Abstract Processes of Place/Transition Systems[☆]

Rob van Glabbeek[a,b], Ursula Goltz[c], Jens-Wolfhard Schicke[c]

[a]*NICTA, Sydney, Australia*
[b]*School of Computer Science and Engineering, University of New South Wales, Sydney, Australia*
[c]*Institute for Programming and Reactive Systems, TU Braunschweig, Germany*

Abstract

A well-known problem in Petri net theory is to formalise an appropriate causality-based concept of process or run for place/transition systems. The so-called individual token interpretation, where tokens are distinguished according to their causal history, giving rise to the processes of Goltz and Reisig, is often considered too detailed. The problem of defining a fully satisfying more abstract concept of process for general place/transition systems has so-far not been solved. In this paper, we recall the proposal of defining an abstract notion of process, here called *BD-process*, in terms of equivalence classes of Goltz-Reisig processes, using an equivalence proposed by Best and Devillers. It yields a fully satisfying solution for at least all one-safe nets. However, for certain nets which intuitively have different conflicting behaviours, it yields only one maximal abstract process. Here we identify a class of place/transition systems, called *structural conflict nets*, where conflict and concurrency due to token multiplicity are clearly separated. We show that, in the case of structural conflict nets, the equivalence proposed by Best and Devillers yields a unique maximal abstract process only for conflict-free nets. Thereby BD-processes constitute a simple and fully satisfying solution in the class of structural conflict nets.

Keywords: Petri nets, P/T systems, causal semantics, processes

1. Introduction

The most frequently used class of Petri nets are *place/transition systems* (P/T systems) where places may carry arbitrary many tokens, or a certain maximal number of tokens when adding place capacities. These tokens are usually assumed to be indistinguishable entities. Multiplicities of tokens may represent for instance the number of available resources in a system. The semantics of this type of Petri nets is well-defined with respect to single firings of transitions or finite sets of transitions firing in parallel (*steps*). Sequences of transition firings or of steps are the usual way to define the behaviour of a P/T system. However, these notions of behaviour do not fully reflect the power of Petri nets, as they do not explicitly represent causal dependencies between transition occurrences. If one wishes to interpret P/T systems with a causal semantics, several interpretations of what "causal semantics" should actually mean are available. In the following we give a short overview.

Initially, Petri introduced the concept of a net together with the definition of the firing rule. He defined *condition/event systems*, where—amongst other restrictions—

places (then called conditions) may carry at most one token. For this class of nets, he proposed what is now the classical notion of a *process*, given as a mapping from an *occurrence net* (acyclic net with unbranched places) to the original net [Pet77, GSW80]. A process models a run of the represented system, obtained by choosing one of the alternatives in case of conflict. It records all occurrences of the places and transitions visited during such a run, together with the causal dependencies between them, which are given by the flow relation of the net. A linear-time causal semantics of a condition/event system is thus obtained by associating with a net the set of its processes. Depending on the desired level of abstraction, it may suffice to extract from each process just the partial order of transition occurrences in it. The firing sequences of transitions or steps can in turn be extracted from these partial orders. Nielsen, Plotkin and Winskel extended this to a branching-time semantics by using occurrence nets with forward branched places [NPW81]. These capture all runs of the represented system, together with the branching structure of choices between them.

Goltz and Reisig generalised Petri's notion of process to general P/T systems where multiple tokens may reside on a single place [GR83]. We call this notion of a process *GR-process*. Engelfriet adapted GR-processes by additionally representing choices between alternative behaviours [Eng91], thereby adopting the approach of [NPW81] to P/T systems, although without arc weights. Meseguer,

Sassone and Montanari extended this to cover also arc weights [MMS97].

However, Goltz argued that when abstracting from the identity of multiple tokens residing in the same place, GR-processes do not accurately reflect runs of nets, because if a Petri net is conflict-free it should intuitively have only one run (for there are no choices to resolve), yet it may have multiple GR-processes [Gol86]. This phenomenon is illustrated in Figure 1 in Section 3. A similar argument is made, e.g., in [HKT95].

At the heart of this issue is the question whether multiple tokens residing in the same place should be seen as individual entities, so that a transition consuming just one of them constitutes a conflict, or whether such tokens are indistinguishable, so that taking one is equivalent to taking the other. Van Glabbeek and Plotkin call the former viewpoint the *individual token interpretation* of P/T systems and the latter the *collective token interpretation* [GP95]. A formalisation of these interpretations occurs in [vG05]. A third option, proposed by Vogler, regards tokens only as notation for a natural number stored in each place; these numbers are incremented or decremented when firing transitions, thereby introducing explicit causality between any transitions removing tokens from the same place [Vog91]. The GR-processes, as well as the work of [Eng91, MMS97], fit with the individual token interpretation.

In this paper we continue the line of research of [Gol86, MM88, DMM96, Maz89, HKT95] to formalise a causality-based notion of run of a Petri net that fits the collective token interpretation. As remarked already in [Gol86], *what we need is some notion of an "abstract process"* and *a notion of maximality for abstract processes*, such that *a P/T-system is conflict-free iff it has exactly one maximal abstract process starting at the initial marking.*

A canonical candidate for such a notion of an abstract process is an equivalence class of GR-processes, using an equivalence notion ($\equiv_1^\infty$) proposed by Best and Devillers [BD87]. This equivalence relation is generated by a transformation for changing causalities in GR-processes, called *swapping*, that identifies GR-processes which differ only in the choice which token was removed from a place. Here we call the resulting notion of a more abstract process *BD-process*. In the special case of one-safe P/T systems (where places carry at most one token), or for condition/event systems, no swapping is possible, and a BD-process is just an isomorphism class of GR-processes.

Meseguer and Montanari formalise runs in a net N as morphisms in a category $\mathcal{T}(N)$ [MM88]. In [DMM96] it has been established that these morphisms "coincide with the commutative processes defined by Best and Devillers" (their terminology for BD-processes). Likewise, Hoogers, Kleijn and Thiagarajan represent an abstract run of a net by a *trace*, thereby generalising the trace theory of Mazurkiewicz [Maz95], and remark that "it is straightforward but laborious to set up a 1-1 correspondence between our traces and the equivalence classes of finite processes generated by the swap operation in [Best and Devillers,

1987].". Mazurkiewicz applies a different approach with his *multitrees* [Maz89], which record possible multisets of fired transitions. This approach applies to nets without self-loops only, and we will not consider it in this paper.

