On Causal Semantics of Petri Nets
(extended abstract)*

Rob van Glabbeek[1,2], Ursula Goltz[3] & Jens-Wolfhard Schicke[3]

[1] NICTA, Sydney, Australia
[2] School of Comp. Sc. and Engineering, Univ. of New South Wales, Sydney, Australia
[3] Institute for Programming and Reactive Systems, TU Braunschweig, Germany

rvg@cs.stanford.edu goltz@ips.cs.tu-bs.de drahflow@gmx.de

Abstract. We consider approaches for causal semantics of Petri nets, explicitly representing dependencies between transition occurrences. For one-safe nets or condition/event-systems, the notion of process as defined by Carl Adam Petri provides a notion of a run of a system where causal dependencies are reflected in terms of a partial order. A well-known problem is how to generalise this notion for nets where places may carry several tokens. Goltz and Reisig have defined such a generalisation by distinguishing tokens according to their causal history. However, this so-called *individual token interpretation* is often considered too detailed. A number of approaches have tackled the problem of defining a more abstract notion of process, thereby obtaining a so-called *collective token interpretation*. Here we give a short overview on these attempts and then identify a subclass of Petri nets, called *structural conflict nets*, where the interplay between conflict and concurrency due to token multiplicity does not occur. For this subclass, we define abstract processes as equivalence classes of Goltz-Reisig processes. We justify this approach by showing that we obtain exactly one maximal abstract process if and only if the underlying net is conflict-free with respect to a canonical notion of conflict.

1 Introduction

In this paper we address a well-known problem in Petri net theory, namely how to generalise Petri's concept of non-sequential processes to nets where places may carry multiple tokens.

One of the most interesting features of Petri nets is that they allow the explicit representation of causal dependencies between action occurrences when modelling reactive systems. This is a key difference with models of reactive systems (like standard transition systems) with an inherent so-called interleaving semantics, modelling concurrency by non-deterministic choice between sequential executions. In [GG01] it has been shown, using the model of event structures or configuration structures, that causal semantics are superior to interleaving semantics when giving up the assumption that actions are atomic entities.

* This work was partially supported by the DFG (German Research Foundation).

In the following, we give a concise overview on existing approaches on semantics of Petri nets that give an account of their runs, without claiming completeness, and following closely a similar presentation in [GGS11a].

Initially, Petri introduced the concept of a net together with a definition of its dynamic behaviour in terms of the firing rule for single transitions or for finite sets (*steps*) of transitions firing in parallel. Sequences of transition firings or of steps are the usual way to define the behaviour of a Petri net. When considering only single transition firings, the set of all firing sequences yields a linear time interleaving semantics (no choices between alternative behaviours are represented). Otherwise we obtain a linear time step semantics, with information on possible parallelism, but without explicit representation of causal dependencies between transition occurrences.

Petri then defined *condition/event systems*, where — amongst other restrictions — places (there 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 transitions and places 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.

However, the most frequently used class of Petri nets are nets where places may carry arbitrary many tokens, or a certain maximal number of tokens when adding place capacities. This type of nets is often called *place/transition systems* (P/T systems). Here tokens are usually assumed to be indistinguishable entities, for example representing a number of available resources in a system. Unfortunately, it is not straightforward to generalise the notion of process, as defined by Petri for condition/event systems, to P/T systems. In fact, it has now for more than 20 years been a well-known problem in Petri net theory how to formalise an appropriate causality-based concept of process or run for general P/T systems. In the following we give an introduction to the problem and a short overview on existing approaches.

As a first approach, Goltz and Reisig generalised Petri's notion of process to general P/T systems [GR83]. We call this notion of a process *GR-process*. It is based on a canonical unfolding of a P/T systems into a condition/event system, representing places that may carry several tokens by a corresponding number of conditions (see [Gol87]). Fig. 1 shows a P/T system with two of its GR-processes.

Engelfriet adapted GR-processes by additionally representing choices between alternative behaviours [Eng91], thereby adopting the approach of [NPW81]

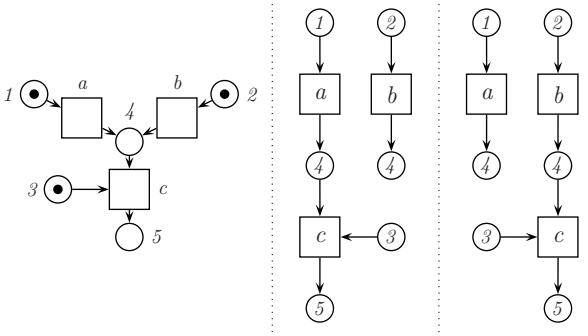

Fig. 1. A net N with its two maximal GR-processes. The correspondence between elements of the net and their occurrences in the processes is indicated by labels.

to P/T systems, although without arc weights. Meseguer, Sassone and Montanari extended this to cover also arc weights [MMS97].

