respective restriction of $\sqsubseteq_P$ is a sublattice of this lattice. $\square$

Proof outline.

Let Z' and Z'' be arbitrary cross-sections of P.

We define $Z' \sqcap_P Z''$ as the set of those $z \in Z' \cup Z''$ for which $z \leq_P z'$ for some $z' \in Z'$ and $z \leq_P z''$ for some $z'' \in Z''$.

The set $Z' \sqcap_P Z''$ is an antichain since otherwise it would contain x and y such that $x <_P y$ and in that of the cross-sections Z', Z'' which contains x there would be z such that $y <_P z$, and this would imply $x <_P z$.

Each maximal chain Y has an element $z' \in Z'$ and an element $z'' \in Z''$. As $z' \leq_P z''$ or $z'' \leq_P z'$, one of these elements belongs to $Z' \sqcap_P Z''$. Consequently, $Z' \sqcap_P Z''$ is a maximal antichain and a cross-section.

From the definition of the partial order $\sqsubseteq_P$ it follows that $Z' \sqcap_P Z''$ is the greatest lower bound of Z' and Z''.

Moreover, for Z' and Z'' being cuts also $Z' \sqcap_P Z''$ is a cut since otherwise it would contain x and y such that $x \prec_P y$ and in that of the cuts Z' and Z'' which contains x there would be z such that $y \leq_P z$, and this would imply $x \prec_P z$.

Similarly, the set $Z' \sqcup_P Z''$ of those $z \in Z' \cup Z''$ for which $z' \leq_P z$ for some $z' \in Z'$ and $z'' \leq_P z$ for some $z'' \in Z''$ is the least upper of Z' and Z'', and it is a cut whenever Z' and Z'' are cuts. □

Let P be a concrete process over $U = (W, V, ob)$.

The following definition introduces notions with the aid of which we formulate in section 5 below the concept of independence of processes.

3.8. Definition. By *objects*(P) we mean $ob(in_P(X_P)$, i.e., the set of those objects from V which occur in the concrete process P. By *static*(P) we mean the set of those objects $v \in$ *objects*(P) which are *static* in the sense that their occurrences in P are both minimal and maximal with respect to $\preceq_P$. By *involved*(P) we mean the set of those objects $v \in$ *objects*(P) which are *involved* in P, where v is said to be involed in P if it has in P an occurrence x such that $x <_P y$ for some y or $(x, z) \in cxt_P$ for some z not being minimal with respect to $<_P$. □

To fix the terminology, we say that P is *global* if each cross-section of P contains occurrences of all the objects from V, we say that P is *bounded* if the set of elements of P that are minimal with respect to $\leq_P$ and the set of elements of P that are maximal with respect to $\leq_P$ are cuts; the respective cuts are then called the *origin* and the *end* of P, and they are written as *origin*(P) and *end*(P). Moreover, we say that P is *finitary* if it is bounded and the set of maximal antichains of the partially ordered set $(X_P, \leq_P)$ is finite, and we say that P is K-*dense* if the partial order $\leq_P$ is K-dense.

As in [Wink 80], we can prove the following property.

3.9. Proposition. If P is K-dense and finitary then the relation $<_P - <_P^2$, where $<_P$ is the strict partial order corresponding to $\leq_P$ and $<_P^2$ is the relation $\{(x, y) : x <_P z \text{ and } z <_P y \text{ for some } z\}$, is the union of the family E_P of maximal "rectangular" subsets of X_P^2, that is maximal subsets of the form $X' \times X''$. Moreover, all members of this family are mutually disjoint, their projections on the first factor of $X_P^2 = X_P \times X_P$ are mutually disjoint, and their projections on the second factor of $X_P^2 = X_P \times X_P$ are mutually disjoint. $\square$

For a proof it suffices to consider a maximal chain of cuts of P and segments between contiguous members of this chain.

Members of E_P play the role of indivisible parts of P. We call them *events*. Each event $e \in E_P$ is of the form $pre(e) \times post(e)$ with a $pre(e) \subseteq X_P$ playing the role of the set of those object occurrences which disappear due to e, called the *preset* of e, and $post(e) \subseteq X_P$ playing the role of the set of those object occurrences which appear due to e, called the *postset* of e. The set of those $x \in X_P$ for which $(x, y) \in cxt_P$ for some $y \in post(e)$, written as $ct(e)$, plays the role of the *context* of e. Moreover, $ob(in_P(pre(e))) = ob(in_P(post(e)))$. Thus we obtain a contextual occurrence net in the sense of [MR 95] that specifies not only occurrences of objects from U, but also occurrences of indecomposable actions and how they depend on and affect occurrences of objects.

For example, for the process P in example 1.1 we obtain the contextual occurrence net shown in figure 3.1. The context relation of this net is empty and thus the net is a standard occurrence net. Similarly, for the process P' in example 1.2 we obtain the contextual occurrence net shown in figure 3.2, where the context relation is represented by dotted arrows.

Note that an occurrence of an object in the contextual occurrence net obtained for a concrete process may be a context for all the elements of the postset of an event even though in the original process it may be a context only for some of these elements. This could be excluded by imposing an appropriate restriction on processes, but it would unnecessarily limit the concept.

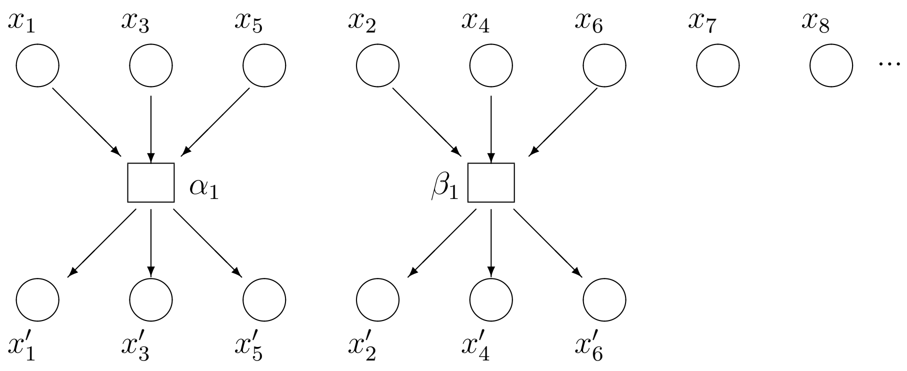


Figure 3.1: The occurrence net for P

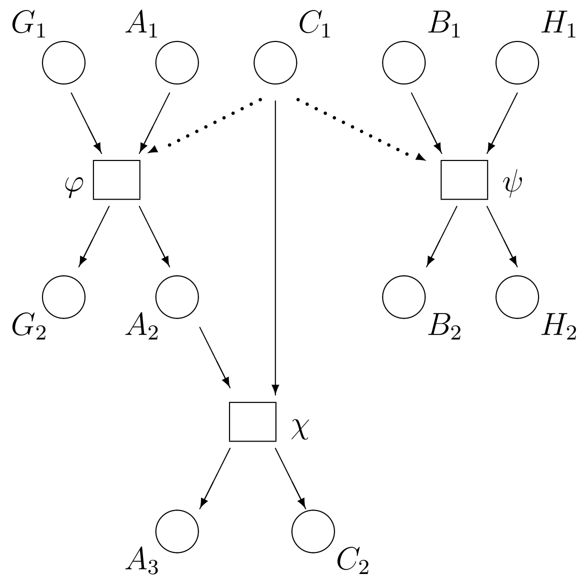


Figure 3.2: The contextual occurrence net for P'

In order to be able to compose processes we consider them up to isomorphisms.

3.10. Definition. An *isomorphism* from a concrete process P to a concrete process P' of the same form is a bijection $b : X_P \to X_{P'}$ such that the images of $\leq_P$, cxt_P, and str_P under the respective mappings induced by b are $\leq_{P'}$, $cxt_{P'}$, and $str_{P'}$, respectively, and $in_{P'}(b(x)) = in_P(x)$. If such an isomorphism exists then the image of $\preceq_P$ under the mapping induced by b is $\preceq_{P'}$ and we say that P and P' are *isomorphic*. $\square$

For concrete processes P and P' that are isomorphic we have $objects(P) = objects(P')$, $static(P) = static(P')$, and $involved(P) = involved(P')$.

3.11. Definition. An *abstract process* is an isomorphism class π of concrete processes. Each concrete process that belongs to π is called an *instance* of π. $\square$

An abstract process corresponding to a concrete process P is written as $[P]$ and we define $objects([P]) = objects(P)$, $static([P]) = static(P)$, and $involved([P]) = involved(P)$. We say that an abstract process π is *global* (resp.: *bounded, finitary,* K-*dense, of type* T) if the instances of π are global (resp.: bounded, finitary, K-dense, of type T).

4 Operations on processes

Following [Wink 80] we can define operations allowing one to construct processes from processes.