Best and Devillers have shown that their equivalence classes of GR-processes are in a bijective correspondence with equivalence classes of firing sequences, generated by swapping two adjacent transitions firings that could have been done in one step. This gives further evidence for the suitability of BD-processes as a formalisation of abstract runs. However, it can be argued that this solution is not fully satisfying for general P/T systems, as we will recall in Section 3 using an example from Ochmański [Och89]. It identifies GR-processes in such a way that certain P/T systems with conflicts have only one maximal BD-process.

In this paper, we analyse the notion of conflict in P/T systems and its interplay with concurrency and causality. We recall the definition of the notion of conflict for P/T systems from [Gol86]. We then define a subclass of P/T systems, called *structural conflict nets*, where the interplay between conflicts and concurrency due to token multiplicities is clearly separated. On this class, the notions of syntactic and semantic conflict are in complete agreement. We show that, for this subclass of P/T systems, the swap transformation by Best and Devillers yields a unique maximal BD-process only for those nets which are conflict-free. The proof of this result is quite involved; it is achieved by using the alternative characterisation of BD-processes by firing sequences from [BD87].

We proceed by defining basic notions for P/T systems in Section 2. In Section 3, we define GR-processes and introduce the swapping equivalence. We give examples and discuss the deficiencies of both GR-processes and BD-processes for a collective token interpretation of general P/T systems. Section 4 recapitulates the concept of conflict in P/T systems and defines structural conflict nets. In Sections 5 and 6, respectively, we introduce the alternative characterisation of BD-processes from [BD87] in terms of equivalence classes of firing sequences and prove in this setting that structural conflict nets with a unique maximal run are indeed conflict-free. Finally we transfer the result to BD-processes in Section 7.

2. Place/Transition Systems

We will employ the following notations for multisets.

Definition 1. Let X be a set.

- A *multiset* over X is a function $A \colon X \to \mathbb{N}$, i.e. $A \in \mathbb{N}^X$.

- $x \in X$ is an *element of* A, notation $x \in A$, iff $A(x) > 0$.

- For multisets A and B over X we write $A \subseteq B$ iff $A(x) \le B(x)$ for all $x \in X$;
  $A \cup B$ denotes the multiset over X with $(A \cup B)(x) := \max(A(x), B(x))$,
  $A + B$ denotes the multiset over X with $(A + B)(x) := A(x) + B(x)$,

$A - B$ is given by $(A - B)(x) := A(x) \overset{\text{\textbullet}}{-} B(x) = \max(A(x) - B(x), 0)$, and

for $k \in \mathbb{N}$ the multiset $k \cdot A$ is given by $(k \cdot A)(x) := k \cdot A(x)$.

- The function $\emptyset : X \to \mathbb{N}$, given by $\emptyset(x) := 0$ for all $x \in X$, is the *empty* multiset over X.

- If A is a multiset over X and $Y \subseteq X$ then $A {\upharpoonright} Y$ denotes the multiset over Y defined by $(A {\upharpoonright} Y)(x) := A(x)$ for all $x \in Y$.

- The cardinality $|A|$ of a multiset A over X is given by $|A| := \sum_{x \in X} A(x)$.

- A multiset A over X is *finite* iff $|A| < \infty$, i.e., iff the set $\{x \mid x \in A\}$ is finite.

Two multisets $A : X \to \mathbb{N}$ and $B : Y \to \mathbb{N}$ are *extensionally equivalent* iff $A {\upharpoonright} (X \cap Y) = B {\upharpoonright} (X \cap Y)$, $A {\upharpoonright} (X \setminus Y) = \emptyset$, and $B {\upharpoonright} (Y \setminus X) = \emptyset$. In this paper we often do not distinguish extensionally equivalent multisets. This enables us, for instance, to use $A \cup B$ even when A and B have different underlying domains. With $\{x, x, y\}$ we will denote the multiset over $\{x, y\}$ with $A(x) = 2$ and $A(y) = 1$, rather than the set $\{x, y\}$ itself. A multiset A with $A(x) \le 1$ for all x is identified with the set $\{x \mid A(x) = 1\}$.

Below we define place/transition systems as net structures with an initial marking. In the literature we find slight variations in the definition of P/T systems concerning the requirements for pre- and postsets of places and transitions. In our case, we do allow isolated places. For transitions we allow empty postsets, but require at least one preplace, thus avoiding problems with infinite self-concurrency. Moreover, following [BD87], we restrict attention to nets of *finite synchronisation*, meaning that each transition has only finitely many pre- and postplaces. Arc weights are included by defining the flow relation as a function to the natural numbers. For succinctness, we will refer to our version of a P/T system as a *net*.

Definition 2.

A *net* is a tuple $N = (S, T, F, M_0)$ where

- S and T are disjoint sets (of *places* and *transitions*),
- $F : (S \times T \cup T \times S) \to \mathbb{N}$ (the *flow relation* including *arc weights*), and
- $M_0 : S \to \mathbb{N}$ (the *initial marking*)

such that for all $t \in T$ the set $\{s \mid F(s, t) > 0\}$ is finite and non-empty, and the set $\{s \mid F(t, s) > 0\}$ is finite.

Graphically, nets are depicted by drawing the places as circles and the transitions as boxes. For $x, y \in S \cup T$ there are $F(x, y)$ arrows (*arcs*) from x to y. When a net represents a concurrent system, a global state of this system is given as a *marking*, a multiset of places, depicted by placing $M(s)$ dots (*tokens*) in each place s. The initial state is M_0. The system behaviour is defined by the possible moves between markings M and M', which take place

when a finite multiset G of transitions *fires*. When firing a transition, tokens on preplaces are consumed and tokens on postplaces are created, one for every incoming or outgoing arc of t, respectively. Obviously, a transition can only fire if all necessary tokens are available in M in the first place. Definition 4 formalises this notion of behaviour.

Definition 3. Let $N = (S, T, F, M_0)$ be a net and $x \in S \cup T$. The multisets $\overset{\text{\textbullet}}{}x$, $x\overset{\text{\textbullet}}{} : S \cup T \to \mathbb{N}$ are given by $\overset{\text{\textbullet}}{}x(y) = F(y, x)$ and $x\overset{\text{\textbullet}}{}(y) = F(x, y)$ for all $y \in S \cup T$. If $x \in T$, the elements of $\overset{\text{\textbullet}}{}x$ and $x\overset{\text{\textbullet}}{}$ are called *pre-* and *postplaces* of x, respectively. These functions extend to multisets $X : S \cup T \to \mathbb{N}$ as usual, by $\overset{\text{\textbullet}}{}X := \Sigma_{x \in S \cup T} X(x) \cdot \overset{\text{\textbullet}}{}x$ and $X\overset{\text{\textbullet}}{} := \Sigma_{x \in S \cup T} X(x) \cdot x\overset{\text{\textbullet}}{}$.