However, if one wishes to interpret P/T systems with a causal semantics, there are alternative interpretations of what "causal semantics" should actually mean. Goltz already 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 complete run (for there are no choices to resolve), yet it may have multiple maximal GR-processes [Gol86]. This phenomenon already occurs in Fig. 1, since the choice between alternative behaviours is here only due to the possibility to choose between two tokens which can or even should be seen as indistinguishable entities. 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, as in the interpretation underlying GR-processes and the approach of [Eng91,MMS97], 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. For an alternative interpretation, they use the term *collective token interpretation* [GP95]. A possible formalisation of these interpretations occurs in [Gla05]. In the following we call process notions for P/T systems which are adherent to a collective token philosophy *abstract processes*. Another 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].

Mazurkiewicz applies again a different approach in [Maz89]. He proposes *multitrees*, which record possible multisets of fired transitions, and then takes confluent subsets of multitrees as abstract processes of P/T systems. This approach does not explicitly represent dependencies between transition occurrences

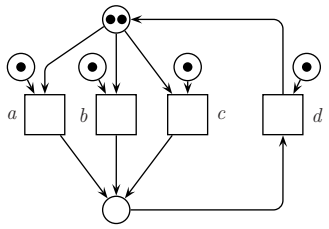

Fig. 2. A net with only a single process up to swapping equivalence.

and hence does not apply to nets with self-loops, where such information may not always be retrieved.

Yet another approach has been proposed by Best and Devillers in [BD87]. Here an 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. In this paper, we adopt this approach and we show that it yields a fully satisfying solution for a subclass of P/T systems. 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 [DMM89] 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]".

To explain why it can be argued that BD-processes are not fully satisfying as abstract processes for general P/T systems, we recall in Fig. 2 an example due to Ochmański [Och89,BMO09], see also [DMM89,GGS11a]. In the initial situation only two of the three enabled transitions can fire, which constitutes a conflict. However, the equivalence obtained from the swapping transformation (formally defined in Section 3) identifies all possible maximal GR-processes and hence yields only one complete abstract run of the system. We are not aware of a solution, i.e. any formalisation of the concept of a run of a net that correctly represents both causality and parallelism of nets, and meets the requirement that for this net there is more than one possible complete run.

In [GGS11a] and in the present paper, we continue the line of research of [MM88,DMM89,Maz89,HKT95] to formalise a causality-based notion of an abstract process of a P/T system that fits a 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'. The example from Fig. 2 shows that BD-processes are in general not suited. We defined in [GGS11a] a subclass of P/T systems where conflict and concurrency are clearly separated. We called these nets *structural conflict nets*. Using the formalisation of conflict for P/T systems from [Gol86], we have shown that, for this subclass of P/T systems, we obtain more than one maximal BD-process whenever the net contains a conflict.[1] The proof of this result is quite involved; it was achieved by using an alternative characterisation of BD-processes via firing sequences from [BD87].

In this paper, we will show the reverse direction of this result, namely that we obtain exactly one maximal BD-process of a structural conflict net if the net is conflict-free. Depending on the precise formalisation of a suitable notion of maximality of BD-processes, this holds even for arbitrary nets. Summarising, we then have established that we obtain exactly one maximal abstract process in terms of BD-processes for structural conflict nets *if and only if* the net is conflict-free with respect to a canonical notion of conflict.

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. Section 4 recalls the concept of conflict in P/T systems and defines structural conflict nets.[2] In Section 5, we recapitulate the alternative characterisation of BD-processes from [BD87] in terms of an equivalence notion on firing sequences [BD87] and prove in this setting that a conflict-free net has exactly one maximal run. Finally, in Section 6, we investigate notions of maximality for BD-processes and then transfer the result from Section 5 to BD-processes. Due to lack of space, the proofs of Lemma's 2, 3, 4, 6 and 7, some quite involved, are omitted; these can be found in [GGS11b].

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 $A(x) \leq B(x)$ for all $x \in X$;
  $A \cup B$ denotes the multiset over X with $(A \cup B)(x) := \max(A(x), B(x))$,
  $A \cap B$ denotes the multiset over X with $(A \cap B)(x) := \min(A(x), B(x))$,
  $A + B$ denotes the multiset over X with $(A + B)(x) := A(x) + B(x)$,
  $A - B$ is given by $(A - B)(x) := A(x) \mathbin{\dot{-}} B(x) = \max(A(x) - B(x), 0)$, and
  for $k \in \mathbb{N}$ the multiset $k \cdot A$ is given by $(k \cdot A)(x) := k \cdot A(x)$.

[1] The notion of maximality for BD-processes is not trivial. However, with the results from Section 6, Corollary 1 from [GGS11a] may be rephrased in this way.

