Semantics of Petri nets: a comparison

Gabriel Juhas, Fedor Lehocki, Robert Lorenz

Angaben zur Veröffentlichung / Publication details:

Juhas, Gabriel, Fedor Lehocki, and Robert Lorenz. 2007. "Semantics of Petri nets: a comparison." In *2007 Winter Simulation Conference, 9-12 Dec. 2007, Washington, DC, USA*, edited by Shane G. Henderson, Bahar Biller, Ming-hua Hsieh, and John Shortle, 617–28. Piscataway, NJ: IEEE. https://doi.org/10.1109/wsc.2007.4419655.

SEMANTICS OF PETRI NETS: A COMPARISON

Gabriel Juhás
Fedor Lehocki

Faculty of Electrical Eng. & Information Technology
Slovak University of Technology
Ilkovičova 3, 812 19 Bratislava, SLOVAK REPUBLIC

Robert Lorenz

Department of Computer Science
Catholic University of Eichstätt-Ingolstadt
85072 Eichstätt, GERMANY

ABSTRACT

In this paper, we investigate results on relationship between different semantics of place/transition Petri nets based on labelled partial orders. We also discuss relationships between so called commutative processes representing collective token philosophy and individual process semantics of place/transition nets.

1 INTRODUCTION

The study of concurrency as a phenomenon of systems behavior becomes much attention in recent years, because of an increasing number of distributed systems, multiprocessors systems and communication networks, which are concurrent in their nature. Petri nets are one of the most prominent formalisms for both understanding the concurrency phenomenon on theoretical and conceptual level and for modelling of real concurrent systems in many application areas, see e.g. Jensen (1997), Volume III. Among others, they became an accepted platform for modelling, control and analysis of various kinds of discrete event dynamic systems (Cassandras and Lafortune 1999) including communication networks (see e.g. Billington, Diaz, and Rozenberg 1999) and flexible manufacturing systems (see e.g. Zhou and Di Cesare 1993), and for modelling and analysis of workflow processes (van der Aalst, Desel, and Oberweis 2000, van der Aalst and van Hee 2002), to mention only a few of them.

There are many reasons for that, among others the combination of graphical notation and sound mathematical description, see e.g. Desel and Juhás (2001) for a more detailed discussion.

Very often, the thesis of Carl Adam Petri (Petri 1962) written in the early sixties is cited as the origin of Petri nets. However, Petri did of course not use his own name for defining a class of nets. Moreover, this fundamental work does not contain a definition of those nets that have been called Petri nets later on, i.e. the definition of place/transition Petri nets, which follows the concept of vector addition systems (Karp and Miller 1969) and can be understood as a natural extension of Petri's definition. A place/transition Petri net (shorty a p/t net) is a weighted directed graph with two kinds of nodes, interpreted as places and transitions, such that no arc connects two nodes of the same kind. The arcs are weighted by positive integers. Graphically, places are usually depicted by circles, transitions by rectangles. A local state of a place of a p/t net is given by a nonnegative integer, or graphically by a number of black tokens in a place. The global state of a p/t net, called a marking, is constituted by all local states. Formally, a marking can be given as a multiset of places (called also bag), or as a vector of nonnegative integers. Given a transition t, the set of places, which are connected with t by an arc ingoing to t, is called pre-set of t. The set of places, which is connected with t by an arc outgoing from t, is called the post-set of t. Transitions of a p/t net can occur, changing the state of the net, and their occurrences represent events. Namely, a transition can occur (it is also said that it is enabled to fire), if each place in its pre-set contains at least so much tokens as the weight of the connecting arc. If an enabled transition occurs, then it removes from each place in its pre-set the number of tokens given by the weight of the connecting arc and adds to each place in its post-set the number of the tokens given by the weight of the connecting arc.

As mentioned by Tony Hoare in Hoare (2002): *Different definitions (of semantics) can be safely and consistently used at different times and for different purposes. It is a characteristic of the most successful theories, in mathematics as well as in natural science, that they can be presented in several apparently independent ways, which are in a useful sense provably equivalent.*

In this paper we investigate the relationship between the different variants of semantics of place/transition Petri nets.

There are several ways to describe concurrency in computations of Petri nets. The simplest way is to extend the usual sequential semantics, where sequences of transition

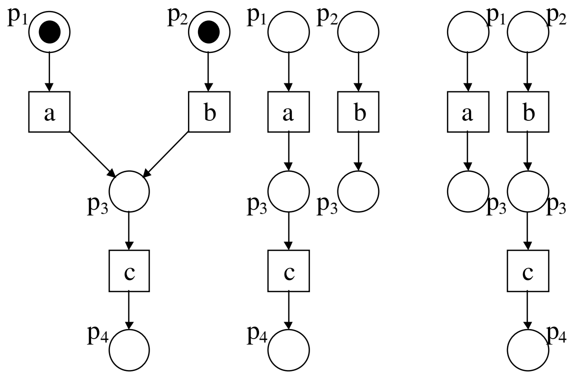

Figure 1: A p/t-net together with two process nets, where transition c occurs after transition a or transition b.

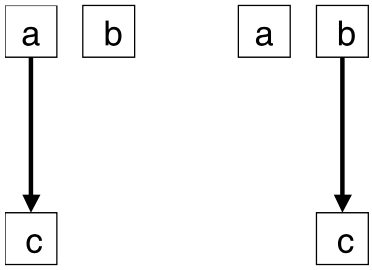

Figure 2: Underlying partial orders of process nets from Figure 1.

occurrences describe computations of nets, to semantics of sequences of steps of transitions which are enabled to occur concurrently in a marking. Here steps are multisets of transitions, which enable to express auto-concurrency.

However it can be easily checked, that in this way concurrent executions of transitions can be expressed only in a restricted way. For illustration, consider the marked p/t net (i.e. a p/t net with a fixed initial marking) given in Figure 1. Here transition c can occur after transition a and concurrently to the sequence ac transition b can occur. This kind of a non-sequential computation cannot be directly expressed by a step sequence.

Therefore, labelled partial order seems to be a better choice to formalize non-sequential semantics (see e.g. Pratt 1986, Grabowski 1981). The above mentioned computation can be described by the left (labelled) partial order in Figure 2. In this concrete computation, it is enough to take a partial order between executed transitions. Because in a computation some transitions can repeatedly occur, in general one has to use labelled partial orders (shortly *LPOs*). Vertices of such an *LPO* (usually called events) are then labelled by transitions of the net. These *LPO*s are called *pomsets* (partially ordered multisets) in Pratt (1986), emphasizing their close relation to partially ordered sets (we have multisets here because the same transition can occur more than once in a pomset, formally represented by two distinct events labelled by the same transition name). *LPO*s are called *partial words* in Grabowski (1981), emphasizing their close relation to words or sequences; the total order of elements in a sequence is replaced by a partial order. Actually, pomsets and partial words do not distinguish isomorphic LPOs, because the order of transition occurrences only depends on the labels.

The natural question arises: which partial orders, labelled by transitions of a marked p/t net, do express non-sequential computations of a given marked p/t net?