Definition 4. Let $N = (S, T, F, M_0)$ be a net, $G \in \mathbb{N}^T$, G non-empty and finite, and $M, M' \in \mathbb{N}^S$.

G is a *step* from M to M', written $M \overset{G}{\longrightarrow}_N M'$, iff

- $\overset{\text{\textbullet}}{}G \subseteq M$ (G is *enabled*) and
- $M' = (M - \overset{\text{\textbullet}}{}G) + G\overset{\text{\textbullet}}{}$.

We may leave out the subscript N if clear from context. Extending the notion to words $\sigma = t_1 t_2 \dots t_n \in T^*$ we write $M \overset{\sigma}{\longrightarrow} M'$ for

$$\exists M_1, M_2, \dots, M_{n-1}.\ M \overset{\{t_1\}}{\longrightarrow} M_1 \overset{\{t_2\}}{\longrightarrow} M_2 \cdots M_{n-1} \overset{\{t_n\}}{\longrightarrow} M'.$$

When omitting σ or M' we always mean it to be existentially quantified. When $M_0 \overset{\sigma}{\longrightarrow}_N$, the word σ is called a *firing sequence* of N. The set of all firing sequences of N is denoted by $\mathrm{FS}(N)$.

Note that steps are (finite) multisets, thus allowing self-concurrency. Also note that $M \overset{\{t, u\}}{\longrightarrow}$ implies $M \overset{tu}{\longrightarrow}$ and $M \overset{ut}{\longrightarrow}$. We use the notation $t \in \sigma$ to indicate that the transition t occurs in the sequence σ, and $\sigma \le \rho$ to indicate that σ is a prefix of the sequence ρ, i.e. $\exists \mu.\ \rho = \sigma \mu$.

3. Processes of Place/Transition Systems

We now define processes of nets. A (GR-)process is essentially a conflict-free, acyclic net together with a mapping function to the original net. It can be obtained by unwinding the original net, choosing one of the alternatives in case of conflict. The acyclic nature of the process gives rise to a notion of causality for transition firings in the original net via the mapping function. Conflicts present in the original net are represented by one net yielding multiple processes, each representing one possible way to decide the conflicts.

Definition 5.

A pair $P = (\mathcal{N}, \pi)$ is a *(GR-)process* of a net $N = (S, T, F, M_0)$ iff

- $\mathcal{N} = (\mathcal{S}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0)$ is a net, satisfying

  - $\forall s \in \mathcal{S}.\ |\overset{\text{\textbullet}}{}s| \le 1 \ge |s\overset{\text{\textbullet}}{}| \wedge \mathcal{M}_0(s) = \begin{cases} 1 & \text{if } \overset{\text{\textbullet}}{}s = \emptyset \\ 0 & \text{otherwise,} \end{cases}$

3

- $\mathcal{F}$ is acyclic, i.e. $\forall x \in \mathcal{P} \cup \mathcal{T}.\ (x,x) \notin \mathcal{F}^+$, where $\mathcal{F}^+$ is the transitive closure of $\{(t,u) \mid F(t,u) > 0\}$,
- and $\{t \mid (t,u) \in \mathcal{F}^+\}$ is finite for all $u \in \mathcal{T}$.

- $\pi : \mathcal{P} \cup \mathcal{T} \to S \cup T$ is a function with $\pi(\mathcal{P}) \subseteq S$ and $\pi(\mathcal{T}) \subseteq T$, satisfying
  - $\pi(\mathcal{M}_0) = M_0$, i.e. $M_0(s) = |\pi^{-1}(s) \cap \mathcal{M}_0|$ for all $s \in S$, and
  - $\forall t \in \mathcal{T}, s \in S.\ F(s, \pi(t)) = |\pi^{-1}(s) \cap {}^\bullet t| \wedge F(\pi(t), s) = |\pi^{-1}(s) \cap t^\bullet|.$

P is called *finite* if $\mathcal{T}$ (and hence $\mathcal{P}$) are finite.

The conditions for $\mathcal{N}$ ensure that a process is indeed a mapping from an occurrence net as defined in [Pet77, GSW80] to the net N; hence we define processes here in the classical way as in [GR83, BD87] (even though not introducing occurrence nets explicitly).

A process is not required to represent a completed run of the original net. It might just as well stop early. In those cases, some set of transitions can be added to the process such that another (larger) process is obtained. This corresponds to the system taking some more steps and gives rise to a natural order between processes.

Definition 6. Let $P = ((\mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0), \pi)$ and $P' = ((\mathcal{P}', \mathcal{T}', \mathcal{F}', \mathcal{M}_0'), \pi')$ be two processes of the same net.
- P' is a *prefix* of P, notation $P' \leq P$, and P an *extension* of P', iff $\mathcal{P}' \subseteq \mathcal{P}$, $\mathcal{T}' \subseteq \mathcal{T}$, $\mathcal{M}_0' = \mathcal{M}_0$, $\mathcal{F}' = \mathcal{F} \restriction (\mathcal{P}' \times \mathcal{T}' \cup \mathcal{T}' \times \mathcal{P}')$ and $\pi' = \pi \restriction (\mathcal{P}' \times \mathcal{T}')$.
- A process of a net is said to be *maximal* if it has no proper extension.

The requirements above imply that if $P' \leq P$, $(x,y) \in \mathcal{F}^+$ and $y \in \mathcal{P}' \cup \mathcal{T}'$ then $x \in \mathcal{P}' \cup \mathcal{T}'$. Conversely, any subset $\mathcal{T}' \subseteq \mathcal{T}$ satisfying $(t,u) \in \mathcal{F}^+ \wedge u \in \mathcal{T}' \Rightarrow t \in \mathcal{T}'$ uniquely determines a prefix of P.

Two processes $(\mathcal{N}, \pi)$ and $(\mathcal{N}', \pi')$ are *isomorphic* iff there exists an isomorphism ϕ from $\mathcal{N}$ to $\mathcal{N}'$ which respects the process mapping, i.e. $\pi = \pi' \circ \phi$. Here an isomorphism ϕ between two nets $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0)$ and $\mathcal{N}' = (\mathcal{P}', \mathcal{T}', \mathcal{F}', \mathcal{M}_0')$ is a bijection between their places and transitions such that $\mathcal{M}_0'(\phi(s)) = \mathcal{M}_0(s)$ for all $s \in \mathcal{P}$ and $F'(\phi(x), \phi(y)) = F(x,y)$ for all $x, y \in \mathcal{P} \cup \mathcal{T}$.