Let T be a structure type that admits the least common extensions of its compatible structures. Let $U = (W, V, ob)$ be a universe of objects. By $Proc(T, U)$ we denote the set of all finitary K-dense processes of type T over U. By $gProc(T, U)$ we denote the set of those processes from $Proc(T, U)$ that are global.

In $Proc(T, U)$ there exists the process with the empty set of object instances, called the *empty process* and denoted by 0.

Processes from $Proc(T, U)$ with flow orders reducing to identities, called *process identities*, or *identities*, or *states*, can be identified with structures consisting of a context relation and of a structure of type T on the respective set of object instances.

For each process π from $Proc(T, U)$ there exists a unique process identity, called the *source* or the *domain* of π and written as $dom(\pi)$ (resp.: a unique process identity, called the *target* or the *codomain* of π and written as $cod(\pi)$), whose instance can be obtained from an instance P of π by restricting P to the set $origin(P)$ of minimal elements (resp.: to the set $end(P)$ of maximal elements).

Thus we have two unary operations on processes: the operation $\pi \mapsto dom(\pi)$ of taking the source (the domain), and the operation $\pi \mapsto cod(\pi)$ of taking the target (the codomain). They have the following obvious properties

(A1) $dom(dom(\pi)) = cod(dom(\pi)) = dom(\pi)$,

(A2) $dom(cod(\pi)) = cod(cod(\pi)) = cod(\pi)$.

Other two operations are binary and partial.

One of these operations allows one to combine two processes whenever one of them is a continuation of the other. It can be defined as follows.

4.1. Definition. A process π is said to *consist* of a process π_1 *followed* by a process π_2 if its instance P has a cut c such that $head(P, c)$ is an instance of π_1 and $tail(P, c)$ is an instance of π_2. □

4.2. Proposition. For every two processes π_1 and π_2 such that $cod(\pi_1) = dom(\pi_2)$ there exists a unique process, written as $\pi_1\pi_2$, that consists of π_1 followed by π_2. □

For a proof it suffices to take instances P_1 and P_2 of π_1 and π_2 with $X_{P_1} \cap X_{P_2} = end(P_1) = origin(P_2)$ and the restriction of P_1 to $end(P_1)$ identical with the restriction of P_2 to $origin(P_2)$, to equip $X_{P_1} \cup X_{P_2}$ with the least common extension of the structures of P_1 and P_2, and to consider the structure thus obtained. The acyclicity of the resulting context relation $cxt_{P_1} \cup cxt_{P_2}$ follows from the fact that no element of P_2 which is not in $origin(P_2)$ is in the relation cxt_{P_2} with an element which is in $origin(P_2)$.

4.3. Definition. The operation $(\pi_1, \pi_2) \mapsto \pi_1\pi_2$ is called the *sequential composition*. □

The following properties of the sequential composition follow easily from the definition.

(A3) $\pi_1\pi_2$ is defined whenever $cod(\pi_1) = dom(\pi_2)$, (A4) $(\pi_1\pi_2)\pi_3 = \pi_1(\pi_2\pi_3)$ whenever either side is defined,

(A5) $dom(\pi_1\pi_2) = dom(\pi_1)$ whenever $\pi_1\pi_2$ is defined,

(A6) $cod(\pi_1\pi_2) = cod(\pi_2)$ whenever $\pi_1\pi_2$ is defined,

(A7) $dom(\pi)\pi$ and $\pi\, cod(\pi)$ are defined and $dom(\pi)\pi = \pi\, cod(\pi) = \pi$ for all π,

(A8) if $\pi_1\pi_2$ is an identity then π_1 and π_2 are also identities,

(A9) if $\pi\sigma$ and $\pi\sigma'$ are defined and $\pi\sigma = \pi\sigma'$ then $\sigma = \sigma'$,

(A10) if $\tau\pi$ and $\tau'\pi$ are defined and $\tau\pi = \tau'\pi$ then $\tau = \tau'$.

We recall that the phrase "$\alpha = \beta$ whenever either side is defined" is an abbreviation of the phrase "α is defined if and only if β is defined and $\alpha = \beta$ if α is defined or β is defined".

Another binary partial operation on processes allows one to combine processes on disjoint sets of involved objects. It can be defined as follows.

4.4. Definition. A process π is said to *consist* of two *parallel* processes π_1 and π_2 if its instance P has a splitting s such that $left(P, s)$ is an instance of π_1 and $right(P, s)$ is an instance of π_2. □

4.5. Proposition. If for two processes π_1 and π_2 there exists a process π with an instance P that has a splitting s such that $left(P, s)$ is an instance of π_1 and $right(P, s)$ is an instance of π_2 then such a process is unique. If such a process π exists then we write it as $\pi_1 + \pi_2$ and say that the processes π_1 and π_2 are *parallel.* □

For a proof it suffices to take instances P_1 and P_2 of π_1 and π_2 with $X_{P_1} \cap X_{P_2} = \emptyset$, to equip $X_{P_1} \cup X_{P_2}$ with the least common extension of the structures of P_1 and P_2, and to consider the structure thus obtained.

4.6. Definition. The operation $(\pi_1, \pi_2) \mapsto \pi_1 + \pi_2$ is called the *parallel composition.* □

The following properties of the parallel composition follow easily from the definition.

(A11) $(\pi_1 + \pi_2) + \pi_3 = \pi_1 + (\pi_2 + \pi_3)$ whenever either side is defined,

(A12) $\pi_1 + \pi_2 = \pi_2 + \pi_1$ whenever either side is defined,

(A13) $\pi + 0$ and $0 + \pi$ are always defined and $\pi + 0 = 0 + \pi = \pi$,

(A14) given a family $(\pi_i : i \in \{1, ..., n\})$, where $n \geq 2$, if $\pi_i + \pi_j$ are defined for all $i, j \in \{1, ..., n\}$ such that $i \neq j$ then $\pi_1 + ... + \pi_n$ is defined, (A15) if $\pi + \sigma$ and $\pi + \sigma'$ are defined and $\pi + \sigma = \pi + \sigma'$ then $\sigma = \sigma'$,

(A16) $\pi + \pi$ is defined only for $\pi = 0$.

From (A11) - (A16) we obtain also the following properties.

4.7. Proposition. If $\pi_1 + \pi_2$ is defined and $\pi_1 + \pi_2 = 0$ then $\pi_1 = \pi_2 = 0$. $\square$

For a proof it suffices to notice that the fact that $\pi_1+\pi_2$ is defined and $\pi_1+\pi_2 = 0$ implies $(\pi_1 + \pi_2) + (\pi_1 + \pi_2) = (\pi_1 + \pi_1) + (\pi_2 + \pi_2) = 0$, which implies that $\pi_1 + \pi_1$ and $\pi_2 + \pi_2$ are defined and thus, by (A16), $\pi_1 = 0$ and $\pi_2 = 0$.

4.8. Proposition. The following relation is a partial order on *Proc*(T, U)

$$\pi_1 \leq \pi_2 \text{ iff } \pi_2 = \pi_1 + \xi \text{ for some } \xi. \ \square$$

Proof.
We have $\pi \leq \pi$ since $\pi + 0 = \pi$. If $\pi \leq \sigma$ and $\sigma \leq \tau$ then $\pi \leq \tau$ since $\sigma = \pi + \xi$ and $\tau = \sigma + \eta$ implies $\tau = (\pi + \xi) + \eta = \pi + (\xi + \eta)$. Finally, $\pi \leq \sigma$ and $\sigma \leq \pi$ implies $\pi = \sigma$ since $\sigma = \pi + \xi$ and $\pi = \sigma + \eta$ implies $\xi + \eta = 0$, which implies that $\xi + (\xi + \eta)$ is defined, which implies that $\xi + \xi$ is defined, which implies $\xi = 0$ (by Proposition 4.7) and thus $\sigma = \pi$. $\square$

Taking into account 4.8 we obtain the following further two properties of the parallel composition.

(A17) for all π_1 and π_2 from *Proc*(T, U) there exists the greatest lower bound of π_1 and π_2, written as $\pi_1 \wedge \pi_2$,

(A18) if $\pi_1 + \pi_2$ is defined then, for every σ, $(\pi_1 \wedge \sigma) + (\pi_2 \wedge \sigma)$ is also defined and $(\pi_1 \wedge \sigma) + (\pi_2 \wedge \sigma) = (\pi_1 + \pi_2) \wedge \sigma$.

Moreover, we obtain the following property.