The answer has close relationships with the step semantics. Namely, in Grabowski (1981), Kiehn (1988) it is suggested to take as non-sequential computations labelled partial orders satisfying: For each slice of events (i.e. for each maximal set of unordered events) there holds: The concurrent step of events in this slice is enabled to occur in the marking obtained by occurrence of all events smaller than the slice. Technically, a labelled partial order with a special structure can be associated to each step sequence in a natural way. In such a labelled partial order, the slices are exactly the sets of events which represent the occurred steps of transitions. These slices are totally ordered with respect to the ordering given by the step sequence, i.e. the labelled partial orders associated with step sequences are stepwise linearized (every event belongs exactly one slice). The criterion whether a labelled partial order is a computation can be elegantly reformulated using step sequences as follows (Kiehn 1988): A labelled partial order lpo is a computation iff for every slice S there exists a labelled partial order lpo_S of a step sequence with slice S, which includes lpo. Let us call the labelled partial orders which fulfil the above criterion enabled to occur. Observe that per such a definition, labelled partial orders of step sequences, which are enabled to occur in a marked p/t net, are enabled. Moreover, every enabled labelled partial order can be per definition obtained by intersection of a set of labelled partial orders of some enabled step sequences. An important property of enabled labelled partial orders is their closeness w.r.t. sequentialization: if a labelled partial order lpo is enabled, then every labelled partial order which includes lpo is enabled.

Another possibility to express non-sequential computations of p/t nets, is to take processes of Goltz and Reisig (1983), which are special kind of acyclic nets, called occurrence nets, together with a labelling which associates the places (called conditions) and transitions (called events) of the occurrence nets to the places and transitions of the original nets preserving pre- and post-sets in the way illustrated in Figure 1. Processes can be understood as (unbranched) unfoldings of the original nets: every event in the process represents an occurrence of its label in the original net. Abstracting from conditions of process nets, labelled partial orders on events representing transitions are defined. These labelled partial orders, called also runs here, express only the causality between events and tell us in addition which events happen independently, in contrast to enabled labelled partial orders, which can contain some sequentializations of events, which are not causally ordered. A special role play those runs, which are minimal w.r.t. inclusion: they express the minimal causality between events. An important result relating enabled labelled partial orders and minimal runs was proven in Kiehn (1988), Vogler (1992), Vogler (1992):

- Every enabled labelled partial order includes a run.
- Every run is an enabled labelled partial order.
- Therefore, minimal enabled labelled partial orders equal minimal runs.

In contrast to sequential semantics and step semantics, processes distinguish between the history of tokens. An example is shown in Figure 1. The process nets distinguish a token in place p_3 produced by the occurrence of transition a from a token in place p_3 produced by the occurrence of transition b. As a consequence, one occurrence sequence, e.g. abc or one step sequence, e.g. $\{a,b\}\{c\}$ can be a sequentialization of two different processes. The process semantics defined in Goltz and Reisig (1983) is also called individual token semantics. Notice that in the case of the process semantics of safe nets (with at most one token in a place), any occurrence sequence and any step sequence uniquely determine a process. Therefore, in Best and Devillers (1987) the collective token semantics, which does not distinguish between the history of tokens, is introduced. Technically, it is defined using an equivalence relation between processes. Roughly speaking, the equivalence relates processes differing only in permuting (swapping) unordered conditions representing tokens on the same place of the original net. For example, the processes in Figure 1 are equivalent w.r.t. swapping equivalence. The intended meaning of the corresponding equivalence class, called also commutative process, is that c occurs either later than a or later than b. For commutative processes there holds that any occurrence sequence and any step sequence uniquely determine a commutative process.

Thus, we establish the relationships between different kinds of Petri net semantics on a common level given by related sets of LPOs. Namely, given two kinds of semantics represented by a set of labelled partial orders **A** and a set of labelled partial orders **B**, we investigate

- whether the set **B** is a subset of the set **A**, and
- whether each labelled partial order from **A** is a sequentialization of a labelled partial order from **B**.

2 PLACE/TRANSITION NETS

2.1 Mathematical Preliminaries

We use $\mathbb{N}$ to denote the nonnegative integers and $\mathbb{N}^+$ to denote the positive integers. Given two arbitrary sets A and B, the symbol B^A denotes the set of all functions from A to B. Given a function f from A to B and a subset C of A we write $f|_C$ to denote the restriction of f to the set C. The symbol 2^A denotes the power set of a set A. Given a set A, the symbol $|A|$ denotes the cardinality of A and the symbol id_A the identity function on the set A. We write id to denote id_A whenever A is clear from the context. The set of all multisets over a set A is denoted by $\mathbb{N}^A$. The addition of multisets over a finite set A is denoted by $+$. Given two multisets m and m' over A, $m+m'$ is defined by $\forall a \in A : (m+m')(a) = m(a)+m'(a)$. Notice that $(\mathbb{N}^A, +)$ is the free commutative monoid over A. We do not distinguish between a subset $X \subseteq A$ and its characteristic multiset m_X given by $m(x) = 1$ for each $x \in X$ and $m(x') = 0$ for each $x' \in A \setminus X$. Finally, we write as usual $\sum_{a \in A} m(a)a$ to denote the multiset m over A. Given a binary relation $R \subseteq A \times A$ over a set A, the symbol R^+ denotes the transitive closure of R and R^* the reflexive and transitive closure of R.

2.2 Place/Transition Net Definitions

Let us now recall the basic definitions of p/t nets, the partial order based semantics and their algebraic semantics.

Definition 1 (Place/Transition Net)**.** *A* place/transition net *(shortly* p/t net*) N is a quadruple* (P,T,F,W)*, where* (P,T,F) *is a net and* $W : F \to \mathbb{N}^+$ *is a* weight function.

Places and transitions of a net are also called elements of the net. As usual, places are drawn as cycles, transitions as boxes, and the flow relation is expressed using arcs connecting places and transitions.

Let (P,T,F) be a net and $x \in P \cup T$ be an element. The *preset* ${}^\bullet x$ is the set $\{y \in P \cup T \mid (y,x) \in F\}$, and the *post-set* $x^\bullet$ is the set $\{y \in P \cup T \mid (x,y) \in F\}$. Given a set $X \subseteq P \cup T$, this notation is extended as follows:

$$ {}^\bullet X = \bigcup_{x \in X} {}^\bullet x \quad \text{and} \quad X^\bullet = \bigcup_{x \in X} x^\bullet . $$

For technical reasons, we consider only nets in which every transition has a nonempty and finite pre-set and post-set.

Definition 2. *A* place/transition net *(shortly* p/t net*) N is a quadruple* (P,T,F,W)*, where* (P,T,F) *is a net and* $W : F \to \mathbb{N}^+$ *is a* weight function.

We extend the weight function W to pairs of net elements (x,y) satisfying $(x,y) \notin F$ by $W(x,y) = 0$. To avoid confusion, sometimes we will write $N = (P_N, T_N, F_N, W_N)$ to denote $N = (P, T, F, W)$.

A *marking* of a net $N = (P, T, F, W)$ is a function $m : P \to \mathbb{N}$, i.e. a multiset over P. Graphically, a marking is expressed using a respective number of black tokens in each place.