The notion of a GR-process presented above may fail to capture the intuitive concept of an abstract run of the represented system if the original net (e.g. the one in Figure 1) reaches a marking with multiple tokens in one place. According to Definition 5, such a net N has processes in which multiple places are mapped to the same place in N, thereby representing multiple tokens there. (In Figure 1 each of the two represented processes has two places that map to place 4 in N.) If such a process features a transition whose counterpart in N consumes just one of the tokens present in this place, one needs to choose which of the equivalent places in the process to connect to this transition. Taking different choices gives rise to different
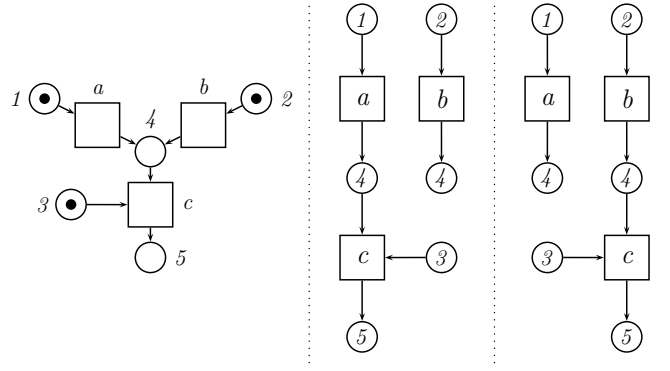

Figure 1: A net N with its two maximal GR-processes. The process mappings are indicated by labels.

processes (such as the two in Figure 1). There is the philosophical question of whether multiple tokens in the same place are distinct entities, that may induce distinct causal relationships,—the individual token interpretation, or together constitute some state of the place—the collective token interpretation [GP95]. In the collective token interpretation of P/T systems, which we take in this paper, the choice which token to remove should not lead to different runs.

As already described in Section 1, a possible strategy to achieve a more abstract notion of process is to introduce a suitable equivalence notion, identifying processes which only differ with respect to the choices of tokens removed from the same place, thus identifying for example the two processes in Figure 1. A candidate for such an equivalence was proposed in [BD87]. It is defined by first introducing a simple transformation on GR-processes; it allows to change causalities in a process by swapping outgoing arrows between places corresponding to the same place in the system net. By reflexive and transitive closure this yields an equivalence notion on finite GR-processes. Slightly deviating from [BD87], we define the equivalence for infinite processes via their finite approximations.

Definition 7. Let $P = ((\mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0), \pi)$ be a process and let $p, q \in \mathcal{P}$ with $(p,q) \notin \mathcal{F}^+ \cup (\mathcal{F}^{-1})^+$ and $\pi(p) = \pi(q)$.

Then $\mathrm{swap}(P, p, q)$ is defined as $((\mathcal{P}, \mathcal{T}, \mathcal{F}', \mathcal{M}_0), \pi)$ with

$$\mathcal{F}'(x,y) = \begin{cases} \mathcal{F}(q,y) & \text{iff } x = p,\ y \in \mathcal{T} \\ \mathcal{F}(p,y) & \text{iff } x = q,\ y \in \mathcal{T} \\ \mathcal{F}(x,y) & \text{otherwise.} \end{cases}$$

Definition 8.
- Two processes P and Q of the same net are *one step swapping equivalent* ($P \approx_{\mathrm{S}} Q$) iff $\mathrm{swap}(P, p, q)$ is isomorphic to Q for some places p and q.
- We write $\approx_{\mathrm{S}}^*$ for the reflexive and transitive closure of $\approx_{\mathrm{S}}$, and $[P]$ for the $\approx_{\mathrm{S}}^*$-equivalence class of a finite process P. The prefix relation $\leq$ between processes is lifted to such equivalence classes by $[P'] \leq [P]$ iff
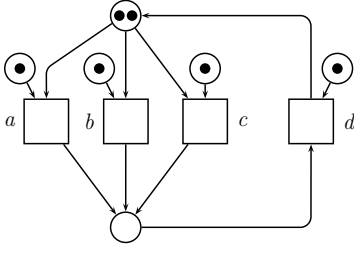
4

Figure 2: A net with only a single process up to swapping equivalence.

$P' \approx_{\mathrm{S}}^{*} Q' \leq Q \approx_{\mathrm{S}}^{*} P$ for some Q', Q.[1]

- Two processes P and Q are *swapping equivalent* ($P \approx_{\mathrm{S}}^{\infty} Q$) iff

$$\downarrow (\{ [\![P']\!] \mid P' \leq P, \ P' \text{ finite}\}) =$$
$$\downarrow (\{ [\![Q']\!] \mid Q' \leq Q, \ Q' \text{ finite}\})$$

where $\downarrow$ denotes prefix-closure under $\leq$.

- We call a $\approx_{\mathrm{S}}^{\infty}$-equivalence class of processes a *BD-process*.

Our definition of $\approx_{\mathrm{S}}^{\infty}$ deviates from the definition of $\equiv_{1}^{\infty}$ from [BD87] to make proofs easier later on. We conjecture however that the two notions coincide.

Unfortunately, with respect to the intuition that different ways to resolve a conflict should give rise to different processes, or the requirement that a P/T system should have exactly one maximal process iff it is conflict-free [Gol86], the swapping equivalence relation is too large. Consider for example the net depicted in Figure 2. In the initial situation only two of the three enabled transitions can fire, which constitutes a conflict. However, there is only one maximal process up to swapping equivalence.

This example has been known for quite some time [Och89, DMM96, BMO09]. However, we are not aware of a solution, i.e. any formalisation of the concept of a run of a net which correctly represents both causality and parallelism of the net, and meets the above requirement. For *one-safe* nets, i.e. nets where places will never carry more than one token in all reachable markings, as well as for condition/event systems, (GR-)processes up to isomorphism are already known to constitute a fully satisfying solution in the above sense. In this paper we will define a larger subclass of P/T systems, including the net of Figure 1, on which BD-processes form a satisfying solution.

4. Conflicts in Place/Transition Systems

Since we desire an abstract notion of process with the property that a P/T-system has exactly one maximal abstract process iff it is conflict-free, it is essential to have a firm definition of conflict. Conflict is a basic notion in the theory of Petri nets, with an easy and clear interpretation

in one-safe P/T systems. Two transitions are in (structural or syntactic) conflict if they share a common preplace [GSW80]. In one-safe nets this coincides with a semantic notion of conflict: if two transitions share a common preplace and they are both enabled, only one of them may fire in the next step. Its firing will (at least temporary) disable the other transition.