4.9. Proposition. If $\pi_1 + \pi_2$ is defined then $\pi_1 \wedge \pi_2 = 0$. $\square$

Proof.
Let $\pi_1 = (\pi_1 \wedge \pi_2) + \xi$ and $\pi_2 = (\pi_1 \wedge \pi_2) + \eta$. From the fact that $\pi_1 + \pi_2$ is defined we have $\pi_1 + \pi_2 = \xi + \eta + (\pi_1 \wedge \pi_2) + (\pi_1 \wedge \pi_2)$. Consequently, $(\pi_1 \wedge \pi_2) + (\pi_1 \wedge \pi_2)$ is defined and, by (A16), $\pi_1 \wedge \pi_2) = 0$. $\square$

By considering concrete processes and their cuts and splittings we obtain that the introduced operations on processes are related as follows.

(A19) $dom(\pi_1 + \pi_2) = dom(\pi_1) + dom(\pi_2)$ whenever $\pi_1 + \pi_2$ is defined,

(A20) $cod(\pi_1 + \pi_2) = cod(\pi_1) + cod(\pi_2)$ whenever $\pi_1 + \pi_2$ is defined,

(A21) $dom(\pi) = 0$ implies $\pi = 0$ and $cod(\pi) = 0$ implies $\pi = 0$, (A22) if $(\pi_{11}\pi_{12}) + (\pi_{21}\pi_{22})$ is defined then $\pi_{11} + \pi_{21}$, $\pi_{11} + \pi_{22}$, $\pi_{12} + \pi_{21}$, $\pi_{12} + \pi_{22}$ are also defined and $(\pi_{11}\pi_{12}) + (\pi_{21}\pi_{22}) = (\pi_{11} + \pi_{21})(\pi_{12} + \pi_{22})$,

(A23) if $\pi_{11}\pi_{12}$ and $\pi_{21}\pi_{22}$ are defined, and $\pi_{11}+\pi_{21}$ is defined, or $\pi_{11}+\pi_{22}$ is defined, or $\pi_{12}+\pi_{21}$ is defined, or $\pi_{12}+\pi_{22}$ is defined, then $(\pi_{11}\pi_{12})+(\pi_{21}\pi_{22})$ is defined,

(A24) $\pi_1 + \pi_2 = \sigma_1\sigma_2$ implies the existence of unique π_{11}, π_{12}, π_{21}, π_{22} such that $\pi_1 = \pi_{11}\pi_{12}$, $\pi_2 = \pi_{21}\pi_{22}$, $\sigma_1 = \pi_{11} + \pi_{21}$, $\sigma_2 = \pi_{12} + \pi_{22}$.

The properties (A22) and (A23) reflect the fact that processes can be composed in parallel if and only if any of their segments can be composed in parallel and describe how the respective composites are related. The property (A24) means that the representations of a process as results of the two compositions are in a sense "orthogonal".

From (A1) - (A7) we obtain that the set $Proc(T, U)$ equipped with the operations $\pi \mapsto dom(\pi)$, $\pi \mapsto cod(\pi)$, $(\pi_1, \pi_2) \mapsto \pi_1\pi_2$

is a (morphisms-only) category, $CatProc(T, U)$, and that the set $gProc(T, U)$ equipped with the respective restrictions of these operations is a full subcategory, $CatgProc(T, U)$, of $CatProc(T, U)$.

By considering identity processes as objects, we can interpret $CatProc(T, U)$ and $CatgProc(T, U)$ as standard categories.

From (A11) - (A13) we obtain that the set $Proc(T, U)$ equipped with the operations $(\pi_1, \pi_2) \mapsto \pi_1 + \pi_2$ and $\mapsto 0$ is a (partial) commutative monoid. Thus this set equipped with all the introduced operations is a category with an additional structure of a partial commutative monoid. Moreover, (A19), (A20), and (A22), correspond to the conditions relating the operations of standard monoidal categories. Consequently, the set $Proc(T, U)$ equipped with the operations

$$\pi \mapsto dom(\pi),\ \pi \mapsto cod(\pi),\ (\pi_1, \pi_2) \mapsto \pi_1\pi_2,\ (\pi_1, \pi_2) \mapsto \pi_1 + \pi_2,\ \mapsto 0$$

is a partially monoidal category, $pmCatProc(T, U)$, in the following sense (cf. [Wink 82]).

4.10. Definition. A *partially monoidal category* is a set equipped with operations

$$\pi \mapsto dom(\pi),\ \pi \mapsto cod(\pi),\ (\pi_1, \pi_2) \mapsto \pi_1\pi_2,\ (\pi_1, \pi_2) \mapsto \pi_1 + \pi_2,\ \mapsto 0$$

such that (A1) - (A7), (A11) - (A13), (A19), (A20), and (A22) are fulfilled. □

Due to the fact that the carrier of $pmCatProc(T, U)$ is the set of processes of type T over the universe U of objects, this category enjoys also the specific properties (A8) - (A10), (A14) - (A18), (A23), (A24), and several other properties which are described in the sequel.

5 Independence of processes

Let T be a structure type that admits the least common extensions of its compatible structures. Let $U = (W, V, ob)$ be a universe of objects.

For processes from the set $Proc(T, U)$ there are two natural notions of independence corresponding to those introduced in [EK 76] for direct derivations of graph grammars and to those considered in [Wink 03] for processes of Petri nets (cf. also [HR 91]).

5.1. Definition. Processes π_1 and π_2 from $Proc(T, U)$ are said to be *parallel independent* (resp.: *sequential independent*) if $dom(\pi_1) = dom(\pi_2)$ (resp.: $cod(\pi_1) = dom(\pi_2)$) and $involved(\pi_1) \cap involved(\pi_2) \subseteq static(\pi_1) \cap static(\pi_2)$. $\square$

We recall that $static(\pi)$ and $involved(\pi)$ denote respectively the set of static objects and the set of involved objects of a process π, where static and involved objects of an abstract process are static and involved object in the sense of the definition 3.8 of any of its instances.

The inclusion in the definition 5.1 means that the only objects which take part in some changes in both the processes may be those which play only the role of contexts of such changes and occur only in the sources of both the processes. For example, the processes representing the executions of the events φ and ψ of the contextual occurrence net in figure 3.2 in the state corresponding to the occurrences A_1, B_1, C_1, G_1, H_1 of A, B, C, G, H are parallel independent. Indeed,
$involved(\varphi) \cap involved(\psi) = \{A, C, G\} \cap \{B, C, H\} = \{C\}$,
$static(\varphi) \cap static(\psi) = \{B, C, H\} \cap \{A, C, G\} = \{C\}$,
and C occurs only in the sources of both the processes (due to the way of defining the partial order $\preceq$ in (4) of 3.1).

The propositions which follow show that independence of processes can be characterized in categorical terms using the concept of *bicartesian squares*. We recall that a bicartesian square in a category is a diagram $(v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w, v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w)$ in this category, that is a diagram as in figure 5.1, such that $v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w$ is a pushout of $v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w$ and $v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w$ is a pullback of $v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w$.

$$\begin{tikzcd} v \arrow[r, "\pi_2'"] & u' \\ u \arrow[u, "\pi_1"] \arrow[r, "\pi_2"'] & w \arrow[u, "\pi_1'"'] \end{tikzcd}$$

Figure 5.1

5.2. Proposition. For each pair $v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w$ of parallel independent processes there exists a unique pair $v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w$ of processes such that the diagram
$(v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w, v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w)$ is a bicartesian square. $\square$

Proof outline (after [Wink 03]).

There exist instances P_1 and P_2 of π_1 and π_2 with $X_{P_1} \cap X_{P_2} = origin(P_1) = origin(P_2)$ and the restriction of P_1 to $origin(P_1)$ identical with the restriction of P_2 to $origin(P_2)$.

As $involved(\pi_1) \cap involved(\pi_2) \subseteq static(\pi_1) \cap static(\pi_2)$, the set $X_{P_1} \cup X_{P_2}$ equipped with the least common extension of the structures of P_1 and P_2 is an instance of a process π, and there are cuts c_1 and c_2 of P such that $P_1 = head(P, c_1)$, $P_2 = head(P, c_2)$. For $\pi_1' = [tail(P, c_2)]$ and $\pi_2' = [tail(P, c_1)]$ we have $\pi_1\pi_2' = \pi_2\pi_1' = \pi$. Thus we obtain the commutative diagram
$\Delta = (w \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} v, w \overset{\pi_1'}{\rightarrow} u' \overset{\pi_2'}{\leftarrow} v)$.

Suppose that $\pi_1\rho_2 = \pi_2\rho_1 = \sigma$. Then in each instance S of σ there are cuts d_1 and d_2 such that $head(S, d_1)$ is an instance of π_1 and $head(S, d_2)$ is an instance of π_2. Consequently, $head(S, d_1 \sqcup d_2)$ is an instance of π and $tail(S, d_1 \sqcup d_2)$ is an instance of a process ρ such that $\pi\rho = \sigma$. By (A9) such a process is unique. Thus $v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w$ is a pushout of $v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w$.