[2] The material in Sections 2 to 4 follows closely the presentation in [GGS11a], but needs to be included to make the paper self-contained.

- 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 $A \restriction Y$ denotes the multiset over Y defined by $(A \restriction Y)(x) := A(x)$ 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 $A \restriction (X \cap Y) = B \restriction (X \cap Y)$, $A \restriction (X \setminus Y) = \emptyset$, and $B \restriction (Y \setminus X) = \emptyset$. 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 $A(x) \leq 1$ 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 ${}^\bullet x,\ x^\bullet : S \cup T \to \mathbb{N}$ are given by ${}^\bullet x(y) = F(y,x)$ and $x^\bullet(y) = F(x,y)$ for all $y \in S \cup T$. If $x \in T$, the elements of ${}^\bullet x$ and $x^\bullet$ are called *pre-* and *postplaces* of x, respectively. These functions extend to multisets $X : S \cup T \to \mathbb{N}$ as usual, by ${}^\bullet X := \Sigma_{x \in S \cup T} X(x) \cdot {}^\bullet x$ and $X^\bullet := \Sigma_{x \in S \cup T} X(x) \cdot x^\bullet$.

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 $M \xrightarrow{G}_N M'$, iff

- ${}^\bullet G \subseteq M$ (G is *enabled*) and
- $M' = (M - {}^\bullet G) + G^\bullet$.

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 $M \xrightarrow{\sigma} M'$ for

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

When omitting σ or M' we always mean it to be existentially quantified. When $M_0 \xrightarrow{\sigma}_N$, 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 $M \xrightarrow{\{t,u\}}$ implies $M \xrightarrow{tu}$ and $M \xrightarrow{ut}$. We use the notation $t \in \sigma$ to indicate that the transition t occurs in the sequence σ, and $\sigma \leq \rho$ 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
  - $\forall s \in \mathcal{S}\,.\ |{}^\bullet s| \leq 1 \geq |s^\bullet| \wedge \mathcal{M}_0(s) = \begin{cases} 1 & \text{if } {}^\bullet s = \emptyset \\ 0 & \text{otherwise,} \end{cases}$
  - $\mathcal{F}$ is acyclic, i.e. $\forall x \in \mathcal{S} \cup \mathcal{T}.\ (x,x) \notin \mathcal{F}^+$, 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 $u \in \mathcal{T}$.

- $\pi : \mathcal{S} \cup \mathcal{T} \to S \cup T$ is a function with $\pi(\mathcal{S}) \subseteq S$ 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 $\mathcal{T}$ is 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 $P = ((\mathcal{S}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0), \pi)$ and
$P' = ((\mathcal{S}', \mathcal{T}', \mathcal{F}', \mathcal{M}_0'), \pi')$ be two processes of the same net.

- P' is a *prefix* of P, notation $P' \leq P$, and P an *extension* of P', iff $\mathcal{S}' \subseteq \mathcal{S}$, $\mathcal{T}' \subseteq \mathcal{T}$, $\mathcal{M}_0' = \mathcal{M}_0$, $\mathcal{F}' = \mathcal{F} \restriction (\mathcal{S}' \times \mathcal{T}' \cup \mathcal{T}' \times \mathcal{S}')$ and $\pi' = \pi \restriction (\mathcal{S}' \times \mathcal{T}')$.
- 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 $y \in \mathcal{S}' \cup \mathcal{T}'$ then $x \in \mathcal{S}' \cup \mathcal{T}'$. 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 $\mathcal{N} = (\mathcal{S}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0)$ and $\mathcal{N}' = (\mathcal{S}', \mathcal{T}', \mathcal{F}', \mathcal{M}_0')$ is a bijection between their places and transitions such that $\mathcal{M}_0'(\phi(s)) = \mathcal{M}_0(s)$ for all $s \in \mathcal{S}$ and $\mathcal{F}'(\phi(x), \phi(y)) = \mathcal{F}(x, y)$ for all $x, y \in \mathcal{S} \cup \mathcal{T}$.

Next we formally introduce the swapping transformation and the resulting equivalence notion on GR-processes from [BD87].

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

Then swap(P, p, q) is defined as $((\mathcal{S}, \mathcal{T}, \mathcal{F}', \mathcal{M}_0), \pi)$ 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 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 $\llbracket P \rrbracket$ 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 $\llbracket P' \rrbracket \leq \llbracket P \rrbracket$ iff $P' \approx_{\mathrm{S}}^* Q' \leq Q \approx_{\mathrm{S}}^* P$ for some Q', Q.
- Two processes P and Q are *swapping equivalent* ($P \approx_{\mathrm{S}}^\infty Q$) iff

$$\downarrow(\{\llbracket P' \rrbracket \mid P' \leq P,\ P' \text{ finite}\}) = \downarrow(\{\llbracket Q' \rrbracket \mid Q' \leq Q,\ Q' \text{ finite}\})$$

  where $\downarrow$ denotes prefix-closure under $\leq$.
- We call a $\approx_{\mathrm{S}}^\infty$-equivalence class of processes a *BD-process*, and write $\llbracket P \rrbracket_\infty$.