In general P/T systems, the situation concerning conflicts is more complicated [Gol86]. First consider the net of Figure 1. In the individual token interpretation of nets, one could postulate that there is a conflict between transition c consuming the token produced by a or by b. Under such an interpretation of conflict, the two maximal GR-processes match our expectations exactly. Under the collective token interpretation used in this paper, on the other hand, we consider this net to be conflict-free, and thus expect only one maximal process.

Next consider the net in Figure 2. In the marking shown, there are three enabled transitions sharing a preplace. Any pair of two of them may fire concurrently (even though they share a preplace), but not all three of them. This ought to be seen as a conflict. Yet, if there would be three tokens in the top-most place of that net, the net would be conflict-free. This shows that conflict is a more involved notion here that may no longer be characterised structurally or syntactically. In [Gol86], it was observed that the traditional definition of conflict covered conflicts between two transitions only, and the following definition of conflict in general P/T systems was proposed.

Definition 9. Let $N=(S,T,F,M_0)$ be a net and $M \in \mathbb{N}^S$.
- A finite, non-empty multiset $G \in \mathbb{N}^T$ is in *(semantic) conflict* in M iff $(\forall t \in G. \ M \xrightarrow{G \restriction \{t\}}) \wedge \neg M \xrightarrow{G}$.
- N is *(semantic) conflict-free* iff no finite, non-empty multiset $G \in \mathbb{N}^T$ is in semantic conflict in any M with $M_0 \longrightarrow M$.

Remark: In a net (S,T,F,M_0) with $S = \{s\}$, $T = \{t,u\}$, $M_0(s) = 1$ and $F(s,t) = F(s,u) = 1$, the multiset $\{t,t\}$ is not enabled in M_0. For this reason the multiset $\{t,t,u\}$ does not count as being in conflict in M_0, even though it is not enabled. However, its subset $\{t,u\}$ is in conflict.

We now propose a class of P/T systems where the structural definition of conflict matches the semantic definition of conflict as given above. We require that two transitions sharing a preplace will never occur both in one step.

Definition 10. Let $N = (S,T,F,M_0)$ be a net.
N is a *structural conflict net* iff $\forall t, u. \ (M_0 \longrightarrow \xrightarrow{\{t,u\}}) \Rightarrow {}^{\bullet}t \cap {}^{\bullet}u = \emptyset$.

Note that this excludes self-concurrency from the possible behaviours in a structural conflict net: as in our setting every transition has at least one preplace, $t = u$ implies ${}^{\bullet}t \cap {}^{\bullet}u \neq \emptyset$. Also note that in a structural conflict net a non-empty, finite multiset G is in conflict in a marking

---

[1]It is not hard to verify that if $P \approx_{\mathrm{S}} Q \leq Q'$ then $P \leq P' \approx_{\mathrm{S}} Q'$ for some process P'. This implies that $\leq$ is a partial order on $\approx_{\mathrm{S}}$-equivalence classes of finite processes. Alternatively, this conclusion will follow from Theorem 4.

M iff G is a set and two distinct transitions in G are in conflict in M.

We will show that the problem outlined in Section 3, namely that the transitive closure of the swapping relation equates processes which we would like to distinguish, vanishes for the class of structural conflict nets. However, the proof of this result is not straightforward. In order to achieve this result, we first introduce the alternative characterisation of BD-processes from [BD87] in terms of an equivalence notion on firing sequences in Section 5 and then characterise problematic situations and prove an appropriate result in terms of this alternative behaviour description in Section 6.

5. Abstract Runs of Place/Transition Systems

This section is largely based on [BD87], but with adapted notation and terminology, and a different treatment of infinite runs. We recall and reformulate these results in order to use them in the following two sections.

The behaviour of a net can be described not only by its processes, but also by its firing sequences. Firing sequences however impose a total order on transition firings, thereby abstracting from information on causal dependence, or concurrency, between transition firings. To retrieve this information we introduce an *adjacency* relation on firing sequences with the intuition that adjacent firing sequences represent the same run of the net. We then define *FS-runs* in terms of the resulting equivalence classes of firing sequences. Adjacency is similar to the idea of Mazurkiewicz traces [Maz95], allowing to exchange concurrent transitions. However, it is based on the semantic notion of concurrency instead of the global syntactic independence relation in trace theory, similar as in the approach of generalising trace theory in [HKT95].

Definition 11. Let $N = (S, T, F, M_0)$ be a net, and $\sigma, \rho \in \mathrm{FS}(N)$.

- σ and ρ are *adjacent*, $\sigma \leftrightarrow \rho$, iff $\sigma = \sigma_1 t u \sigma_2$, $\rho = \sigma_1 u t \sigma_2$ and $M_0 \xrightarrow{\sigma_1} \xrightarrow{\{t,u\}}$.

- We write $\leftrightarrow^*$ for the reflexive and transitive closure of $\leftrightarrow$, and $[\sigma]$ for the $\leftrightarrow^*$-equivalence class of a firing sequence σ.

Note that $\leftrightarrow^*$-related firing sequences contain the same (finite) multiset of transition occurrences. When writing $\sigma \leftrightarrow^* \rho$ we implicitly claim that $\sigma, \rho \in \mathrm{FS}(N)$. Furthermore $\sigma \leftrightarrow^* \rho \wedge \sigma\mu \in \mathrm{FS}(N)$ implies $\sigma\mu \leftrightarrow^* \rho\mu$ for all $\mu \in T^*$.

The following definition introduces the notion of *partial* FS-run which is a formalisation of the intuitive concept of a finite, partial run of a net.

Definition 12. Let N be a net and $\sigma, \rho \in \mathrm{FS}(N)$.

- A *partial FS-run* of N is an $\leftrightarrow^*$-equivalence class of firing sequences.

- A partial FS-run $[\sigma]$ is a *prefix* of another partial FS-run $[\rho]$, notation $[\sigma] \leq [\rho]$, iff $\exists\mu.\ \sigma \leq \mu \leftrightarrow^* \rho$.

Note that $\rho' \leftrightarrow^* \rho \leq \mu \leftrightarrow^* \sigma$ implies $\exists\mu'.\ \rho' \leq \mu' \leftrightarrow^* \mu$, thus the notion of prefix is well-defined, and a partial order.

The following concept of an FS-run extends the notion of a partial FS-run to possibly infinite runs, in such a way that an FS-run is completely determined by its finite approximations.