Suppose that $\xi_1\pi_2' = \xi_2\pi_1' = \tau$. Then in each instance T of τ there are cuts f_1 and f_2 such that $tail(T, f_1)$ is an instance of π_1' and $tail(T, f_2)$ is an instance of π_2'. Consequently, $tail(T, f_1 \sqcap f_2)$ is an instance of π and $head(T, f_1 \sqcap f_2)$ is an instance of a process ξ such that $\xi\pi = \tau$. By (A10) such a process is unique. Thus $v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w$ is a pullback of $v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w$.

Hence Δ is a bicartesian square. The uniqueness of π_1' and π_2' follows from the fact that in the category $CatProc(T, U)$ only identity processes are isomorphisms. $\square$

5.3. Proposition. For each pair $u \overset{\pi_1}{\rightarrow} v \overset{\pi_2'}{\rightarrow} u'$ of sequential independent processes there exists a unique pair $u \overset{\pi_2}{\rightarrow} w \overset{\pi_1'}{\rightarrow} u'$ of processes such that the diagram $(v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w, v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w)$ is a bicartesian square. $\square$

For a proof it suffices to use arguments similar to those in the proof of 5.2.

5.4. Proposition. Let Δ be the diagram $(v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w, v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w)$. If Δ is a bicartesian square, then processes $u \overset{\pi_1}{\rightarrow} v$ and $u \overset{\pi_2}{\rightarrow} w$ are parallel independent, processes $u \overset{\pi_1}{\rightarrow} v$ and $v \overset{\pi_2'}{\rightarrow} u'$ are sequential independent, and processes $u \overset{\pi_2}{\rightarrow} w$ and $w \overset{\pi_1'}{\rightarrow} u'$ are sequential independent. $\square$

Proof outline.

Let P be an instance of $\pi = \pi_1\pi_2' = \pi_2\pi_1'$. As Δ is a bicartesian square, in P there exist cuts c_1 and c_2 such that $origin(P) = c_1 \sqcap c_2$, $end(P) = c_1 \sqcup c_2$, $\pi_1 = [head(P, c_1)]$, $\pi_2' = [tail(P, c_1)]$, $\pi_2 = [head(P, c_2)]$, $\pi_1' = [tail(P, c_2)]$.

In order to prove that π_1 and π_2 are parallel independent suppose the contrary. Then there exists an object v that is involved both in $P_1 = head(P, c_1)$ and in $P_2 = head(P, c_2)$. If v is not static in P_1 then either there exist x,y,z,t such that

$x <_{P_1} y$, $z <_{P_2} t$, $x\ cxt_{P_2}\ t$, which implies $z \prec_P y$ and consequently implies that c_1 cannot be a cut, or there exist x,y,z,t such that $x <_{P_1} y$, $z <_{P_2} t$, $x <_{P_2} t$, which implies that c_2 cannot be a cut. Similarly, if v is not static in P_2 then c_1 cannot be a cut or c_2 cannot be a cut.

Hence π_1 and π_2 must be parallel independent.

Similarly, $u \overset{\pi_1}{\to} v$ and $v \overset{\pi_2'}{\to} u'$ must be sequential independent, and $u \overset{\pi_2}{\to} w$ and $w \overset{\pi_1'}{\to} u'$ must be sequential independent. $\square$

From 5.2 - 5.4 we obtain the following characterization of independence of processes.

5.5. Theorem. Processes of the pair $v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\to} w$ are parallel independent iff there exists a unique pair $v \overset{\pi_2'}{\to} u' \overset{\pi_1'}{\leftarrow} w$ such that $(v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\to} w, v \overset{\pi_2'}{\to} u' \overset{\pi_1'}{\leftarrow} w)$ is a bicartesian square. $\square$

5.6. Theorem. Processes of the pair $u \overset{\pi_1}{\to} v \overset{\pi_2'}{\to} u'$ are sequential independent iff there exists a unique pair $u \overset{\pi_2}{\to} w \overset{\pi_1'}{\to} u'$ such that $(v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\to} w, v \overset{\pi_2'}{\to} u' \overset{\pi_1'}{\leftarrow} w)$ is a bicartesian square. $\square$

The following proposition reflects the fact that independence of processes implies independence of their segments.

5.7. Proposition. Given a bicartesian square $(v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\to} w, v \overset{\pi_2'}{\to} u' \overset{\pi_1'}{\leftarrow} w)$ and a decomposition $u \overset{\pi_1}{\to} v = u \overset{\pi_{11}}{\to} v_1 \overset{\pi_{12}}{\to} v$, there exist a unique decomposition $w \overset{\pi_1'}{\to} u' = w \overset{\pi_{11}'}{\to} w_1 \overset{\pi_{12}'}{\to} u'$, and a unique $v_1 \overset{\pi_2''}{\to} w_1$ such that $(v_1 \overset{\pi_{11}}{\leftarrow} u \overset{\pi_2}{\to} w, v_1 \overset{\pi_2''}{\to} w_1 \overset{\pi_{11}'}{\leftarrow} w)$ and $(v \overset{\pi_{12}}{\leftarrow} v_1 \overset{\pi_2''}{\to} w_1, v \overset{\pi_2'}{\to} u' \overset{\pi_{12}'}{\leftarrow} w_1)$ are bicartesian squares (see figure 5.2). $\square$

$$\begin{tikzcd}
v \arrow[r, "\pi_2'"] & u' \\
v_1 \arrow[u, "\pi_{12}"] \arrow[r, "\pi_2''"] & w_1 \arrow[u, "\pi_{12}'"'] \\
u \arrow[u, "\pi_{11}"] \arrow[uu, "\pi_1"', shift right=2] \arrow[r, "\pi_2"'] & w \arrow[u, "\pi_{11}'"'] \arrow[uu, "\pi_1'", shift left=2]
\end{tikzcd}$$

Figure 5.2

Proof outline (after [Wink 03]).

Let P be an instance of $\pi = \pi_1\pi_2' = \pi_2\pi_1'$. In P there are cuts c_1, c_2, d such that $P_1 = head(P, c_1)$, $P_2 = head(P, c_2)$, $P_1' = tail(P, c_2)$, $P_2' = tail(P, c_1)$, $P_{11} =$

$head(head(P, c_1), d) = head(P, d)$, $P_{12} = tail(head(P, c_1), d)$, are instances of π_1, π_2, π'_1, π'_2, π_{11}, π_{12}, respectively. It suffices to define $\pi''_2 = [tail(head(P, c_2 \sqcup d), d)]$, $\pi'_{11} = [head(P, c_2 \sqcup d)]$, $\pi'_{12} = [tail(P, c_2 \sqcup d)]$. $\square$

6 Categories of processes

Let T be a structure type that admits the least common extensions of its compatible structures. Let $U = (W, V, ob)$ be a universe of objects.

Each process from $CatProc(T, U)$ with instances having no cuts different from their origin or end is *atomic*, or an *atom*, in the sense that it cannot be represented as the result of composing sequentially two processes which are not identities (cf. (A8)).

Following [Wink 80] we can show that each process from $Proc(T, U)$ that is not identity can be obtained by composing sequentially one-event processes. More precisely, we have the following proposition.

6.1. Proposition. Each process $\pi \in Proc(T, U)$ that is not identity can be represented in the form $\pi = \pi_1 ... \pi_n$, where π_1,..., π_n are one-event processes of $Proc(T, U)$. $\square$

For a proof it suffices to consider a maximal chain of cuts of an instance of π.

In general, the representation of a process as the result of composition of atomic processes is not unique. The following proposition makes clear why this may take place.

6.2. Proposition. Let ξ_1, ξ_2, η_1, η_2 be processes from $Proc(T, U)$ such that $\xi_1\xi_2 = \eta_1\eta_2$. Then there exist unique processes σ_1, σ_2, and a unique bicartesian square $(v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w, v \overset{\pi'_2}{\rightarrow} u' \overset{\pi'_1}{\leftarrow} w)$, such that $\xi_1 = \sigma_1\pi_1$, $\xi_2 = \pi'_2\sigma_2$, $\eta_1 = \sigma_1\pi_2$, $\eta_2 = \pi'_1\sigma_2$ (see figure 6.1). $\square$

$$
\begin{tikzcd}
 & & & y \\
 & v \arrow[r, "\pi'_2"] \arrow[urr, "\xi_2"] & u' \arrow[ur, "\sigma_2"] & \\
 & u \arrow[u, "\pi_1"] \arrow[r, "\pi_2"] & w \arrow[u, "\pi'_1"] \arrow[uur, "\eta_2"] & \\
x \arrow[ur, "\sigma_1"] \arrow[urr, "\eta_1"] \arrow[uur, "\xi_1"] & & &
\end{tikzcd}
$$

Figure 6.1

For a proof it suffices to consider an instance S of $\xi_1\xi_2 = \eta_1\eta_2$, its cuts c_1 and c_2 such that $\xi_1 = [head(S, c_1)]$, $\xi_2 = [tail(S, c_1)]$, $\eta_1 = [head(S, c_2)]$, $\eta_2 = [tail(S, c_2)]$, define $\sigma_1 = [head(S, c_1 \sqcap c_2)]$, $\sigma_2 = [tail(S, c_1 \sqcup c_2)]$, $\pi_1 = [head(tail(S, c_1 \sqcap c_2), c_1)]$, $\pi_2' = [head(tail(S, c_1), c_1 \sqcup c_2)]$, $\pi_2 = [head(tail(S, c_1 \sqcap c_2), c_2)]$, $\pi_1' = [head(tail(S, c_2), c_1 \sqcup c_2)]$, and exploit the fact that π_1 and π_2 are parallel independent.