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 follows from Theorem 4 in [GGS11a].

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

Note that if $P \approx_{\mathrm{S}}^\infty Q$ and P is finite, then also Q is finite. Moreover, for finite GR-processes P and Q we have $P \approx_{\mathrm{S}}^\infty Q$ iff $P \approx_{\mathrm{S}}^* Q$. Thus, for a finite GR-process P, we have $\llbracket P \rrbracket_\infty = \llbracket P \rrbracket$. In that case we call $\llbracket P \rrbracket$ a *finite* BD-process.

We define a *BD-run* as a more abstract and more general form of BD-process. Like a BD-process, a BD-run is completely determined by its finite approximations, which are finite BD-processes; however, a BD-run does not require that these finite approximations are generated by a given GR-process.

Definition 9. Let N be a net.

A *BD-run* $\mathcal{R}$ of N is a non-empty set of finite BD-processes of N such that

- $\llbracket P \rrbracket \leq \llbracket Q \rrbracket \in \mathcal{R} \Rightarrow \llbracket P \rrbracket \in \mathcal{R}$ ($\mathcal{R}$ is prefix-closed), and
- $\llbracket P \rrbracket, \llbracket Q \rrbracket \in \mathcal{R} \Rightarrow \exists \llbracket U \rrbracket \in \mathcal{R}.\ \llbracket P \rrbracket \leq \llbracket U \rrbracket \wedge \llbracket Q \rrbracket \leq \llbracket U \rrbracket$ ($\mathcal{R}$ is directed).

The class of finite BD-processes and the finite elements (in the set theoretical sense) in the class of BD-runs are in bijective correspondence. Every finite BD-run $\mathcal{R}$ must have a largest element, say $\llbracket P \rrbracket$, and the set of all prefixes of $\llbracket P \rrbracket$ is $\mathcal{R}$. Conversely, the set of prefixes of a finite BD-process $\llbracket P \rrbracket$ is a finite BD-run of which the largest element is again $\llbracket P \rrbracket$.

We now define a canonical mapping from GR-processes to BD-runs.

Definition 10. Let N be a net and P a process thereof.

Then $BD(P) := \downarrow\{\llbracket P' \rrbracket \mid P' \leq P,\ P' \text{ finite}\}$.

Lemma 1. Let N be a net and P a process thereof.

Then $BD(P)$ is a BD-run.

Proof. See [GGS11a, Lemma 1]. □

This immediately yields an injective function from BD-processes to BD-runs, since by Definition 8, $P \approx_{\mathrm{S}}^{\infty} Q$ iff $BD(P) = BD(Q)$. For countable nets (i.e. nets with countably many places and transitions), this function is even a bijection.

Lemma 2. Let N be a countable net and $\mathcal{R}$ a BD-run of N.

Then $\mathcal{R}$ is countable and there exists a process P of N such that $\mathcal{R} = BD(P)$.

Lemma 2 does not hold for uncountable nets, as witnessed by the counterexample in Fig. 3. This net N has a transition t for each real number $t \in \mathbb{R}$. Each such transition has a private preplace s_t with $M_0(s_t) = 1$ and $F(s_t, t) = 1$, which ensures that t can fire only once. Furthermore there is one shared place s with $M_0(s) = 2$ and a loop $F(s, t) = F(t, s) = 1$ for each transition t. There are no other places, transitions or arcs besides the ones mentioned above.

Each GR-process of N, and hence also each BD-process, has only countably many transitions. Yet, any two GR-processes firing the same finite set of transitions of N are swapping equivalent, and the set of all finite BD-processes of N constitutes a single BD-run involving all transitions.

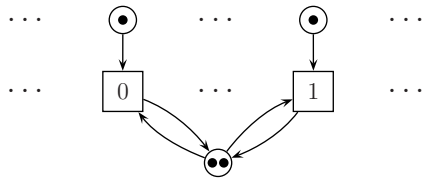

Fig. 3. A net with no maximal GR-process, but with a maximal BD-run.

We now show that the mapping BD respects the ordering of processes.

Lemma 3. Let N be a net, and P and P' two GR-processes of N.

If $P \leq P'$ then $BD(P) \subseteq BD(P')$.

4 Conflicts in place/transition systems

We recall the canonical notion of conflict introduced in [Gol86].

Definition 11. 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 $M_0 \longrightarrow M$.
- N is *binary-conflict–free* iff no multiset $G \in \mathbb{N}^T$ with $|G| = 2$ 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 proposed in [GGS11a] a class of P/T systems where the structural definition of conflict in terms of shared preplaces, as often used in Petri net theory, matches the semantic definition of conflict as given above. We called this class of nets *structural conflict nets*. For a net to be a structural conflict net, we require that two transitions sharing a preplace will never occur both in one step.