Definition 13. Let N be a net.

An *FS-run* R of N is a set of partial FS-runs of N such that

- $[\rho] \leq [\sigma] \in R \Rightarrow [\rho] \in R$ (R is prefix-closed), and

- $[\sigma], [\rho] \in R \Rightarrow \exists[\mu] \in R.\ [\sigma] \leq [\mu] \wedge [\rho] \leq [\mu]$ (R is directed).

The class of partial FS-runs and the finite elements (in the set theoretical sense) in the class of FS-runs are in bijective correspondence. Every finite FS-run R must have a largest element, say $[\sigma]$, and the set of all prefixes of $[\sigma]$ is R. Conversely, the set of prefixes of a partial FS-run $[\sigma]$ is a finite FS-run of which the largest element is again $[\sigma]$.

Similar to the construction of FS-runs as sets of equivalence classes of firing sequences, we define *BD-runs* as sets of swapping equivalence classes of finite GR-processes. There is a close relationship between BD-runs and BD-processes, some details of which we will give in Section 7.

Definition 14. Let N be a net.
- A *partial BD-run* of N is a $\approx_S^*$-equivalence class of finite processes.

- A *BD-run* of N is a prefix-closed and directed set of partial BD-runs of N.

There is a bijective correspondence between partial BD-runs and the finite elements in the class of BD-runs, just as in the case of FS-runs above.

Much more interesting however is the bijective correspondence between BD-runs and FS-runs we will now establish. In particular, it allows us to prove theorems about firing sequences and lift them to processes with relative ease.

Definition 15. Let $N = (S, T, F, M_0)$ be a net, and let $P = ((\mathcal{S}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0), \pi)$ be a finite process of N and $\sigma \in \mathrm{FS}(N)$.
- $\mathrm{Lin}(P) := \{\pi(t_1)\pi(t_2)\dots\pi(t_n) \mid t_i \in \mathcal{T} \wedge n = |\mathcal{T}| \wedge t_i \mathcal{F}^* t_j \Rightarrow i \leq j\}$ (the *linearisations* of P).

- $\Pi(\sigma) := \{P \mid \sigma \in \mathrm{Lin}(P)\}$.

For one-safe nets, $\Pi(\sigma)$ contains exactly one process up to isomorphism, for any firing sequence σ [BD87].

Theorem 1. Let $N = (S, T, F, M_0)$ be a net, $\sigma, \rho \in \mathrm{FS}(N)$, and P, Q two finite processes of N.

1. If $\Pi(\sigma) \cap \Pi(\rho) \neq \emptyset$ then $\sigma \leftrightarrow^* \rho$.
2. If $\mathrm{Lin}(P) \cap \mathrm{Lin}(Q) \neq \emptyset$ then $P \approx_S^* Q$.

Proof: See [BD87]. $\qquad\square$

6

Theorem 2. Let $N=(S,T,F,M_0)$ be a net, $\sigma,\rho \in \mathrm{FS}(N)$, and P,Q two finite processes of N.

1. If $\sigma \leftrightarrow \rho$ then $\Pi(\sigma) \cap \Pi(\rho) \neq \emptyset$.
2. If $P \approx_{\mathrm{S}} Q$ then $\mathrm{Lin}(P) \cap \mathrm{Lin}(Q) \neq \emptyset$.

Proof: See [BD87]. $\qquad\square$

Theorem 3. Let $N = (S,T,F,M_0)$ be a net, P,Q two finite processes of N, $\sigma \in \mathrm{Lin}(P)$, and $\rho \in \mathrm{Lin}(Q)$.
$\qquad \sigma \leftrightarrow^* \rho$ iff $P \approx_{\mathrm{S}}^* Q$.

Proof: "$\Rightarrow$": We show that $\forall n \in \mathbb{N}.\ (\sigma \leftrightarrow^n \rho \Rightarrow P \approx_{\mathrm{S}}^* Q)$ by induction on n. To start, $\sigma \leftrightarrow^0 \rho$ means $\sigma = \rho$, so $\sigma \in \mathrm{Lin}(P) \cap \mathrm{Lin}(Q)$. By Theorem 1 then $P \approx_{\mathrm{S}}^* Q$. For the induction step, we need to show that $\sigma \leftrightarrow^n \rho \Rightarrow P \approx_{\mathrm{S}}^* Q$. There must exists some μ such that $\sigma \leftrightarrow \mu \leftrightarrow^{(n-1)} \rho$. By Theorem 2 there is some $P' \in \Pi(\sigma) \cap \Pi(\mu)$. So $\sigma \in \mathrm{Lin}(P) \cap \mathrm{Lin}(P')$ and per Theorem 1, $P \approx_{\mathrm{S}}^* P'$. That $\mu \leftrightarrow^{(n-1)} \rho \Rightarrow P' \approx_{\mathrm{S}}^* Q$ follows from the induction assumption.

"$\Leftarrow$": Goes likewise but with the rôles of $\leftrightarrow$ and $\approx_{\mathrm{S}}$ and those of Π and Lin exchanged. $\qquad\square$

The functions Lin and Π can be lifted to equivalence classes of finite processes and firing sequences, respectively, by
$\qquad \mathrm{Lin}(\llbracket P \rrbracket) := [\sigma]$ for σ an arbitrary element of $\mathrm{Lin}(P)$, and
$\qquad \Pi([\sigma]) := \llbracket P \rrbracket$ for P an arbitrary element of $\Pi(\sigma)$.
Theorem 3 ensures that these liftings are well-defined, and that they are inverses of each other, thereby obtaining a bijective correspondence between partial BD-runs and partial FS-runs. The following theorem tells that this bijection respects the prefix ordering between runs.

Theorem 4. Let $N = (S,T,F,M_0)$ be a net, P,Q two finite processes of N, $\sigma \in \mathrm{Lin}(P)$, and $\rho \in \mathrm{Lin}(Q)$.
$\qquad [\sigma] \leq [\rho]$ iff $\llbracket P \rrbracket \leq \llbracket Q \rrbracket$.

Proof: "$\Leftarrow$": Take $P' \in \llbracket P \rrbracket$ and $Q' \in \llbracket Q \rrbracket$ such that $P' \leq Q'$. If follows immediately from Definitions 6 and 15 that any $\sigma' \in \mathrm{Lin}(P')$ can be extended to some $\rho' \in \mathrm{Lin}(Q')$, so that $[\sigma] = [\sigma'] \leq [\rho'] = [\rho]$.