The uniqueness follows from the fact that, due to the cancellation laws (A9) and (A10), $v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w$ is the pullback of $v \overset{\xi_2}{\rightarrow} y \overset{\eta_2}{\leftarrow} w$ and $v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w$ is the pushout of $v \overset{\xi_1}{\leftarrow} x \overset{\eta_1}{\rightarrow} w$.

Note that 6.2 is a generalization of the Levi Lemma for strings and traces (cf. [Maz 88]).

Note that independence of any finite set of processes can be defined as independence of every two different processes from this set. This fact finds expression in a proposition which can be formulated using the concept of a bicartesian n-cube.

Given a graph G, by a *n-cube* in G we mean a subgraph G' of G whose vertices correspond to sequences $(a_1, ..., a_n)$ of binary coordinates $a_i = 0$ or 1, and whose edges lead from one vertex to another whenever one of the coordinates of the latter is obtained from the corresponding coordinate of the former by replacing 0 by 1. The vertex with all coordinates 0 and the edges leading from this vertex to other vertices are termed *initial.* The vertex with all coordinates 1 and the edges leading to this vertex from other vertices are termed *final.* Subgraphs of G' whose all vertices have some of the coordinates identical are m-cubes for the respective $m \leq n$, called *m-faces* of G'.

As categories are also graphs, all these notions apply to categories as well. In particular, one can define a *bicartesian* n-cube in a category C as an n-cube C' in C that commutes and is such that, for each face C'' of C', the family of initial morphisms of C'' extends to a unique limiting cone for the remaining part of C'', and the family of final morphisms of C'' extends to a unique colimiting cone for the remaining part of C''. For example, each bicartesian square is a bicartesian 2-cube.

Taking into account 5.5 and following the line of the proof of 5.2 we obtain the following property of the category $CatProc(T, U)$.

6.3. Proposition. Given a family $\pi = (u \overset{\pi_i}{\rightarrow} v_i : i \in \{1, ..., n\})$ of processes from $Proc(T, U)$, where $n \geq 2$, the existence in $CatProc(T, U)$ for all $i, j \in \{1, ..., n\}$ such that $i \neq j$ of bicartesian squares of the form $(v_i \overset{\pi_i}{\leftarrow} u \overset{\pi_j}{\rightarrow} v_j, v_i \overset{\pi_j'}{\rightarrow} u_{ij}' \overset{\pi_i'}{\leftarrow} v_j)$ implies the existence in $CatProc(T, U)$ of a unique bicartesian n-cube with π being the family of its initial morphisms. □

Combining 6.1 - 6.3 with the results described in section 4 and with 5.7 we obtain the following description of the properties of $CatProc(T, U)$ and $pmCatProc(T, U)$.

6.4. Theorem. The structure $CatProc(T, U)$ is a category that enjoys the properties (A1) - (A10) and the following properties (A25) - (A28):

(A25) for every bicartesian square $(v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w, v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w)$ and every decomposition $u \overset{\pi_1}{\rightarrow} v = u \overset{\pi_{11}}{\rightarrow} v_1 \overset{\pi_{12}}{\rightarrow} v$, there exist a unique decomposition $w \overset{\pi_1'}{\rightarrow} u' = w \overset{\pi_{11}'}{\rightarrow} w_1 \overset{\pi_{12}'}{\rightarrow} u'$, and a unique $v_1 \overset{\pi_2''}{\rightarrow} w_1$ such that the diagrams

$(v_1 \overset{\pi_{11}}{\leftarrow} u \overset{\pi_2}{\rightarrow} w, v_1 \overset{\pi_2''}{\rightarrow} w_1 \overset{\pi_{11}'}{\leftarrow} w)$ and $(v \overset{\pi_{12}}{\leftarrow} v_1 \overset{\pi_2''}{\rightarrow} w_1, v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_{12}'}{\leftarrow} w_1)$ are bicartesian squares,

(A26) for all ξ_1, ξ_2, η_1, η_2 such that $\xi_1\xi_2 = \eta_1\eta_2$ there exist unique σ_1, σ_2, and a unique bicartesian square $(v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w, v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w)$, such that $\xi_1 = \sigma_1\pi_1$, $\xi_2 = \pi_2'\sigma_2$, $\eta_1 = \sigma_1\pi_2$, $\eta_2 = \pi_1'\sigma_2$,

(A27) given a family $\pi = (u \overset{\pi_i}{\rightarrow} v_i : i \in \{1, ..., n\})$, where $n \geq 2$, the existence for all $i, j \in \{1, ..., n\}$ such that $i \neq j$ of bicartesian squares of the form $(v_i \overset{\pi_i}{\leftarrow} u \overset{\pi_j}{\rightarrow} v_j, v_i \overset{\pi_j'}{\rightarrow} u_{ij}' \overset{\pi_i'}{\leftarrow} v_j)$ implies the existence in *CatProc*(T, U) of a unique bicartesian n-cube with π being the family of its initial morphisms.

(A28) every π that is not an identity can be represented in the form $\pi = \pi_1 ... \pi_n$, where π_1,...,π_n are atomic.

The structure *pmCatProc*(T, U) is a partially monoidal category that enjoys the properties (A1) - (A28). □

In the category *CatProc*(T, U) there are subcategories which enjoy all the properties formulated in 6.4. Of this type are *inheriting subcategories*, where by an inheriting subcategory of a category C we mean a subcategory C' which is closed with respect to components of its morphisms, that is such that morphisms α and β of C are also morphisms of C' whenever the composite $\alpha\beta$ is a morphism of C'. This follows easily from the following proposition.

6.5. Proposition. If C' is an inheriting subcategory of a category C that has the properties described in 6.4 then:

(1) each bicartesian square of C whose morphisms are in C' is a bicartesian square in C',

(2) each bicartesian square in C' is a bicartesian square in C. □

The first part of this proposition is immediate. For the second part it suffices to exploit the property (A26) of C and the fact that C' is an inheriting subcategory of C.

Observe that, due to 5.5, 5.6, and 6.5, we can define parallel and sequential independence in *pmCatProc*(T, U), and in inheriting subcategories of *pmCatProc*(T, U), as the existence of an appropriate bicartesian square. Moreover, the definition of independence using bicartesian squares is possible in arbitrary category with the properties specified in 6.4. This suggests that categories of this type or their parts are interesting candidates for new models of distributed systems and justifies the following definition.

6.6. Definition. A *process category* is a category that enjoys the properties (A1) - (A10) and (A25) - (A27). Objects of such a category are called *states*. Morphisms are called *processes*. A *process system*, is a partially monoidal category that enjoys the properties (A1) - (A27). A process category (resp.: process system) is said to be *discrete* if it enjoys also (A28). □

Process categories and process systems are models richer than other ones in the sense that they specify not only states, transitions, and independence of transitions of the modelled systems, but also their processes (runs) and how they compose. Moreover, independence becomes a definable notion, and it can be defined not only for transitions, but also for arbitrary processes. However, the axioms characterizing process categories do not specify completely categories of processes of nontrivial types, and it would not be realistic to expect such a characterization since we cannot expect even a characterization of the respective classes of involved structures.

An important feature of process systems is that in such systems the parallelism of processes implies their independence. This is a direct consequence of the following general fact.

6.7. Theorem. If Π is any process system then for all its processes α and β such that $\alpha + \beta$ is defined the diagram in figure 6.2 is a bicartesian square. □

$$\begin{tikzcd}
cod(\alpha) + dom(\beta) \arrow[r, "cod(\alpha) + \beta"] & cod(\alpha) + cod(\beta) \\
dom(\alpha) + dom(\beta) \arrow[u, "\alpha + dom(\beta)"] \arrow[r, "dom(\alpha) + \beta"'] & dom(\alpha) + cod(\beta) \arrow[u, "\alpha + cod(\beta)"']
\end{tikzcd}$$

Figure 6.2

Proof outline.