Definition 3 (Marked p/t-net)**.** *A* marked p/t-net *is a pair* (N, m_0)*, where* N *is a p/t-net and* m_0 *is a marking of* N *called initial marking.*

2.3 Sequential Semantics

Definition 4 (Occurrence rule)**.** *Let* $N = (P, T, F, W)$ *be a p/t-net. A transition* $t \in T$ *is* enabled to occur in a marking m *of* N *iff* $m(p) \geq W(p,t)$ *for every place* $p \in {}^\bullet t$*. If a transition* t *is enabled to occur in a marking* m*, then its* occurrence *leads to the new marking* m' *defined by* $m'(p) = m(p) - W(p,t) + W(t,p)$ *for every* $p \in P$*. We write* $m \xrightarrow{t} m'$ *to denote that* t *is enabled to occur in* m *and that its occurrence leads to* m'*.*

Definition 5 (Occurrence sequence, Reachability)**.** *Let* $N = (P, T, F, W)$ *be a p/t-net and* m *be a marking of* N*. A finite sequence of transitions* $\sigma = t_1 \ldots t_n$*,* $n \in \mathbb{N}$ *is called an* occurrence sequence enabled in m and leading to m_n *if there exists a sequence of markings* $m_1, \ldots, m_n$ *such that*

$$m \xrightarrow{t_1} m_1 \xrightarrow{t_2} \ldots \xrightarrow{t_n} m_n.$$

The marking m_n *is said to be* reachable from the marking m*.*
In a marked p/t-net, markings reachable from the initial marking m_0 *are shortly called* reachable markings*.*

3 PARTIAL ORDER BASED SEMANTICS

3.1 Step Semantics

In this section we recall the definition of step semantics for p/t nets. For more details see e.g. Vogler (1992).

The occurrence of single transitions can be extended to the occurrence of multisets of transitions, called steps.

Definition 6 (Step occurrence rule)**.** *Let* $N = (P, T, F, W)$ *be a p/t-net. A multiset of transitions* $s \in \mathbb{N}^T$*, called* step*, is* enabled to occur in a marking m *of* N *iff*

$$\forall p \in P : m(p) \geq \sum_{t \in T} W(p,t) \cdot s(t).$$

If a step s *is enabled to occur in a marking* m*, then its* occurrence *leads to the new marking* m' *defined by*

$$\forall p \in P : m'(p) = m(p) + \sum_{t \in T} (W(t,p) - W(p,t)) \cdot s(t).$$

We write $m \xrightarrow{s} m'$ *to denote that* s *is enabled to occur in* m *and that its occurrence leads to* m'*.*

Definition 7 (Step sequence)**.** *Let* $N = (P, T, F, W)$ *be a p/t-net and* m *be a marking of* N*. A finite sequence of steps (step sequence)* $\sigma = s_1 \ldots s_n$*,* $n \in \mathbb{N}$ *is called a* step sequence enabled in m and leading to m_n *if there exists a sequence of markings* $m_1, \ldots, m_n$ *such that*

$$m \xrightarrow{s_1} m_1 \xrightarrow{s_2} \ldots \xrightarrow{s_n} m_n.$$

Proposition 1. *The marking* m' *is reachable from the marking* m *if and only if there exists a steps sequence enabled in* m *and leading to* m'*.*

3.2 Labeled Partial Orders

In this section we recall the definition of semantics of p/t nets based on labelled partial orders, also known as partial words (Grabowski 1981) or pomsets (Pratt 1986). For the presented results see e.g. Vogler (1992).

Definition 8 (Directed graph, (Labelled) partial order)**.** *A* directed graph *is a pair* $(V, \to)$*, where* V *is a finite* set of nodes *and* $\to \subseteq V \times V$ *is a binary relation over* V *called the* set of arcs*. As usual, given a binary relation* $\to$ *we write* $a \to b$ *to denote* $(a,b) \in \to$*.*

A partial order *is a directed graph* $po = (V, <)$*, where* $<$ *is an irreflexive and transitive binary relation on* V*.*

Two nodes v, v' *of a partial order* $(V, <)$ *are called* independent*, if* $v \not< v'$ *and* $v' \not< v$*. Denote* $co^< \subseteq V \times V$ *the set of all pairs of independent nodes of* V*. A* co-set *in a partial order* $(V, <)$ *is a subset* $S \subseteq V$ *fulfilling:*

$$\forall x, y \in S : x \, co \, y.$$

A slice *is a maximal co-set.*

If $v \, co \, v' \implies v = v'$*, then we say that* $<$ *is* total order*. If the relation* co *is transitive, then we say that* $<$ *is* stepwise linearized.

For a co-set S *of a partial order* $(V, <)$ *and a node* $v \in V \setminus S$ *we write:*

- $v < S$*, if* $v < s$ *for a* $s \in S$*, and*
- $v \, co \, S$*, if* $v \, co \, s$ *for all* $s \in S$*.*

If $<$ *is stepwise linearized, we also write* $S < S'$ *for two slices* S, S' *of* $<$ *whenever* $v < S'$ *for an event* $v \in S$*.*

Given partial orders $po_1 = (V, <_1)$ and $po_2 = (V, <_2)$, we say that po_2 is a sequentialization *of po_1 if $<_1 \subseteq <_2$. If a sequentialization po_2 of po_1 is a total order, we say that po_2 is* linearization *of po_1 and if po_2 is stepwise linearized, we say that it is* step linearization *of po_1.*

A labelled partial order *is a triple $lpo = (V, <, l)$, where $(V, <)$ is a partial order, and l is a* labelling function *on V. If X is a set of labels of lpo, i.e. $l : V \to X$, then for a slice $S \subseteq V$, we define the multiset $|S| \subseteq \mathbb{N}^X$ by*

$$\forall x \in X : |S|(x) = |\{v \in V \mid v \in S \wedge l(v) = x\}|.$$

We use the above notation defined for partial orders also for labelled partial orders.

Two labelled partial orders $(V_1, <_1, l_1), (V_2, <_2, l_2)$ are isomorphic iff there exists a bijection $\gamma : V_1 \to V_2$ between nodes which preserve the partial order relation and the labelling function, i.e. $\forall v_1, v_2 \in V : v_1 <_1 v_2 \iff \gamma(v_1) < \gamma(v_2) \wedge l(v_1) = l(\gamma(v_1))$.

Consider from now a fixed p/t net $N = (P, T, F, W)$. Obviously, the step sequences can be characterized by stepwise linearized labelled partial orders.

Definition 9. *Let $\sigma = s_1 \ldots s_n$ $(n \in \mathbb{N})$ be a sequence of steps from $\mathbb{N}^T$. Then the stepwise linearized labelled partial order $lpo_\sigma = (V, <, l)$ with $l : V \to T$ and with slices $S_1, \ldots S_n$ satisfying $|S_i| = s_i$ and $i < j \Rightarrow S_i < S_j$ for every $i, j \in \{1, \ldots n\}$ is said to be associated to σ.*

As it was observed in Kiehn (1988), the step sequences can be used to define enabledness of labelled partial orders.