"$\Rightarrow$": Take $\rho' \in [\rho]$ such that $\sigma \leq \rho'$ and take $Q' = ((\mathscr{P}', \mathscr{T}', \mathscr{F}', \mathscr{M}'_0), \pi') \in \Pi(\rho')$. By Definition 15, $\mathscr{T}'$ can be enumerated as $t_1 t_2 \ldots t_n$ such that $t_i \mathscr{F}'^* t_j \Rightarrow i \leq j$ and $\rho' = \pi'(t_1)\pi'(t_2)\ldots\pi'(t_n)$. So $\sigma = \pi'(t_1)\pi'(t_2)\ldots\pi'(t_m)$ with $m \leq n$. It follows from the remark below Definition 6 that Q' has a prefix P' with transitions $\{t_1, \ldots, t_m\}$ such that $P' \in \Pi(\sigma)$. Hence $\llbracket P \rrbracket = \llbracket P' \rrbracket \leq \llbracket Q' \rrbracket = \llbracket Q \rrbracket$. $\qquad\square$

Since BD-runs are created out of the ordered space of partial BD-runs of a net in the same way as FS-runs are created out of partial FS-runs, this immediately yields a bijective correspondence also between infinite (in the set-theoretical sense) BD-runs and infinite FS-runs. This bijection respects the subset relation $\subseteq$ between runs, which is the counterpart of the prefix relation $\leq$ between partial runs, and hence also the concept of a maximal run.

6. Abstract Runs of Structural Conflict Nets

This section formally uses FS-runs; however the results carry over to BD-runs easily, via the bijection established in Section 5.

Returning to the example of Figure 2, we find that the depicted net has only one maximal FS-run: $[abdc]=[adbc]=[adcb]=[acdb]=[cadb]=[cdab]=[cdba]=[cbda]=[bcda]=[bdca]=[bdac]=[badc]$. The conflict between the initially enabled sets of transitions $\{a,b\}$, $\{b,c\}$, and $\{a,c\}$ has not been resolved; rather all possibilities have been included in the same run. The following definition describes runs for which this is not the case.

Definition 16. Let $N = (S,T,F,M_0)$ be a net.
An FS-run R is *conflict-free* iff for all finite, non-empty multisets $G \in \mathbb{N}^T$ and all $\sigma \in T^*$
$$(\forall t \in G.\ [\sigma t^{G(t)}] \in R \wedge M_0 \xrightarrow{\sigma} \xrightarrow{G\restriction\{t\}}) \Rightarrow M_0 \xrightarrow{\sigma} \xrightarrow{G}\ .$$

We will now show that in structural conflict nets every run is conflict free. For structural conflict nets thus holds what one would intuitively expect: every conflict in the net gives rise to distinct runs, each one representing a particular way to resolve the conflict.

Theorem 5. Let N be a structural conflict net.
Every FS-run R of N is conflict-free.

Proof: Let R be an FS-run of $N = (S,T,F,M_0)$, $\sigma \in T^*$, and $G \in \mathbb{N}^T$ a finite, non-empty multiset such that $\forall t \in G.\ [\sigma t^{G(t)}] \in R \wedge M_0 \xrightarrow{\sigma} \xrightarrow{G\restriction\{t\}}$. Let M be the unique marking of N with $M_0 \xrightarrow{\sigma} M$. We have to show that $M \xrightarrow{G}$.

No transition t can occur more than once in G as self-concurrency cannot occur in structural conflict nets and $M_0 \xrightarrow{\sigma} \xrightarrow{G\restriction\{t\}}$.

Let $t,u \in G$, $t \neq u$. Then $M \xrightarrow{t} \wedge M \xrightarrow{u}$. Since R is directed, there exist $\rho, \mu \in T^*$ with $\sigma t \rho \leftrightarrow^* \sigma u \mu$. By Definition 11, $\sigma t \rho$ and $\sigma u \mu$ must contain the same multiset of transitions. Hence somewhere in the sequence $\sigma t \rho = \nu_1 \leftrightarrow \nu_2 \leftrightarrow \cdots \leftrightarrow \nu_n = \sigma u \mu$ the transitions t and u must be exchanged, i.e. $\nu_i = \nu' t u \nu'' \leftrightarrow \nu' u t \nu'' = \nu_{i+1}$. Thus there is a marking M' with $M_0 \xrightarrow{\nu'} M' \xrightarrow{\{t,u\}}$. Since N is a structural conflict net, ${}^\bullet t \cap {}^\bullet u = \emptyset$. As this holds for all $t,u \in G$, it follows that $M \xrightarrow{G}$. $\qquad\square$

Theorem 6. Let N be a structural conflict net.
If N has exactly one maximal FS-run then N is conflict-free.

Proof: Let $N = (S,T,F,M_0)$. Assume N has a conflict, i.e. there exists $\sigma \in T^*$, $M \in \mathbb{N}^S$, $G \in \mathbb{N}^T$, G finite, with $M_0 \xrightarrow{\sigma} M$, $\neg M \xrightarrow{G}$ and $\forall t \in G.\ M \xrightarrow{G\restriction\{t\}}$. We show that N has no unique maximal FS-run.

For every $t \in G$, the set $\{[\rho] \mid [\rho] \leq [\sigma t^{G(t)}]\}$ constitutes an FS-run of N. Hence, a unique maximal FS-run of N would be a superset of $\{[\sigma t^{G(t)}] \mid t \in G\}$, and thus not conflict-free. However, every FS-run of N must be conflict-free according to Theorem 5. $\qquad\square$

7. BD-Processes of Structural Conflict Nets

In this section we adapt Theorem 6 from runs to BD-processes, i.e. GR-processes up to $\approx_{\mathrm{S}}^{\infty}$. To this end, we give a mapping from GR-processes to BD-runs.

Definition 17. Let N be a net and P a process thereof. Then $BD(P) := \downarrow \{[\![P']\!] \mid P' \leq P, \ P' \text{ finite}\}$.

Note that, by Definition 8, $P \approx_{\mathrm{S}}^{\infty} Q$ iff $BD(P) = BD(Q)$.

Lemma 1. Let N be a net and P a process thereof. $BD(P)$ is a BD-run.

Proof: Prefix-closure holds by definition of $\downarrow$, using the transitivity of $\leq$.