Let $\xi_1 = \alpha + dom(\beta)$, $\xi_2 = cod(\alpha) + \beta$, $\eta_1 = dom(\alpha) + \beta$, $\eta_2 = \alpha + cod(\beta)$. Then $\xi_1 \xi_2 = \eta_1 \eta_2$ and by (A26) there exist unique σ_1, σ_2, and a unique bicartesian square $(v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w, v \overset{\pi'_2}{\rightarrow} u' \overset{\pi'_1}{\leftarrow} w)$, such that $\xi_1 = \sigma_1 \pi_1$, $\xi_2 = \pi'_2 \sigma_2$, $\eta_1 = \sigma_1 \pi_2$, $\eta_2 = \pi'_1 \sigma_2$.

As $\xi_1 = \alpha + dom(\beta) = \sigma_1 \pi_1$, by (A24) there exist unique γ and λ such that $\sigma_1 = \gamma + dom(\beta)$, $\pi_1 = \lambda + dom(\beta)$, and $\gamma\lambda = \alpha$. As $\eta_1 = dom(\alpha) + \beta = \sigma_1 \pi_2$, by (A24) there exist unique δ and μ such that $\sigma_1 = dom(\alpha) + \delta$, $\pi_2 = dom(\alpha) + \mu$, and $\delta\mu = \beta$. Consequently, $\gamma + dom(\beta) = dom(\alpha) + \delta$.

By 4.9 we obtain $dom(\beta) = \delta \wedge dom(\beta)$ and $dom(\alpha) = \gamma \wedge dom(\alpha)$, and this implies $\delta = dom(\beta) + \phi$ for some ϕ and $\gamma = dom(\alpha) + \psi$ for some ψ. Together with $\gamma + dom(\beta) = dom(\alpha) + \delta$ this implies by (A15) that $\phi = \psi$.

On the other hand, $\gamma\lambda = \alpha$ implies $dom(\gamma) = dom(\alpha)$ and $\gamma = dom(\alpha) + \psi$

implies $dom(\gamma) = dom(\alpha) + dom(\psi)$. Hence $dom(\psi) = 0$ and by (A21) this implies $\psi = 0$.

Hence $\phi = \psi = 0$, $\gamma = dom(\alpha)$, $\delta = dom(\beta)$, and $\sigma_1 = dom(\alpha) + dom(\beta)$.

Similarly, $\sigma_2 = cod(\alpha) + cod(\beta)$.

Thus the diagram in figure 6.2 reduces to the bicartesian square $(v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w, v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w)$. □

In the rest of the paper we concentrate on process categories.

If we reduce discrete process categories to their states and atoms then we obtain transition systems. If we endow the transition systems thus obtained with the existing in the original process categories information on independence of atomic processes then we obtain structures close to introduced in [WN 95] transition systems with independence and to other similar models as those in [Sh 85] and [Bedn 88].

For the rest of the paper transition systems with independence are defined as follows.

6.8. Definition. A *transition system with independence* is $\Theta = (S, Tran, dom, cod, I)$, where S is a set of *states*, *Tran* is a set of *transitions*, $dom, cod : Tran \to S$ are functions assigning to each transition τ a *source*, $dom(\tau)$, and a *target*, $cod(\tau)$, and I is a binary *independence relation* in *Tran* such that

(1) $(s, \alpha, s')I(u, \beta, u')$ implies $s = u$ or $s' = u$,

(2) $(s, \alpha, s_1)I(s, \beta, s_2)$ implies the existence of unique (s_1, β', u) and (s_2, α', u) such that $(s, \alpha, s_1)I(s_1, \beta', u)$ and $(s, \beta, s_2)I(s_2, \alpha', u)$,

(3) $(s, \alpha, s_1)I(s_1, \beta', u)$ implies the existence of unique (s, β, s_2) and (s_2, α', u) such that $(s, \alpha, s_1)I(s, \beta, s_2)$ and $(s, \beta, s_2)I(s_2, \alpha', u)$,

(4) if $\pi = ((s, \pi_i, s_i) : i \in \{1, ..., n\})$ is a family of transitions such that $(s, \pi_i, s_i)I(s, \pi_j, s_j)$ for all $i, j \in \{1, ..., n\}$ such that $i \neq j$ then in $T(\Pi)$ regarded as a graph there exists a unique n-cube $Q(\pi)$ such that $(u, \alpha, v)I(u, \beta, w)$ and $(u, \beta, w)I(w, \delta, t)$ and $(u, \alpha, v)I(v, \gamma, t)$ for each 2-face of this cube that consists of transitions (u, α, v), (u, β, w), (v, γ, t), (w, δ, t). □

Note that the properties (1) - (3) correspond to the basic axioms characterizing transition systems with independence of [WN 95].

The following proposition describes how descrete process categories define transition systems with independence.

6.9. Proposition. Let Π be a discrete process category with the set S_Π of states and the set A_Π of atomic processes. Let $T(\Pi) = (S, Tran, dom, cod, I)$, where $S = S_\Pi$, *Tran* is the set of triples (s, α, s') such that $\alpha \in A_\Pi$, $s = dom(\alpha)$, $s' = cod(\alpha)$, *dom* and *cod* are the mappings from *Tran* to S defined by $dom(s, \alpha, s') = s$ and $cod(s, \alpha, s') = s'$, and I is the least binary relation in *Tran* such that $(s, \alpha, s_1)I(s, \beta, s_2)$ whenever α and β are parallel independent and $(s, \alpha, s_1)I(s_1, \beta', u)$ whenever α and β' are sequential independent.

Then $T(\Pi)$ is a transition system with independence. $\Box$

The properties (1) - (3) formulated in 6.8 follow from the definition of independence in process categories as the existence of a suitable bicartesian square. The property (4) follows from (A27). Thus we may call $T(\Pi)$ the transition system with independence corresponding to the process category Π.

By defining $Paths(\Theta)$ as the set of paths of Θ, and by defining in the obvious way the source and the target of each path p and the composition of paths p_1 and p_2 such that p_2 follows p_1, we obtain the category of paths of Θ, written as $PATHS(\Theta)$. By defining $\sim_\Theta$ as the least equivalence relation in $Paths(\Theta)$ such that $p_1 \sim_\Theta p_2$ whenever $p_1 = r\alpha\beta s$ and $p_2 = r\beta'\alpha's$ with $\alpha I\beta$ and the unique α' and β' such that $\alpha I\beta'$ and $\beta' I\alpha'$, we obtain a congruence in the category $PATHS(\Theta)$, and the respective quotient category, $RUNS(\Theta)$, called the category of runs of Θ.

6.10. Theorem. For each transition system with independence, Θ, the category of its runs, $RUNS(\Theta)$, is a discrete process category in the sense of 6.6. $\Box$

Proof outline.

A diagram $(v \overset{\pi_1}{\leftarrow} u \overset{\pi_2}{\rightarrow} w, v \overset{\pi_2'}{\rightarrow} u' \overset{\pi_1'}{\leftarrow} w)$ in $RUNS(\Theta)$ is a bicartesian square in iff it consists of independent transitions or by applying decompositions as in figure 5.2 it can be decomposed into bicartesian squares consisting of independent transitions. Taking this into account we obtain (A25). As among the other required properties only (A26) and (A27) are not obvious, it suffices to verify (A26) and (A27).

For (A26) this can be done as follows.

First, it is convenient to fix some terminology. Given two paths p_1 and p_2 such that $p_1 = r\alpha\beta s$ and $p_2 = r\beta'\alpha's$ with $\alpha I\beta$ and the unique α' and β' such that $\alpha I\beta'$ and $\beta' I\alpha'$, we call the pair (p_1, p_2) a *derivation step*. Given a sequence $p_1, ...p_n$ of paths such that each pair (p_i, p_{i+1}) of contiguous paths in this sequence is a derivation step, we call such a sequence a *derivation* of p_n from p_1. Given two paths p_1 and p_2, by the *distance* between p_1 and p_2, written as $d(p_1, p_2)$ we mean the length of the shortest derivation of p_2 from p_1, if such a derivation exists, or $+\infty$ otherwise. Finally, given two representations $\xi_1\xi_2$ and $\eta_1\eta_2$ of a run from $RUNS(\Theta)$, i.e., $\xi_1\xi_2 = \eta_1\eta_2$, by the *distance* between such representations, written as $d(\xi_1, \xi_2; \eta_1, \eta_2)$, we mean the least distance between paths p_1 and p_2 such that $p_1 = p_{11}p_{12}$ for some $p_{11} \in \xi_1$ and $p_{12} \in \xi_2$, and $p_2 = p_{21}p_{22}$ for some $p_{21} \in \eta_1$ and $p_{22} \in \eta_2$.

In order to verify that the equality $\xi_1\xi_2 = \eta_1\eta_2$ implies the existence of σ_1, σ_2, π_1, π_2, π_1', π_2' as in (A26) we proceed by induction on the distance between the representations $\xi_1\xi_2$ and $\eta_1\eta_2$.