Definition 10. *A labelled partial order $lpo = (V, <, l)$ with $l : V \to T$ is said to be* enabled to occur *in a marking m (shortly enabled in m) iff the following statement holds: Every step linearization of lpo is associated to a step sequence enabled to occur in m.*

Directly from the above definitions, we can also observe that the labelling of a linearization of an enabled labelled partial order is an occurrence sequence.

Remark 2. *The sequence of transitions $\sigma = l(v_1) \ldots l(v_n)$ is an occurrence sequence enabled in m and leading to m' if and only if the total order $(\{v_1, \ldots, v_n\}, \prec, l)$ satisfying $\forall i, j \in \{1, \ldots, n\} : i < j \Rightarrow e_i \prec e_j$ is enabled in m and leads to m'. This total order is said to be associated to occurrence sequence σ.*

Looking to the definition of enabledness of a step, we obtain the following proposition:

Proposition 3. *If a labelled partial order $lpo = (V, <, l)$ with $l : V \to T$ is enabled to occur in a marking m then the following statement holds: For every co-set C of $<$ and every $p \in P$:*

$$m(p) + \sum_{v \in V \wedge v < C} (W(l(v), p) - W(p, l(v))) \geq \sum_{v \in C} W(p, l(v))$$

Actually, the definition of enabledness can be reformulated considering only slices of labelled partial orders (for the proof see e.g. Vogler 1992).

Proposition 4. *A labelled partial order $lpo = (V, <, l)$ with $l : V \to T$ is enabled to occur in a marking m if and only if the following statement holds: For every slice S of $<$ there exists a step linearization lpo_S of lpo associated to a step sequence enabled to occur in m, with S being a slice in lpo_S.*

It is easy to observe that enabled labelled partial orders are closed w.r.t. sequentializations.

Proposition 5. *If a labelled partial order is enabled in m and leads to m', then every its sequentialization is enabled in m and leads to m'.*

Special enabled labelled partial orders are those which are minimal w.r.t. inclusion.

Definition 11 (Enabled labelled partial order)**.** *A labelled partial order $lpo = (V, <, l)$ enabled in m is said to be* minimal *iff there exists no labelled partial order $lpo' = (V, <', l)$ enabled in m with $<' \subset <$.*

We say that a set of labelled partial orders enabled to occur in m over the same set of events is compatible if the intersection of the labelled partial orders from this set is a labelled partial order enabled in m.

Definition 12. *Let m be a marking of N. Let $l : V \to T$ be a labelling and let $\lll$ be a set of partial orders on V satisfying: $(V, <, l)$ is a labelled partial order enabled to occur in m for every partial order $<$ from $\lll$.*

If the labelled partial order $lpo = (V, \prec = \cap_{< \in \lll}, l)$ is enabled to occur in m w.r.t. N, then we say that the set of labelled partial orders $\mathscr{C}_N^m\{(V, <, l) \mid < \in \lll\}$ is compatible w.r.t. N and m. *$\mathscr{C}_N^m$ and lpo are said to be* associated *with each other.*

The following proposition says that enabled labelled partial orders can be constructed by intersection of labelled partial orders associated to step sequences. In other words, for every enabled labelled partial order there exists an associated compatible set of labelled partial orders.

Proposition 6. *Let $lpo = (V, \prec, l)$ be a labelled partial order enabled to occur in m w.r.t. N. Then there exists a set X of labelled partial orders compatible w.r.t. N and m such* *that each labelled partial order from X is associated to a step sequence enabled to occur in m and X is associated to lpo.*

Proof. Directly from the enabledness of lpo, for every slice S of $(V,\prec)$ there exists a step sequence of N enabled to occur in m with associated labelled partial order $(V,<,l)$ enabled to occur in m satisfying: S is slice of $(V,<)$ and $\prec \subseteq <_\beta$. Clearly, the intersection of these partial orders equals $\prec$, i.e. the set of these labelled partial orders is compatible w.r.t. N and m and associated to lpo. □

In other words, the previous proposition together with the definition of enabledness says that every enabled labelled partial order can be *constructed* from LPOs associated to step sequences.

3.3 Processes and Runs

Definition 13 (Occurrence net)**.** *An* occurrence net *is a net $O = (B,E,G)$ satisfying:*

1. $|{}^\bullet b|, |b^\bullet| \le 1$ *for every $b \in B$ (places are* unbranched*).*
2. *O is* acyclic*, i.e. the transitive closure G^+ of G is a partial order.*

Places of an occurrence net are called conditions *and transitions of an occurrence net are called* events*.*

The set of conditions of an occurrence net $O = (B,E,G)$ which are minimal (maximal) according to G^+ are denoted by $Min(O)$ $(Max(O))$. Clearly, $Min(O)$ and $Max(O)$ are slices w.r.t. G^+. To avoid confusion, sometimes we will write $O = (B_O,E_O,G_O)$ to denote $O = (B,E,G)$.

Definition 14 (Process)**.** *Let (N,m_0) be a marked p/t-net,with $N = (P,T,F,W)$. A* process *of (N,m_0) is a pair $K = (O,\rho)$, where $O = (B,E,G)$ is an occurrence net and $\rho : B \cup E \to P \cup T$ is a labelling function, satisfying*

1. *$\rho(B) \subseteq P$ and $\rho(E) \subseteq T$.*
2. *$\forall e \in E, \forall p \in P : |\{b \in {}^\bullet e \mid \rho(b) = p\}| = W(p,\rho(e))$ and*
   $\forall e \in E, \forall p \in P : |\{b \in e^\bullet \mid \rho(b) = p\}| = W(\rho(e),p)$.
3. *$\forall p \in P : |\{b \in Min(O) \mid \rho(b) = p\}| = m_0(p)$.*

Two processes $K_1 = ((B_1,E_1,G_1),\rho_1)$ and $K_2 = ((B_2,E_2,G_2),\rho_2)$ are isomorphic (in symbols $K_1 \simeq K_2$) iff there exist bijections $\gamma : B_1 \to B_2, \delta : E_1 \to E_2$ such that $\forall b \in B_1, \forall e \in E_1$:

$$(b,e) \in G_1 \iff (\gamma(b),\delta(e)) \in G_2,$$
$$(e,b) \in G_1 \iff (\delta(e),\gamma(b)) \in G_2,$$
$$\rho_1(b) = \rho_2(\gamma(b)), \rho_1(e) = \rho_2(\delta(e)).$$

Definition 15 (Run)**.** *Let $K = (O,\rho)$ be a process of a marked p/t-net (N,m_0). The labelled partial order $lpo_K = (E, G^+|_{E\times E}, \rho|_E)$ is called* run *of (N,m_0) representing K.*

A run $lpo = (E,<,\rho|_E)$ of (N,m_0) is said to be minimal *iff there exists no other run $lpo' = (E,<',\rho|_E)$ of (N,m_0) with $<' \subset <$.*

It is well known (see e.g. Kiehn 1988, Vogler 1992, Vogler 1992) and easy to show from definition of processes that:

Proposition 7. *Every run of (N,m_0) is enabled in m_0.*

From proposition 5 and proposition 7 follows:

Proposition 8. *If a labelled partial order is a sequentialization of a run of (N,m_0), then it is enabled in m_0.*

The important result completing the relationship between enabled labelled partial orders and runs was proven in Kiehn (1988), Vogler (1992), Vogler (1992).