For any ordered set X, if X is directed, so is $\downarrow X$. Hence it suffices to show that $\{[\![P']\!] \mid P' \leq P, \ P' \text{ finite}\}$ is directed. Take finite $P_i := ((\mathscr{P}_i, \mathscr{T}_i, \mathscr{F}_i, \mathscr{M}_{0i}), \pi_i) \leq P$ for $i = 1, 2$. Then $P' := ((\mathscr{P}_1 \cup \mathscr{P}_2, \mathscr{T}_1 \cup \mathscr{T}_2, \mathscr{F}_1 \cup \mathscr{F}_2, \mathscr{M}_{01} \cup \mathscr{M}_{02}), \pi_1 \cup \pi_2) \leq P$ and P' is finite. Moreover, $P_i \leq P'$ for $i = 1, 2$ and thus $[\![P_i]\!] \leq [\![P']\!]$. Hence $BD(P)$ is directed. $\qquad\square$

We now show that the existence of a unique maximal GR-process implies the existence of a unique maximal BD-run.

Lemma 2. Let N be a net.
Every process P of N is a prefix of a maximal process of N.

Proof: The set of all processes of N of which P is a prefix is partially ordered by $\leq$. Every chain in this set has an upper bound, obtained by componentwise union. Via Zorn's Lemma this set contains at least one maximal process. $\qquad\square$

Lemma 3. Let N be a net.
If N has exactly one maximal GR-process up to $\approx_{\mathrm{S}}^{\infty}$ then N has exactly one maximal BD-run.

Proof: Take any finite processes P, P' of N. According to Lemma 2 there are maximal processes Q, Q' of N with $P \leq Q$, $P' \leq Q'$. As N has exactly one maximal process up to $\approx_{\mathrm{S}}^{\infty}$, $Q \approx_{\mathrm{S}}^{\infty} Q'$ and $BD(Q) = BD(Q')$. Hence as $[\![P']\!] \in BD(Q')$ also $[\![P']\!] \in BD(Q)$. Since $BD(Q)$ is directed, there exists a Q'' with $[\![P]\!] \leq [\![Q'']\!] \wedge [\![P']\!] \leq [\![Q'']\!]$. As this holds for any finite processes P, P' the set of all equivalence classes of finite processes of N is directed and hence a BD-run. Naturally this is the largest BD-run. $\quad\square$

We can now conclude our main result: A semantic conflict in structural conflict nets generates multiple maximal GR-processes even up to swapping equivalence.

Corollary 1. Let N be a structural conflict net.
If N has only one maximal GR-process up to $\approx_{\mathrm{S}}^{\infty}$ then N is conflict-free.

Proof: This follows directly from Lemma 3 and Theorem 6, using the bijection between FS-runs and BD-runs of Section 5. $\qquad\square$

It would be interesting to show the reverse direction of Corollary 1, i.e. to prove that a structural conflict net has exactly one maximal GR-process up to $\approx_{\mathrm{S}}^{\infty}$ *iff* it is conflict-free. We do conjecture that this holds for countable nets. Even for processes generated by finite nets though, we find it difficult to apply a similar proof technique by establishing the necessary bijective correspondence between infinite BD-processes and infinite BD-runs.

References

[BD87] E. Best & R.R. Devillers (1987): *Sequential and concurrent behaviour in Petri net theory.* Theoretical Computer Science 55(1), pp. 87–136. See also: E. Best & R.R. Devillers (1987): *Interleaving and Partial Orders in Concurrency: A Formal Comparison.* In M. Wirsing, editor: *Formal Description of Programming Concepts III*, pp. 299–321, North-Holland.

[BMO09] K. Barylska, Ł. Mikulski & E. Ochmański (2009): *Nonviolence Petri nets.* In Proc. Worksh. on Concurrency, Specification and Programming, CS&P'09, pp. 50–59.

[DMM96] P. Degano, J. Meseguer & U. Montanari (1996): *Axiomatizing the algebra of net computations and processes.* Acta Informatica 33, pp. 641–667.

[Eng91] J. Engelfriet (1991): *Branching processes of Petri nets.* Acta Informatica 28(6), pp. 575–591.

[vG05] R.J. van Glabbeek (2005): *The individual and collective token interpretations of Petri nets.* In M. Abadi & L. de Alfaro, eds.: Proc. 16th Int. Conf. on Concurrency Theory, CONCUR'05, LNCS 3653, Springer, pp. 323–337.

[Gol86] U. Goltz (1986): *How many transitions may be in conflict? Petri Net Newsletter 25, pp. 4–9.

[GP95] R.J. van Glabbeek & G.D. Plotkin (1995): *Configuration structures (extended abstract).* In D. Kozen, editor: Proc. Tenth Annual Symp. on Logic in Computer Science, LICS'95, San Diego, USA, IEEE, pp. 199–209.

[GR83] U. Goltz & W. Reisig (1983): *The non-sequential behaviour of Petri nets.* Information and Control 57(2-3), pp. 125–147.

[GSW80] H.J. Genrich & E. Stankiewicz-Wiechno (1980): *A dictionary of some basic notions of net theory.* In W. Brauer, editor: Advanced Course: Net Theory and Applications, LNCS 84, Springer, pp. 519–531.

[HKT95] P.W. Hoogers, H.C.M. Kleijn & P.S. Thiagarajan (1995): *A trace semantics for Petri nets.* Information and Computation 117, pp. 98–114.

[Maz89] A.W. Mazurkiewicz (1989): *Concurrency, modularity, and synchronization.* In Proc. Math. Foundations of Comp. Sci., MFCS'89, LNCS 379, Springer, pp. 577–598.

[Maz95] A.W. Mazurkiewicz (1995): *Introduction to trace theory.* In V. Diekert & G. Rozenberg, editors: The Book of Traces, World Scientific, pp. 3–41.

[MM88] J. Meseguer & U. Montanari (1988): *Petri nets are monoids: A new algebraic foundation for net theory.* In Proc. Third Annual Symp. on Logic in Computer Science, LICS'88, Edinburgh, Scotland, IEEE, pp. 155–164.

[MMS97] J. Meseguer, U. Montanari & V. Sassone (1997): *On the semantics of place/transition Petri nets.* Mathematical Structures in Computer Science 7(4), pp. 359–397.

[NPW81] M. Nielsen, G.D. Plotkin & G. Winskel (1981): *Petri nets, event structures and domains, part I.* Theoretical Computer Science 13(1), pp. 85–108.

[Och89] Edward Ochmański, 1989: *Personal communication.*

[Pet77] C.A. Petri (1977): *Non-sequential processes.* GMD-ISF Report 77.05, GMD.

[Vog91] W. Vogler (1991): *Executions: a new partial-order semantics of Petri nets.* Theoretical Computer Science 91(2), pp. 205–238.