If the distance between the representations is 0 then the required property is immediate.

Suppose that the property holds true for the distance not exceeding n and consider ξ_1, ξ_2, η_1, η_2 such that $d(\xi_1, \xi_2; \eta_1, \eta_2) = n + 1$.

In $RUNS(\Theta)$ there exist ζ_1 and ζ_2 such that $d(\xi_1, \xi_2; \zeta_1, \zeta_2) = n$ and

$d(\zeta_1, \zeta_2; \eta_1, \eta_2) = 1$. Consequently, there exist unique τ_1, τ_2, and a unique bicartesian square $(v \overset{\alpha_1}{\leftarrow} u \overset{\alpha_2}{\rightarrow} w, v \overset{\alpha_2'}{\rightarrow} u' \overset{\alpha_1'}{\leftarrow} w)$ such that $\xi_1 = \tau_1\alpha_1$, $\xi_2 = \alpha_2'\tau_2$, $\zeta_1 = \tau_1\alpha_2$, $\zeta_2 = \alpha_1'\sigma_2$.

Now, if one of the equalities $\eta_1 = \zeta_1$, or $\eta_2 = \zeta_2$, holds true then also the other holds true, and we have the required property.

Otherwise, there exist γ_1, γ_2, and indecomposable β_1, β_2, β_1', β_2' such that $\beta_1 I \beta_2$, $\beta_1 I \beta_2'$, $\beta_2 I \beta_1'$, and $\zeta_1 = \gamma_1\beta_1$, $\eta_1 = \gamma_1\beta_2$, $\zeta_2 = \beta_2'\gamma_2$, $\eta_2 = \beta_1'\gamma_2$, as shown in figure 6.3. As $d(\tau_1, \alpha_2; \gamma_1, \beta_1) \leq n$, $d(\alpha_1', \tau_2; \beta_2', \gamma_2) \leq n$, and β_1, β_2, β_1', β_2' are indecomposable, we obtain one of the diagrams in figure 6.4 with all their rectangles being bicartesian squares and the outermost rectangle determining the respective representation of $\xi_1\xi_2 = \eta_1\eta_2$, as required.

A proof of (A27) can be carried out by decomposing the bicartesian squares $(v_i \overset{\pi_i}{\leftarrow} u \overset{\pi_j}{\rightarrow} v_j, v_i \overset{\pi_j'}{\rightarrow} u_{ij}' \overset{\pi_i'}{\leftarrow} v_j)$ into atomic bicartesian squares which correspond to pairs of independent transitions, by exploiting the properties (1) - (4) of the independence relation of Θ and constructing from the atomic bicartesian squares thus obtained the corresponding atomic bicartesian n-cubes, and by combining these n-cubes along their matching $(n-1)$-faces and thus constructing the required bicartesian n-cube for the original runs. $\square$

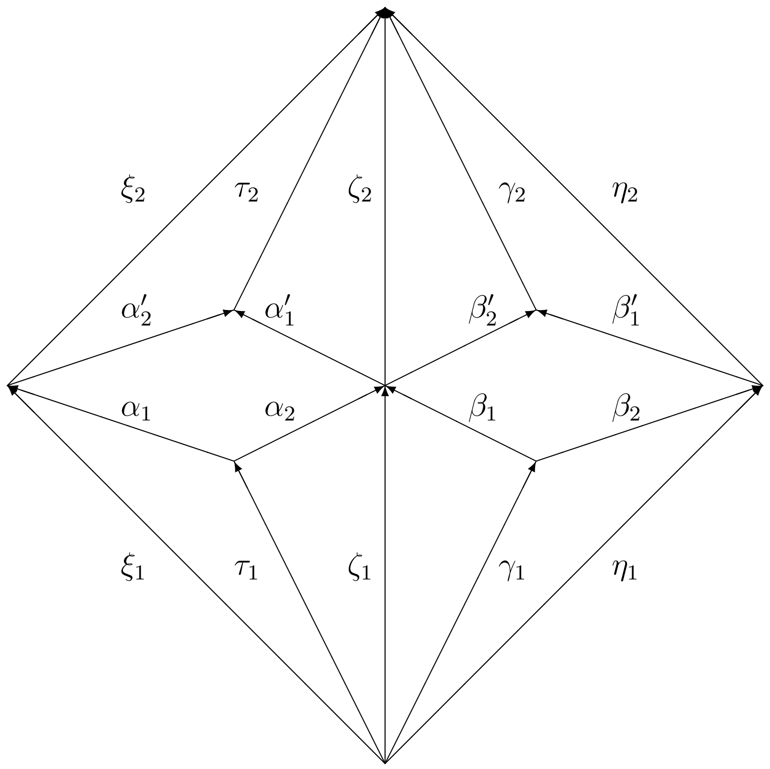


Figure 6.3: A representation of $\xi_1\xi_2 = \eta_1\eta_2$

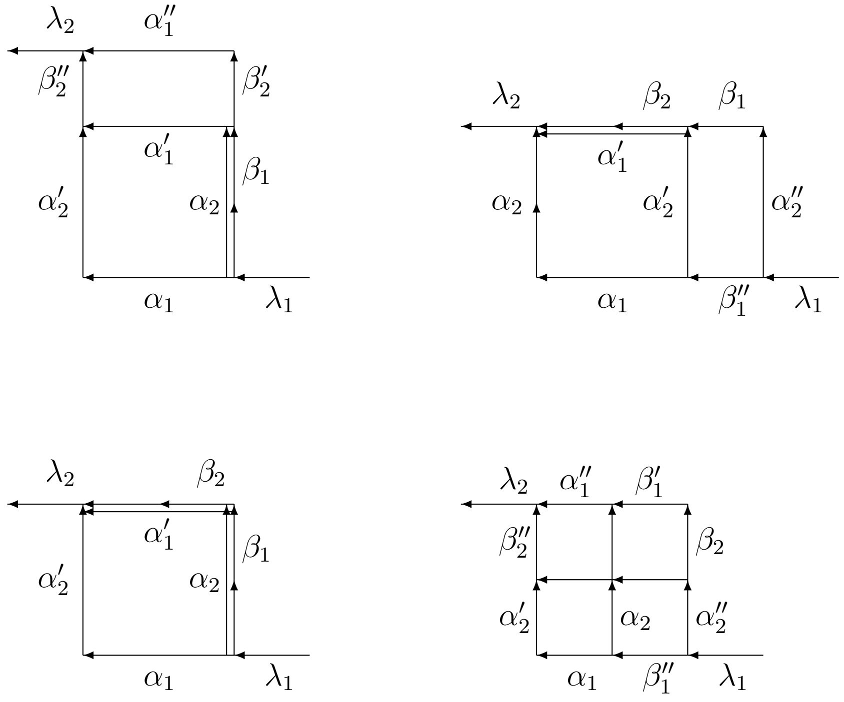


Figure 6.4: The possible more detailed representations of $\xi_1\xi_2 = \eta_1\eta_2$

The relation between transition systems with independence and process categories can be described regarding these structures as objects of categories which can be defined as follows.

6.11. Definition. A *morphism* from a transition system with independence $\Theta = (S, Tran, dom, cod, I)$ to another such a system $\Theta' = (S', Tran', dom', cod', I')$ is a pair (f, g) of mappings $f : S \to S'$ and $g : Tran \to Tran'$ such that $dom'(g(\alpha)) = f(dom(\alpha))$, $cod'(g(\alpha)) = f(cod(\alpha))$, and $\alpha I \beta$ implies $g(\alpha)I'g(\beta)$. $\Box$

By **TI** we denote the category of transition systems with independence and their morphisms.

6.12. Definition. A *morphism* from a process category Π to a process category Π' is a functor from Π to Π' that preserves bicartesian squares. $\Box$

By **P** we denote the category of process categories and their morphisms.

Due to 6.10 we obtain the following result.

6.13. Theorem. Each transition system with independence Θ generates freely the process category $RUNS(\Theta)$ in the sense that each morphism from Θ to the transition system with independence $T(\Pi)$ that corresponds to a process category Π has a unique extension to a morphism from $RUNS(\Theta)$ to Π. $\square$

It is clear that the correspondence $\Theta \mapsto RUNS(\Theta)$ defines a functor $RUNS : \mathbf{TI} \to \mathbf{P}$ and the correspondence $\Pi \mapsto T(\Pi)$ defines a functor $T : \mathbf{P} \to \mathbf{TI}$. Consequently, 6.13 can be formulated as follows.