Theorem 9. *If a labelled partial order is enabled in m_0 in a p/t net N, then it is a sequentialization of a run of the marked p/t net (N,m_0).*

As a consequence we obtain:

Theorem 10. *A run of (N,m_0) is minimal if and only if it is a minimal labelled partial order enabled in m_0.*

The previous theorem (together with the definition of enabledness of LPOs, compatible sets of enabled LPOs, and the fact that (minimal) enabled LPOs can be constructed from LPOs of step sequences) says that every minimal run can be *constructed* from step sequences. In other words, LPOs associated to step sequences gives enough information about minimal runs, i.e. about the necessary causality between events in runs.

4 SUMMARY

In order to summarize the relationships between occurrence sequences, step sequences, enabled LPOs and processes stated above we can compare the related sets of LPOs associated to them.

Given a p/t net N and a marking m_0, let us denote:

- the set of (isomorphism classes of) LPOs associated to occurrence sequences enabled in m_0 by **SEQ**,
- the set of (isomorphism classes of) LPOs associated to step sequences enabled in m_0 by **STEPSEQ**,
- the set of (isomorphism classes of) LPOs enabled to occur in m_0 by **ENABLED** and the set of (isomorphism classes of) minimal LPOs enabled to occur in m_0 by **MINENABLED**
- the set of (isomorphism classes of) runs of (N,m_0) by **RUN** and the set of (isomorphism clsses of) minimal runs of (N,m_0) by **MINRUN**.

. The relationship between these sets w.r.t. set inclusion is given as follows:

$$\mathbf{SEQ} \subseteq \mathbf{STEPSEQ} \subseteq \mathbf{ENABLED},$$

$$\mathbf{RUN} \subseteq \mathbf{ENABLED}.$$

Another important relationship between these sets is the relationship w.r.t. sequentialization. Taking two sets X,Y of LPOs, we denote by $X \Supset Y$ fact that each LPO from X is a sequentialization of an LPO from Y, i.e. each LPO from X includes an LPO from Y:

$$\mathbf{SEQ} \Supset \mathbf{STEPSEQ} \Supset \mathbf{ENABLED} \Supset \mathbf{RUN}.$$

As a consequence:

$$\mathbf{MINENABLED} = \mathbf{MINRUN}.$$

Importantly, enabled labelled partial orders and therefore also minimal runs can be constructed from LPOs associated to step sequences.

5 COMMUTATIVE PROCESSES AND RUNS

As mentioned in the introduction, one occurrence sequence can be in general a sequentialization of two different processes. On the other hand, there are in general many occurrence sequences, which are sequentializations of one process. One may wonder if there exists an equivalence on occurrence sequences and an equivalence on processes, which will respect the relation "being a sequentialization" between occurrence sequences and processes in the following sense: two occurrence sequences are equivalent if and only if they are sequentializations of equivalent processes. This question is investigated in Best and Devillers (1987): For finite occurrence sequences and processes with a finite number of events, which are of interests in this paper, such equivalences are identified and shown to be the finest equivalences with the property.

Definition 16. *Let (N,m_0) be a marked p/t-net, with $N=(P,T,F,W)$. Let $K=(O,\rho)$, be a process of (N,m_0) with $O=(B,E,G)$. Let $b_1,b_2 \in B$, $b_1\, co\, b_2$ and $\rho(b_1)=\rho(b_2)$. We define $G_1=\{(b_1,e) \mid (b_2,e)\in G\}$ and $G_2=\{(b_2,e) \mid (b_1,e)\in G\}$. Then we define $G'=G_1\cup G_2\cup (G\cap(E\times B))\cup(G\cap((B\setminus\{b_1,b_2\})\times E))$. Thus, G' is obtained from G by interchanging arcs from b_1 and b_2. Finally, we define $swap(K,b_1,b_2)=((B,E,G'),\rho)$.*

As it is shown in Best and Devillers (1987):

Theorem 11. *Let $K=(O,\rho)$ be a process of a marked p/t net (N,m_0) with $O=(B,E,G)$. Let $b_1,b_2\in B$, $b_1\, co\, b_2$ and $\rho(b_1)=\rho(b_2)$. Then $swap(K,b_1,b_2)$ is a process of (N,m_0).*

Definition 17. *Let $K_1=((B,E,G),\rho)$ and K_2 be processes of a marked p/t net (N,m_0). Then we define $K_1\equiv_1 K_2$ if there are conditions $b_1,b_2\in B$ such that $b_1\, co\, b_2$ and $\rho(b_1)=\rho(b_2)$ and K_2 is (isomorphic to) $swap(K_1,b_1,b_2)$.*

It is easy to see that $\equiv_1$ is symmetric. Thus, $\equiv_1^*$ is an equivalence relation on processes of (N,m_0).

Definition 18. *The equivalence relation $\equiv_1^*$ on processes of (N,m_0) is called* swapping equivalence. *The equivalence classes of processes w.r.t. the swapping equivalence are called* commutative processes *of (N,m_0).*

The searched equivalence on occurrence sequences is defined in Best and Devillers (1987) using a relation $\equiv_0$ as follows:

Definition 19 (Exchange relation $\equiv_0$ on occurrence sequences)**.** *Let N be a p/t net and m_0 be a marking of N. Let $\sigma_1=t_1\ldots t_{i-1}t_it_{i+1}t_{i+2}\ldots t_n, \sigma_2=t_1\ldots t_{i-1}t_{i+1}t_it_{i+2}\ldots t_n$ be occurrence sequences of N enabled to occur in m_0. Then $\sigma_1\equiv_0\sigma_2$ iff $\sigma=\{t_1\}\ldots\{t_{i-1}\}\{t_i,t_{i+1}\}\{t_{i+2}\}\ldots\{t_n\}$ is a step sequence of N enabled to occur in m_0.*

Again, it is easy to see that $\equiv_0$ is symmetric and therefore $\equiv_0^*$ is an equivalence relation.

Definition 20. *The equivalence relation $\equiv_0^*$ on occurrence sequences of N enabled to occur in m_0 is called* exchange equivalence.

The relationship between exchange equivalence classes and swapping equivalence classes proven in Best and Devillers (1987) says:

Theorem 12. *Let N be a p/t net and m_0 a marking of N. Let σ_1,σ_2 be occurrence sequences of N enabled to occur in m_0 and let K_1,K_2 be processes of (N,m_0) such that the total labelled partial order associated to σ_i is a linearization of* *the run representing K_i ($i \in \{1,2\}$). Then $\sigma_1 \equiv_0^* \sigma_2$ if and only if $K_1 \equiv_1^* K_2$.*

Moreover, as it is proved in Best and Devillers (1987) for the finite case, $\equiv_0^*$ and $\equiv_1^*$ are the finest equivalences which satisfy the above result: These equivalences partition the set of occurrence sequences and processes respectively into finest equivalence classes such that the relation "being a sequentialization" define a bijection on these classes.