Definition 12. 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 M iff G is a set and two distinct transitions in G are in conflict in M. Hence a structural conflict net is conflict-free if and only if it is binary-conflict–free. Moreover, two transitions enabled in M are in (semantic) conflict iff they share a preplace.

5 A conflict-free net has exactly one maximal run

In this section, we recapitulate results from [BD87], giving an alternative characterisation of runs of a net in terms of firing sequences. We use an adapted notation and terminology and a different treatment of infinite runs, as in [GGS11a]. As a main result of the present paper, we then prove in this setting that a conflict-free net has exactly one maximal run. In the following section, this result will be transferred to BD-processes.

The behaviour of a net can be described not only by its processes, but also by its firing sequences. The imposed total order on transition firings abstracts from information on causal dependence, or concurrency, between transition firings. To retrieve this information we introduce an *adjacency* relation on firing sequences, recording which interchanges of transition occurrences are due to semantic independence of transitions. Hence 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.

Definition 13. 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 14. 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 $\sigma' \leftrightarrow^* \sigma \leq \mu$ implies $\exists \mu'.\ \sigma' \leq \mu' \leftrightarrow^* \mu$; thus the notion of prefix is well-defined, and a partial order.

Similar to the construction of BD-runs out of finite BD-processes, 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 15. Let N be a net.

An *FS-run* of N is a non-empty, prefix-closed and directed set of partial FS-runs of N.

There is a bijective correspondence between partial FS-runs and the finite elements in the class of FS-runs, just as in the case of BD-runs in Section 3. Much more interesting however is the following bijective correspondence between BD-runs and FS-runs.

Theorem 1. There exists a bijective function Π from FS-runs to BD-runs such that $\Pi(\mathcal{R}) \subseteq \Pi(\mathcal{R}')$ iff $\mathcal{R} \subseteq \mathcal{R}'$.

Proof. See [GGS11a], in particular the remarks at the end of Section 5. □

We now show that a conflict-free net has exactly one maximal run. As we have a bijective correspondence, it does not matter which notion of run we use here (FS-run or BD-run). We prove an even stronger result, using binary-conflict–free instead of conflict-free. In preparation we need the following lemma.

Lemma 4. Let N be a binary-conflict–free net.

If $\sigma, \sigma' \in \mathrm{FS}(N)$ then $\exists \mu, \mu'.\ \sigma\mu \in \mathrm{FS}(N) \wedge \sigma'\mu' \in \mathrm{FS}(N) \wedge \sigma\mu \leftrightarrow^* \sigma'\mu'$.

Theorem 2. Let N be a binary-conflict–free net.

There is exactly one maximal FS-run of N.

Proof. Let $\mathcal{R} = \{[\sigma] \mid \sigma \text{ is a finite firing sequence of } N\}$. We claim that $\mathcal{R}$ is said maximal FS-run of N.

First we show that $\mathcal{R}$ is prefix closed and directed, and thus indeed an FS-run.

Take any $[\rho] \leq [\sigma] \in \mathcal{R}$. Then by definition of $\leq$, $\exists \nu.\ \rho \leq \nu \wedge \nu \leftrightarrow^* \sigma$. We need to show that $[\rho] \in \mathcal{R}$, i.e. that ρ is a firing sequence of N. Since σ is a firing sequence of N and $\nu \leftrightarrow^* \sigma$, ν is also a firing sequence of N. Together with $\rho \leq \nu$ follows that ρ, too, is a firing sequence of N. Thus $\mathcal{R}$ is prefix closed.

To show directedness, let $[\sigma], [\rho] \in \mathcal{R}$. We need to show that $\exists [\nu] \in \mathcal{R}.\ [\sigma] \leq [\nu] \wedge [\rho] \leq [\nu]$, or with the definitions of $\leq$ and $[\ ]$ expanded, $\exists \nu.\ (\exists \alpha.\ \sigma \leq \alpha \hookrightarrow^* \nu \wedge \exists \beta.\ \rho \leq \beta \hookrightarrow^* \nu)$. We now apply Lemma 4 to $\sigma, \rho \in \mathrm{FS}(N)$, obtaining μ and μ' as mentioned in that lemma, and take $\alpha = \sigma\mu$ and $\beta = \rho\mu'$. Then Lemma 4 gives us $\alpha \hookrightarrow^* \beta$ and we take $\nu = \alpha$. Thus $\mathcal{R}$ is directed.

Finally we show that $\mathcal{R}$ is maximal. Take any run $\mathcal{R}'$ of N. Then $\mathcal{R}' \subseteq \mathcal{R}$ by definition of $\mathcal{R}$, hence $\mathcal{R}$ is maximal. □

6 BD-processes fit structural conflict nets

In this section we show that BD-processes are adequate as abstract processes for the subclass of structural conflict nets.