6.14. Theorem. The functor $RUNS : \mathbf{TI} \to \mathbf{P}$ is the left adjoint of the functor $T : \mathbf{P} \to \mathbf{TI}$. $\square$

7 Discussion and relation to other work

We have defined a process as an activity in a universe of objects that changes states of some objects and establishes or destroys relations among objects. This has been done without relating explicitly a process to a particular system. Instead, we have defined partial operations of composing processes over a universe of objects sequentially and in parallel and the respective categories of processes, we have defined independence of processes, and we have proposed to represent behaviours of systems as parts of categories of processes. Independence of processes can be characterized as the existence in such subcategories of suitable bicartesian squares. Processes in which only some objects are involved can be represented with any degree of locality due to the possibility of composing them in parallel with states of sets of objects that are not involved.

We have described basic properties of categories of processes and we have shown that inheriting subcategories of processes also enjoy these properties. Thus we have obtained a list of properties which can be regarded as axioms characterizing a certain class of models of distributed systems, called process categories, and a certain class of richer models, called process systems, the latter with an explicit representation and characterization of parallel composition. These models generalize asynchronous systems of [Sh 85] and [Bedn 88], and transition systems with independence of [WN 95]. They are richer than the mentioned models in the sense that they allow one to specify not only states, transitions, and independence of transitions of modelled systems, but also their processes (runs), the internal structures of processes, and how processes compose. Moreover, independence becomes a definable notion, and it can be defined not only for transitions, but also for arbitrary processes.

We have shown that by reducing discrete process categories to their objects and atomic morphisms, and by endowing the results of reduction with the existing information on independence, we obtain models close to transition systems with independence of [WN 95]. Finally, we have shown that our transition systems with independence generate freely process categories.

Thus we have created a framework for modelling distributed systems that satisfies the requirements formulated in the introduction.

The relations of our solutions to other to other work are as follows.

We consider systems without a distinguished initial state and represent their runs starting in all the possible states. We have decided to restrict ourselves to such systems in order to get for each system a space of processes that admits the well recognized algebraic structure of a category. This does not limit the possibilities of applications since the behaviours of systems with a distinguished initial state can be represented as subsets of the respective categories of processes that contains only processes starting in the given initial state. Processes in such subsets may be prefixes of other processes, which results in a natural partial order similar to the partial order in configuration structures as those in [GP 95]. In particular, for systems with K-dense finitary processes we can derive from processes the corresponding contextual occurrence nets and next deal the sets of their events as configurations of a configuration structure. However, configuration structures thus obtained are specific since the indeterminism in the underlying sets of processes is fully expressible in terms of state components.

In the case of systems in the form of elementary Petri nets, that is nets whose states are given by sets of conditions, and whose transitions correspond to events which depend on and affect only some conditions, concrete processes of a net can be defined as deterministic occurrence nets, called causal nets, with a homomorphism to the so called safe completion of the original net, and isomorphic concrete processes can be identified (cf. [Wink 03], for details). In the present formulation such processes can be defined as activities in the respective universum of conditions, each condition with two instances corresponding to the states "satisfied" and "not satisfied". This way of defining processes extends easily on contextual Petri nets as those considered in [MR 95] and [BBM 02]. However, the notion of independence of processes is more subtle for contextual Petri nets since processes which share a context may be independent.

In the case of systems represented by Place/Transition Petri nets it is not enough to define concrete processes of a net as causal nets with a homomorphism to this net since the corresponding abstract processes do not contain information sufficient for defining the operations on processes and independence of processes. In [MMS 96] it has been shown that the notion of concatenable decorated processes is what one needs. This notion takes into account the identities of the tokens taking part in a process and it makes possible to define the corresponding operations on processes and independence of processes. An essential feature of this approach is that the identification of tokens in a process is an intrinsic property of this process. In our approach we propose instead to regard processes as running in a fixed universe of objects which may become tokens, and such a universe is external with respect to the considered processes. In the case of processes of Place/Transition nets this solution is less elegant that that in [MMS 96], but in general it may be more universal. For instance, it does not require explicit references to events as in [MMS 96] and thus is more natural for continuous systems.

Processes equipped with graph structures are close to graph processes of [CMR 96], and thus to derivations of graph grammars in the sense of the so called double pushout approach. A grammar generating derivations represented by processes from a given set of processes can be recovered by decomposing processes of this set into atoms and by defining productions as chosen representants of the equivalence classes of atoms thus obtained, where atoms are regarded to be equivalent if elements of their instances are in a bijective correspondence that induces isomorphisms of the structures of their sources and targets. However, our approach is less flexible than the existing standard approach because it limits the set of objects (nodes and edges) which may appear in processes representing derivations of a grammar to a universe that must be fixed in advance. On the other hand, we need not restrict ourselves only to graph structures.

Finally, our methods of representing systems and their processes seem to be well suited for modelling object oriented computations like those that can be programmed in Java or in other similar languages. This is however a subject that requires a special presentation and we do not resume it in the present paper.

Acknowledgements: The author is grateful to the referees for their remarks and suggestions which helped to improve the earlier versions of the paper.

References

[BBM 02] Baldan, P., Bruni, R., Montanari, U., *Pre-nets, read arcs and unfolding: a functorial presentation*, Proceedings of WADT'02, Wirsing, M., Pattison, D., Hennicker, R., (Eds.), Springer LNCS 2755 (2002) 145-164

[Bedn 88] Bednarczyk, M. A., *Categories of Asynchronous Systems*, PhD thesis in Computer Science, University of Sussex, Report no. 1/88 (1988)

[BD 87] Best, E., Devillers, R., *Sequential and Concurrent Behaviour in Petri Net Theory*, Theoret. Comput. Sci. 55 (1987) 87-136

[Bou 57] Bourbaki, N., *Éléments de mathématique, Livre I (Théorie des ensembles), Chapitre 4 (Structures)*, Act. Sci. Ind. 1258, Hermann, Paris, 1957

[BuDe 68] Bucur, I., Deleanu, A., *Introduction to the Theory of Categories and Functors*, John Wiley and Sons Ltd., Lozanna, New York, Sydney, 1968

[Carn 58] Carnap, R., *Introduction to Symbolic Logic and Its Applications, Chapter G: ASs of physics*, Dover Publications, Inc., New York, 1958

[CMR 96] Corradini, A., Montanari, U., Rossi, F., Graph Processes, Fundamenta Informaticae 26 (1996), 241-265

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

[EK 76] Ehrig, H., Kreowski, H. -J., *Parallelism of Manipulations in Multidimensional Information Structures*, in A. Mazurkiewicz (Ed.): Proc. of MFCS'76, Springer LNCS 45 (1976) 284-293

[Eng 91] Engelfriet, J., *Branching Processes of Petri Nets*, Acta Informatica 28 (1991) 575-591

[GP 95] Glabbeek, R., J., van, Plotkin, G., D., *Configuration Structures*, Proceedings of LICS'95, Kozen, D., (Ed.), IEEE Computer Society Press (1995) 199-209

[HR 91] Hoogeboom, H. J., Rozenberg, G., *Diamond Properties of Elementary Net Systems*, Fundamenta Informaticae 14 (1991) 287-300

[Maz 88] Mazurkiewicz, A., *Basic Notions of Trace Theory*, in J. W. de Bakker, W. P. de Roever and G. Rozenberg (Eds.): Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency, Springer LNCS 354 (1988) 285-363

[MMS 96] Meseguer, J., Montanari, U., Sassone, V., *Process versus Unfolding semantics for Place/Transition Nets*, Theoret. Comput. Sci. 153, n. 1-2 (1996) 171-210

[MR 95] Montanari, U., Rossi, F., *Contextual Nets*, Acta Informatica 32 (1995) 545-596

[Petri 77] Petri, C., A., *Non-Sequential Processes*, Interner Bericht ISF-77-5, Gesellschaft fuer Mathematik und Datenverarbeitung, 5205 St. Augustin, Germany (1977)

[RT 86] Rozenberg, G., Thiagarajan, P. S., *Petri Nets: Basic Notions, Structure, Behaviour*, in J. W. de Bakker, W. P. de Roever and G. Rozenberg (Eds.): Current Trends in Concurrency, Springer LNCS 224 (1986) 585-668

[Sh 85] Shields, E. W., *Concurrent Machines*, Computer Journal, vol. 28 (1985) 449-465

[Wink 80] Winkowski, J., *Behaviours of Concurrent Systems*, Theoret. Comput. Sci. 12 (1980) 39-60

[Wink 82] Winkowski, J., *An Algebraic Description of System Behaviours*, Theoret. Comput. Sci. 21 (1982) 315-340

[Wink 03] Winkowski, J., *An Algebraic Characterization of Independence of Petri Net Processes*, Information Processing Letters 88 (2003), 73-81

[WN 95] Winskel, G., Nielsen, M., *Models for Concurrency*, in S. Abramsky, Dov M. Gabbay and T. S. E. Maibaum (Eds.): Handbook of Logic in Computer Science 4 (1995), 1-148