As a consequence, these result extend on runs and enabled labelled partial orders. The relation $\frown$ on runs, relating runs if and only if the processes represented by these runs are swapping equivalent, is an equivalence relation. Similarly, the relation $\smile$ on the set of all labelled partial orders enabled in a fixed marking, which relates these labelled partial orders if and only if some of their linearizations are associated to exchange equivalent occurrence sequences, is an equivalence relation.

Definition 21. *Let K_1, K_2 be processes of a marked p/t net (N, m_0) and let lpo_1, lpo_2 be runs representing K_1, K_2, respectively. Then the equivalence relation $\frown$ on runs given by $lpo_1 \frown lpo_1 \iff K_1 \equiv_1^* K_2$ is called swapping equivalence on runs of (N, m_0). The equivalence classes of runs w.r.t. the swapping equivalence are called* commutative runs *of (N, m_0).*

Definition 22. *Let N be a p/t net and m_0 be a marking of N. Let lpo_1, lpo_2 be labelled partial orders enabled to occur in m_0. Then the equivalence relation $\smile$ on the set of all labelled partial orders enabled to occur in m_0 given by:*

- *$lpo_1 \smile lpo_1$ if and only if there exists occurrence sequences σ_1, σ_2 such that the total labelled partial order associated to σ_i is a linearization of lpo_i ($i \in \{1,2\}$) and $\sigma_1 \equiv_0^* \sigma_2$*

is called exchange equivalence on labelled partial orders of N enabled to occur in m_0. The equivalence classes of labelled partial orders enabled to occur in m_0 w.r.t. the exchange equivalence are called commutative labelled partial orders *enabled to occur in m_0.*

From Theorem 12 and the results on relationships between runs and enabled labelled partial orders we get:

Theorem 13. *Let N be a p/t net and m_0 a marking of N. Let lpo_1, lpo_2 be labelled partial orders enabled to occur in m_0 and let lpo'_1, lpo'_2 be runs of (N, m_0) such that lpo_i be a sequentialization of the run lpo'_i ($i \in \{1,2\}$). Then $lpo_1 \smile lpo_2$ if and only if $lpo'_1 \frown lpo'_2$.*

Thus, the equivalences $\smile, \frown$ partition the set of enabled labelled partial orders and runs respectively into commutative enabled labelled partial orders and commutative runs such that the relation "being a sequentialization" define a bijection between them.

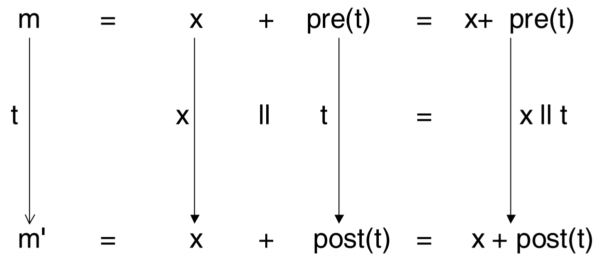

Figure 3: Occurrence of a transition t from a marking m to a marking m' and its interpretation as a concurrent rewriting of the transition t and the marking x.

6 ALGEBRAIC SEMANTICS OF PETRI NETS: AN OVERVIEW

In Meseguer and Montanari (1990) a different approach to non-sequential semantics of Petri nets is proposed: processes terms are generated from elementary rewrite terms using two algebraic operations, namely concurrent and sequential composition.

In this algebraic approach a transition t is understood to be an elementary rewrite term, allowing to replace the marking $pre(t)$ by the marking $post(t)$. Markings are (finite) multisets of places and for concurrent composition of markings usual multiset addition is used. Markings with this addition form a free commutative monoid over the set of places. Any marking consisting of a single place p is understood to be an elementary term rewriting p by p itself.

Process terms are constructed inductively from elementary terms using operators for sequential and for concurrent composition, denoted by ; and $\|$, respectively. Each process term has an associated initial marking and final marking.

Initial and final markings are necessary for sequential composition: Two process terms can be composed sequentially only if the final marking of the first process term coincides with the initial marking of the second one.

For concurrent composition of two process terms, the initial marking of the resulting term is obtained by concurrent composition of the initial markings of the two composed terms, and likewise for the final marking.

For example, the single occurrence of a transition t leading from a marking m to a marking m' can be then understood as a concurrent composition of the elementary term t and the term corresponding to the marking x which satisfies $m = x + pre(t)$ and $m' = x + post(t)$, where $+$ denotes a suitable operation on markings. The non-sequential behavior of a net is given by equivalence classes of process terms defined by a set of equations. As it is observed in Meseguer and Montanari (1990), the set of all markings understood as objects together with the equivalence classes of process terms forming the morphisms rises in a symmetric monoidal category.

In Degano, Meseguer, and Montanari (1996) it is shown that an equivalence class of process terms as defined in Meseguer and Montanari (1990) corresponds to a swapping equivalence class of processes, according to collective token semantics of p/t-nets.

In the individual token approach, where single processes are of interest (Goltz and Reisig 1983), one has to take more sophisticated algebras than usual multisets with addition. For example, in the case of concatenated and strongly concatenated processes (Sassone 1998), the conditions of processes are ordered to remove ambiguous possibilities of concatenation. In case of pre-nets (Sassone 2004), the authors use strings instead of multisets for markings. One of the main reasons why (finite) multisets of places with usual multiset addition (i.e. free commutative monoids) were never used to describe individual token semantics is that process terms belonging to a single equivalence class have never been distinguished.

The question is how we can understand single process terms, or better, what information do they give us. Is it possible to receive single labelled partial orders which are enabled from process terms given by collective token semantics from Meseguer and Montanari (1990), where the simple algebra of multisets of places with multiset addition is used?

In Juhás (2005), we show that it is possible. Namely, we show a general way, how to attach partial orders to process terms and then how to obtain enabled labelled partial orders and therefore also minimal runs.

There is a strong connection between the algebraic process term semantics mentioned above and the partial order based semantics. Each process term α defines a partially ordered set of events representing transition occurrences in an obvious way: an event e_2 *depends on* an event e_1 if the process term α contains a sub-term $\alpha_1;\alpha_2$ such that e_1 occurs in α_1 and e_2 occurs in α_2.

According to the definition of process terms, one can use process terms of a special form to express all step sequences. We will call these terms step sequence terms. But there are also process terms, which give labelled partial orders which have less ordering than the labelled partial orders of any step sequence. An example is the terms of form $(a;c) \parallel b$ expressing the left run from Figure 2.

Thus, we see that a labelled partial order derived from a process term might equal a run. But this is not the case in general because the structure of process terms is too simple for representing any partial order. This fact is illustrated in Figure 4. The run *po* associated to the process in Figure 4 is shown in Figure 5. This enabled labelled partial order expresses the true causalities between the occurred transitions. It is easy to show by induction on the structure of process terms that this partial order

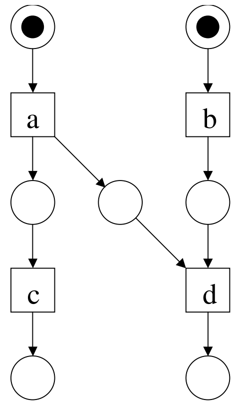

Figure 4: A process net whose associated run cannot be derived directly from any associated process term (see Figure 5).