In [GGS11a] we have shown that a semantic conflict in a structural conflict net always gives rise to multiple maximal GR-processes even up to swapping equivalence.

Theorem 3. 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. Corollary 1 from [GGS11a]. □

We conjectured in [GGS11a] that, for countable nets, also the reverse direction holds, namely that a countable conflict-free structural conflict net has exactly one maximal GR-process up to $\approx_{\mathrm{S}}^{\infty}$.

In Section 5 we have already shown that a corresponding result holds for runs instead of processes. We will now transfer this result to BD-processes, and hence prove the conjecture.

We proceed by investigating three notions of maximality for BD-processes; they will turn out to coincide for structural conflict nets.

Definition 16.

- A BD-process $\mathbf{[}P\mathbf{]}_\infty$ is *weakly maximal* (or a maximal GR-process up to $\approx_{\mathrm{S}}^{\infty}$), iff some $P' \in \mathbf{[}P\mathbf{]}_\infty$ is maximal (in the GR-process sense).
- A BD-process $\mathbf{[}P\mathbf{]}_\infty$ is *maximal* iff $\forall P' \in \mathbf{[}P\mathbf{]}_\infty\ \forall Q.\ (P' \leq Q \Rightarrow P' \approx_{\mathrm{S}}^{\infty} Q)$.
- A BD-process $\mathbf{[}P\mathbf{]}_\infty$ is *run-maximal* iff the BD-run $BD(P)$ is maximal.

The first notion is the simplest way of inheriting the notion of maximality of GR-process by BD-processes, whereas the last one inherits the notion of maximality from BD-runs. The middle notion is the canonical notion of maximality with respect to a natural order on BD-process, defined below.

Definition 17. Let N be a net.

We define a relation $\preceq$ between BD-processes, via

$$\mathbf{[}P\mathbf{]}_\infty \preceq \mathbf{[}Q\mathbf{]}_\infty :\Leftrightarrow \exists P' \approx_{\mathrm{S}}^{\infty} P\ \exists Q' \approx_{\mathrm{S}}^{\infty} Q.\ P' \leq Q'\ ,$$

and construct an order between BD-processes via

$$\mathbf{[}P\mathbf{]}_\infty \leq \mathbf{[}Q\mathbf{]}_\infty :\Leftrightarrow \mathbf{[}P\mathbf{]}_\infty \preceq^+ \mathbf{[}Q\mathbf{]}_\infty\ .$$

By construction, the relation $\leq$ is reflexive and transitive (even though $\preceq$ in general is not transitive). Lemma 3 yields that it also is antisymmetric, and hence a partial order. Namely, if $[\![P]\!]_\infty \leq [\![Q]\!]_\infty$ and $[\![Q]\!]_\infty \leq [\![P]\!]_\infty$, then $BD(P) = BD(Q)$, so $P \approx_{\mathrm{S}}^{\infty} Q$, implying $[\![P]\!]_\infty = [\![Q]\!]_\infty$.

Now maximality according to Definition 16 is simply maximality w.r.t. $\leq$:

$$[\![P]\!]_\infty \text{ is maximal iff } \nexists [\![P']\!]_\infty.\ [\![P]\!]_\infty \leq [\![P']\!]_\infty \wedge [\![P]\!]_\infty \neq [\![P']\!]_\infty.$$

The following lemma tells how the above notions of maximality form a hierarchy.

Lemma 5. Let N be a net and P a process thereof.

1. If $[\![P]\!]_\infty$ is run-maximal, it is maximal.
2. If $[\![P]\!]_\infty$ is maximal, it is weakly maximal.

Proof. "1": This follows since $[\![P]\!]_\infty \leq [\![Q]\!]_\infty \Rightarrow BD(P) \subseteq BD(Q)$ by Lemma 3.

"2": Assume $[\![P]\!]_\infty$ is maximal. By Lemma 2 in [GGS11a], which follows via Zorn's Lemma, there exists some maximal Q with $P \leq Q$. Since $[\![P]\!]_\infty$ is maximal we have $Q \approx_{\mathrm{S}}^{\infty} P$ and Q is a maximal process within $[\![P]\!]_\infty$. □

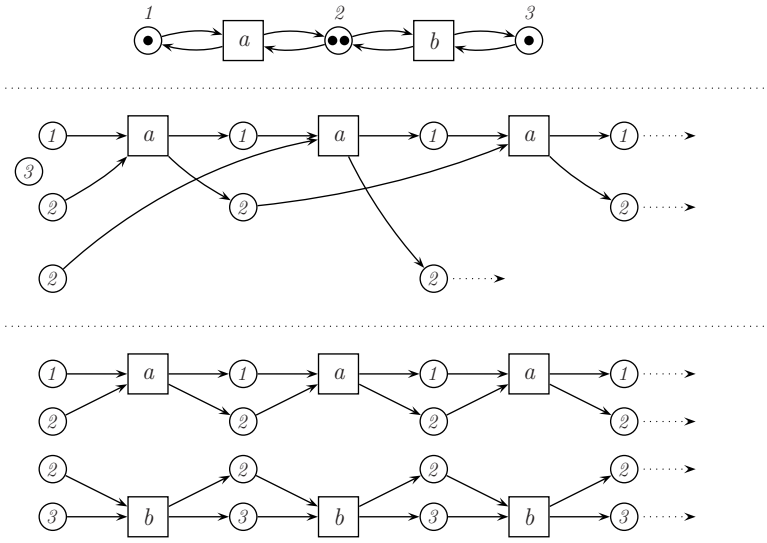

Fig. 4. A net and two weakly maximal processes thereof.

The three notions of maximality are all distinct. The first process depicted in Fig. 4 is an example of a weakly maximal BD-process that is not maximal. Namely, the process itself cannot be extended (for none of the tokens in place 2 will in the end come to rest), but the process is swapping equivalent with the top half of the second process (using only one of the tokens in place 2), which can be extended with the bottom half.

The process depicted in Fig. 5 is an example of a BD-process $[\![P]\!]_\infty$ which is maximal, but not run-maximal. It is maximal, because no matter how it is

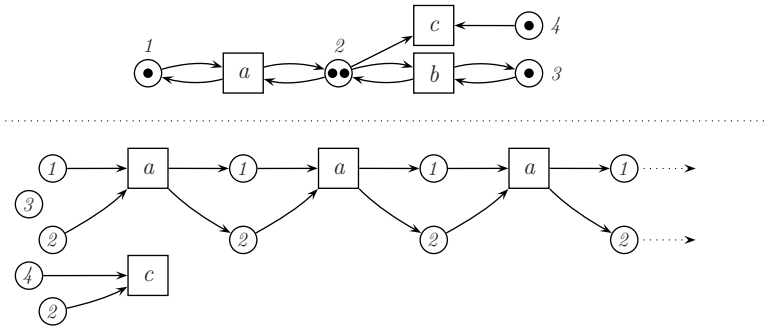

Fig. 5. A net and a maximal process thereof.

swapped, at some point the c-transition will fire, and after that the only token left in place 2 will be in use forever, making it impossible to extend the process with any (b-)transition. It is not run-maximal, as the set of all finite processes of N constitutes a larger run. Note that every two finite processes of N mapping to the same multiset of transitions are swapping equivalent.

The following lemmas show that for countable conflict-free nets maximality and run-maximality coincide, and that for structural conflict nets all three notions of maximality coincide.

Lemma 6. Let N be a countable binary-conflict–free net, and P be a GR-process of N.

If $\mathbf{[}P\mathbf{]}_\infty$ is maximal, then $\mathbf{[}P\mathbf{]}_\infty$ is run-maximal.

Lemma 7. Let N be a structural conflict net, and P be a GR-process of N.

If $\mathbf{[}P\mathbf{]}_\infty$ is weakly maximal, then $\mathbf{[}P\mathbf{]}_\infty$ is run-maximal.

Finally, we are able to show, using Theorem 2, that a countable, binary-conflict–free net has only one maximal BD-process. In case of a conflict-free structural conflict net we can do the stronger statement that it has only one weakly maximal BD-process, i.e. only one GR-process up to swapping equivalence.

Lemma 8. Let N be a binary-conflict–free net.

(1) N has at most one run-maximal BD-process.
(2) If N moreover is countable, then it has exactly one run-maximal BD-process.

Proof. Suppose N had two run-maximal BD-processes $\mathbf{[}P\mathbf{]}_\infty$ and $\mathbf{[}P'\mathbf{]}_\infty$. Then $BD(P)$ and $BD(P')$ are maximal BD-runs. By Theorem 2 N has only one maximal BD-run. Hence $BD(P) = BD(P')$ and thus $\mathbf{[}P\mathbf{]}_\infty = \mathbf{[}P'\mathbf{]}_\infty$.

Now assume that N is countable. By Theorem 2, N has a maximal BD-run $\mathcal{R}$. By Lemma 2 there is a process P with $BD(P) = \mathcal{R}$. By Definition 16 $\mathbf{[}P\mathbf{]}_\infty$ is run-maximal, so at least one run-maximal BD-process exists. □

Theorem 4. Let N be a countable binary-conflict–free net.

N has exactly one maximal BD-process.

Proof. By Lemmas 5 and 6 the notions of maximality and run-maximality coincide for N, and the result follows from Lemma 8. □

The net of Fig. 3 is an example of an uncountable binary-conflict–free net without any maximal or run-maximal BD-process.