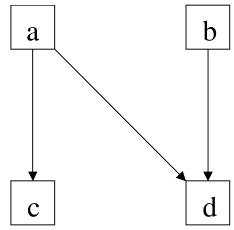

Figure 5: The run *po* corresponding to the process in Figure 4.

cannot be generated by any process term. It is proven in Gischer (1988) that a labelled partial order is generated by concurrent and sequential composition from single element labelled partial orders if and only if it does not contain the shape of so a called N-form (the shape with four nodes connected as in the Figure 5, with absenting arc between the node b and c). As a consequence, we get the characterization of labelled partial orders which are associated to process terms of a p/t net: a labelled partial order is associated with a process term of a p/t net if and only if it is N-free.

The first important question is whether each labelled partial order of a process term is enabled. We show in Juhás (2005) the positive answer using a simple procedure.

Thus, we have that each labelled partial order of a process term is enabled, but not each enabled labelled partial order can be described by a process terms. In other words, with respect to non-sequentiality expressed by single process terms we can see that they are more expressive than single step sequences, but less expressive than enabled labelled partial orders.

But because every enabled labelled partial order can be obtained by intersection of labelled partial orders of enabled step sequences, in the same way it can be obtained from process terms.

6.1 Summary Revisited

To summarize the relationships between the process term semantics of p/t nets and the partial order based semantics of p/t nets we get the following results.

Denote by **TERM** the (isomorphism classes of) LPOs associated to process terms of an algebraic p/t net N' with initial marking m_0 we get:

$$\mathbf{STEPSEQ} \subseteq \mathbf{TERM} \subseteq \mathbf{ENABLED},$$

$$\mathbf{TERM} \Supset \mathbf{ENABLED}.$$

6.2 Process Terms and Commutative Processes

With respect to the result from Degano, Meseguer, and Montanari (1996), which shows that an equivalence class of process terms can determine more than one run, but no run is determined by process terms from different equivalent classes, we show in Juhás (2005), that labelled partial orders of two process terms include runs given by swapping equivalent processes if and only if these process terms are equivalent. We show this one-to-one correspondence between Best-Devillers commutative processes and equivalence classes of process terms using the idea of attaching LPOs to single partial orders and then using the relationship of these LPOs to enabled LPOs and therefore to minimal runs. Let us mention that to show this result in this way is much simpler than the way the similar result was shown in Degano, Meseguer, and Montanari (1996) or recently in Coja-Oghlan and Stehr (2003), where in order to show the correspondence, the processes are first equipped with operation of sequential and concurrent composition and then their correspondence to process terms is investigated. In general, most of the results showing the correspondence between algebraic semantics and process semantics based on process nets is done in this way, for example Degano, Meseguer, and Montanari (1996), Sassone (1996), Sassone (1998), Gadducci and Montanari (1998), Bruni et al. (1998), Sassone (2000), Coja-Oghlan and Stehr (2003), including our previous works (Desel, Juhás, and Lorenz 2000, Desel, Juhás, and Lorenz 2001, Desel, Juhás, and Lorenz 2001a, Juhás, Lorenz, and Šingliar 2003). It is usually quite complicated, because one has to deal with complex definitions of process nets of the related net class. After defining the both compositions one has to prove that the resulting structure is still a process net, that all process nets are generated etc. Further, one has to compare processes equipped with concurrent and sequential composition with process terms which are formally entities of different nature. It is much easier as well as more clear to compare LPOs with LPOs: namely LPOs attached to process terms with (minimal) enabled LPOs and (minimal) runs w.r.t. sequentialization.

REFERENCES

van der Aalst, W. M. P., J. Desel, and A. Oberweis, eds. 2000. *Business Process Management*. Springer, LNCS 1806.

van der Aalst, W. M. P., and K. van Hee. 2002. *Workflow Management, Models Methods and Systems*. Cambridge, Massachusetts: MIT Press.

Best, E., and R. Devillers. 1987. Sequential and concurrent behaviour in Petri net theory. *Theoretical Computer Science* 55:87–136.

Billington, J., M. Diaz, and G. Rozenberg. 1999. *Application of Petri Nets to Communication Networks, Advances in Petri Nets*. Springer, LNCS 1605.

Bruni, R., J. Meseguer, U. Montanari, and V. Sassone. 1998. A comparison of Petri net semantics under the collective token philosophy. *Proc. of ASIAN 1998*, Springer, LNCS 1538, 225–244.

Bruni, R., and V. Sassone. 2000. Algebraic models for contextual nets. *Proc. of ICALP 2000*, Springer, LNCS 1853, 175–186.

Burmeister, P. 2002. *Lecture Notes on Universal Algebra – Many Sorted Partial Algebras*. TU Darmstadt.

Cassandras, C. G., and S. Lafortune. 1999. *Introduction to Discrete Event Systems*. Kluwer.

Corradini, A., P. Baldan, and U. Montanari. 2001. Contextual Petri nets, asymmetric event structures, and processes. *Information and Computation* 171(1):1–49.

Degano, E., J. Meseguer, and U. Montanari. 1996. Axiomatizing the algebra of net computations and processes. *Acta Informatica* 33(7):641–667.

Desel, J., G. Juhás, and R. Lorenz. 2000. Process semantics of Petri nets over partial algebra. In *Proc. of 21th International Conference on Applications and Theory of Petri Nets*, ed. M. Nielsen and D. Simpson. Springer, LNCS 1825, 146–165.

Desel, J., G. Juhás, and R. Lorenz. 2001. Petri nets over partial algebra. In *Unifying Petri Nets*, ed. H. Ehrig, G. Juhás, J. Padberg, G. Rozenberg, Springer, LNCS 2128, 126–172.

Desel, J., G. Juhás, and R. Lorenz. 2001. Unifying Petri nets with restricted occurrence rule using partial algebra. *ENTCS* 44(4).

Desel, J., G. Juhás, and R. Lorenz. 2003. Relating axiomatic and operational semantics of place/transition nets: From process terms to partial orders. *ENTCS* 82(7).

Desel, J., and G. Juhás. 2001. What is a Petri net? In *Unifying Petri Nets*, ed. H. Ehrig, G. Juhás, J. Padberg, G. Rozenberg, Springer, LNCS 2128, 1–25.

Desel, J., and W. Reisig. 1998. Place/transition Petri nets. In *Lectures on Petri nets I: Basic Models*, LNCS 1491, 123–174.

Devillers, R. 1990. The semantics of capacities in P/T Nets. In *Advances in Petri Nets 1989*, LNCS 424, 128–150.

Esparza, J., S. Römer, and W. Vogler. 1996. An improvement of McMillan's unfolding algorithm. In *Proceedings of Tools and Algorithms for the Construction and Analysis of Systems*, Springer, LNCS 1055, 87–106.

Gadducci, F., and U. Montanari. 1998. Axioms for contextual net processes. In *Proc of. ICALP'98*, Springer, LNCS 1443, 296–308.

Gischer, J. L. 1988. The equational theory of pomsets. *Theoretical Computer Science* 61:199–224.

Goltz, U., and W. Reisig. 1983. The non-sequential behaviour of Petri nets. *Information and Control* 57(2-3):125–147.

Goltz, U., and W. Reisig. 1983. Processes of place/transition nets. 10th Int. Springer, LNCS 154, 264–277.

Grabowski, J. 1981. On partial languages. *Fundamenta Informaticae* IV.2, 428–498.

Hanisch, H-M., and A. Lüder. 2000. A signal extension for Petri nets and its use in controller design. *Fundamenta informaticae*.

Hoare, T. 2002. *Unified Theories of Programming*. Springer.

, Jensen, K. 1997. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use I II III*. Springer.

Janicki, R., and M. Koutny. 1993. Structure of concurrency. *Theoretical Computer Science* 112:5–52.

Janicki, R., and M. Koutny. 1995. Semantics of inhibitor nets. *Information and Computations* 123:1–16.

Janicki, R. and M. Koutny. 1999. On causality semantics of nets with priorities. *Fundamenta Informaticae* 38:1–33.

Juhás, G. 1999. Reasoning about algebraic generalisation of Petri nets. In *Proc. of 20th International Conference on Application and Theory of Petri Nets*, ed. S. Donatelli and J. Klein, Springer, LNCS 1639, 324–343.

Juhás, G. 1999. On semantics of Petri nets over partial algebra. In *Proc. of 26th Seminar on Current Trends in Theory and Practice of Informatics SOFSEM'99*, ed. J. Pavelka, G. Tel and M. Bartosek, Springer, LNCS 1725, 408–415.

Juhás, G. 2005. *Are these events independent? It depends!* Habilitation thesis, Catholic University Eichstätt-Ingolstadt.

Juhás, G., R. Lorenz, and T. Šingliar. 2003. On synchronicity and concurrency in Petri nets. In *Proc. of 24th International Conference on Applications and Theory of Petri Nets*, ed. W. M. P. van der Aalst and E. Best, Springer, LNCS 2679, 357–376.

Karp, R. M., and R. E. Miller. 1969. Parallel program schemata. *JCSS* 3(2):147–195.

Kiehn, A. 1988. On the interrelationship between synchronized and non-synchronized behavior of Petri nets. *Journal Inf. Process. Cybern. EIK* 24:3–18.

Kindler, E., and M. Weber. 1998. The dimensions of Petri nets: The Petri net cube. *EATCS Bulletin* (66):155–166.

Kleijn, H. C. M., and M. Koutny. 2000. Process semantics of P/T-Nets with inhibitor arcs. In *Proc. of 21th International Conference on Applications and Theory of Petri Nets*, ed. M. Nielsen and D. Simpson, Springer, LNCS 1825, 261–281.

Kleijn, H. C. M., and M. Koutny. 2004. Process semantics of general inhibitor nets. *Information and Computation* 190(1):18–69.

McMillan, K. L. 1995. A technique of state space search based on unfolding. *Formal Methods in System in System Design* 6(1):45–65.

Meseguer, J., and U. Montanari. 1990. Petri nets are monoids. *Information and Computation* 88(2):105–155.

Montanari, U., and F. Rossi. 1995. Contextual nets. *Acta Informatica* 32(6):545–596.

Coja-Oghlan, A., and M. O. Stehr. 2003. Revisiting the algebra of P/T net processes under the collective token philosophy. *Fundamenta informaticae* 54:151–164.

Padberg, J. 1996. *Abstract Petri nets: Uniform approach and rule-based refinement*. Ph.D. thesis, TU Berlin, Germany.

Padberg, J. 1998. Classification of Petri nets using adjoint functors. *Bulletin of EACTS* No. 66.

Padberg, J., and H. Ehrig. 2001. Parametrized Net Classes: A uniform approach to net classes. In *Unifying Petri Nets*, ed. H. Ehrig, G. Juhás, J. Padberg, and G. Rozenberg, Springer, LNCS 2128, 173–229.

Petri, C. A. 1962. *Kommunikation mit Automaten*. Ph.D. thesis, University of Bonn.

Pinna, G. M., and A. Poigné. 1995. On the nature of events: another perspective in concurrency. *Theoretical Computer Science* 138(2):425–454.

Pratt, V. 1986. Modelling concurrency with partial orders. *Int. Journal of Parallel Programming* 15:33–71.

Rozenberg, G., and J. Engelfriet. 1998. Elementary net systems. In *Lectures on Petri Nets I: Basic Models*, ed. W. Reisig and G. Rozenberg, Springer, LNCS 1491, 12–121.

Sassone, V. 1996. An axiomatization of the algebra of Petri net concatenable processes. *Theoretical Computer Science* 170:277-296.

Sassone, V. 1998. An axiomatization of the category of Petri net computations. *Mathematical Structures in Computer Science* 8:117–151.

Sassone, V. 2004 The algebraic structure of Petri nets. In *Current Trends in Theoretical Computer Science*, ed. G. Paun, G. Rozenberg, and A. Salomaa. World Scientific.

Stehr, M.-O., J. Meseguer, and P. Ölveczky. 2001. Rewriting logic as a unifying framework for Petri nets. In *Unifying Petri Nets*, ed. H. Ehrig, G. Juhás, J. Padberg, and G. Rozenberg, Springer, LNCS 2128, 250–303.

Vogler, W. 1992. Modular construction and partial order semantics of Petri nets. *LNCS* 625.

Vogler, W. 1992. Partial words versus processes: a short comparison. In *Advances in Petri Nets*, Springer, LNCS 609, 292–303.
Vogler, W. 2002. Partial order semantics and read arcs. *Theoretical Computer Science* 286(1):33–63.
Winkowski, J. 1980. Behaviours of concurrent systems. *Theoretical Computer Science* 12:39–60.
Winkowski, J. 1982. An algebraic description of system behaviours. *Theoretical Computer Science* 21:315–340.
Zhou, M. C., and F. Di Cesare. 1993. *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems.* Kluwer.

AUTHOR BIOGRAPHIES

GABRIEL JUHÁS is a vice-dean and an associate professor in the Faculty of Electrical Engineering and Information Technology at the Slovak University of Technology in Bratislava. He has previously held positions in the Department of Applied Computer Science at the Catholic Univeristy of Eichstaett-Ingolstadt, University of Karlsruhe and Technical University of Berlin in Germany. He is an associate editor for the LNCS Transactions on Petri Nets and Other Models of Concurrency published by Springer-Verlag. His research interests include modelling, analysis and synthesis of discrete event systems, in particular theory and applications of Petri nets and other models of concurrency. He has worked also on e-government strategies. His web page can be found via www.elf.stuba.sk/~juhas.

FEDOR LEHOCKI is a researcher at the Faculty of Electrical Engineering and Information Technology of the Slovak University of Technology in Bratislava. His research interests include modeling knowledge bases using Petri Nets a its applications in health care. He is coordinator of several national and EU- funded grants concerned with research in application of medical decision support systems and education of health professionals in medical informatics.

ROBERT LORENZ is an associate professor in the Department of Applied Computer Science at the Catholic University Eichstaett-Ingolstadt. He is an executive editor of the Petri net newsletter of the German Society of Informatics. His research interests include theory of concurrency, in particular methods for synthesis of concurrent systems and related algorithms. His web page can be found via www.infomatik.ku-eichstaett.de.