Theorem 5. Let N be a conflict-free structural conflict net.

N has exactly one weakly maximal BD-process, i.e. exactly one maximal GR-process up to $\approx_{\mathrm{S}}^{\infty}$.

Proof. By Lemmas 5 and 7 the three maximality notions coincide for N, and the "at most one"-direction follows from Lemma 8.

Surely, N has at least one process (with an empty set of transitions). By Lemma 2 in [GGS11a], which in turn invokes Zorn's lemma, every GR-process is a prefix of a maximal GR-process. Hence N has a maximal GR-process, and thus a maximal GR-process up to $\approx_{\mathrm{S}}^{\infty}$. □

The assumption that N is a structural conflict net is essential in Theorem 5. The net in Fig. 4 is countable (even finite) and conflict-free, yet has multiple maximal GR-process up to $\approx_{\mathrm{S}}^{\infty}$.

We can now justify BD-processes as an abstract notion of process for structural conflict nets since we obtain exactly one maximal abstract process if and only if the underlying net is conflict-free.

Corollary 1. Let N be a structural conflict net.

N is conflict-free iff N has exactly one maximal BD-process, which is the case iff N has exactly one maximal GR-process up to $\approx_{\mathrm{S}}^{\infty}$.

Proof. All three notions of maximality coincide for structural conflict nets according to Lemma 7 and Lemma 5.

"⇒": By Theorem 5.

"⇐": By Theorem 3. □

References

BD87. E. Best & R.R. Devillers (1987): *Sequential and concurrent behaviour in Petri net theory. Theoretical Computer Science* 55(1), pp. 87–136, doi:10.1016/0304-3975(87)90090-9. See also: E. Best and R.R. Devillers (1987): *Interleaving and Partial Orders in Concurrency: A Formal Comparison.* In M. Wirsing, editor: *Formal Description of Programming Concepts III*, 1987, pp. 299–321, North-Holland.

BMO09. K. Barylska, Ł. Mikulski & E. Ochmański (2009): *Nonviolence Petri Nets.* In: *Proceedings Workshop on Concurrency, Specification and Programming (CS&P 2009)*, pp. 50–59.

DMM89. P. Degano, J. Meseguer & U. Montanari (1989): *Axiomatizing Net Computations and Processes.* In: Proceedings Fourth Annual Symposium on *Logic in Computer Science*, LICS'89, Pacific Grove, CA, IEEE, pp. 175–185. See also P. Degano, J. Meseguer & U. Montanari (1996): *Axiomatizing the algebra of net computations and processes.* Acta Informatica 33(5), pp. 641-667, doi:10.1007/BF03036469.

Eng91. J. Engelfriet (1991): *Branching Processes of Petri Nets*. Acta Informatica 28(6), pp. 575–591.

GG01. R.J. van Glabbeek & U. Goltz (2001): *Refinement of actions and equivalence notions for concurrent systems*. Acta Informatica 37(4/5), pp. 229–327, doi:10.1007/s002360000041.

GGS11a. R.J. van Glabbeek, U. Goltz & J.-W. Schicke (2011): *Abstract Processes of Place/Transition Systems*. Information Processing Letters 111(13), pp. 626 – 633, doi:10.1016/j.ipl.2011.03.013.

GGS11b. R.J. van Glabbeek, U. Goltz & J.-W. Schicke (2011): *On Causal Semantics of Petri Nets*. Technical Report 2011-06, TU Braunschweig.

Gla05. R.J. van Glabbeek (2005): *The Individual and Collective Token Interpretations of Petri Nets*. In M. Abadi & L. de Alfaro, editors: Proceedings CONCUR 2005 — 16th International Conference on Concurrency Theory, LNCS 3653, Springer, pp. 323–337, doi:10.1007/11539452_26.

Gol86. U. Goltz (1986): *How Many Transitions may be in Conflict?* Petri Net Newsletter 25, pp. 4–9.

Gol87. U. Goltz (1987): *On condition/event representations of place/transition nets*. In: Concurrency and Nets: Advances in Petri Nets, LNCS, Springer, pp. 217–231.

GP95. R.J. van Glabbeek & G.D. Plotkin (1995): *Configuration Structures (extended abstract)*. In D. Kozen, editor: Proceedings LICS'95, pp. 199–209. See also R.J. van Glabbeek & G.D. Plotkin (2009): *Configuration Structures, Event Structures and Petri Nets*, Theoretical Computer Science 410(41) (2009), pp. 4111-4159, doi:10.1016/j.tcs.2009.06.014.

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: MFCS '89: Proceedings Mathematical Foundations of Computer Science 1989, 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: Proceedings Third Annual Symposium 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, doi:10.1016/0304-3975(81)90112-2.

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, doi:10.1016/0304-3975(91)90084-F.