Abstract Processes and Conflicts in Place/Transition Systems*

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

[1] Data61, CSIRO, 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

Abstract. For one-safe Petri nets or condition/event-systems, a *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. Goltz and Reisig have generalised this concept for nets where places carry multiple tokens, by distinguishing tokens according to their causal history. However, this so-called *individual token interpretation* is often considered too detailed. Here we identify a subclass of Petri nets, called *structural conflict nets*, where no interplay between conflict and concurrency due to token multiplicity occurs. For this subclass, we define abstract processes as equivalence classes of Goltz-Reisig processes. We justify this approach by showing that there is a largest 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. We propose and justify a solution for a subclass of Petri nets, called *structural conflict nets*.

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. Petri 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.

However, the most frequently used class of Petri nets are nets where places may carry arbitrarily many tokens, or a certain maximal number of tokens when

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

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 30 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 system 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.

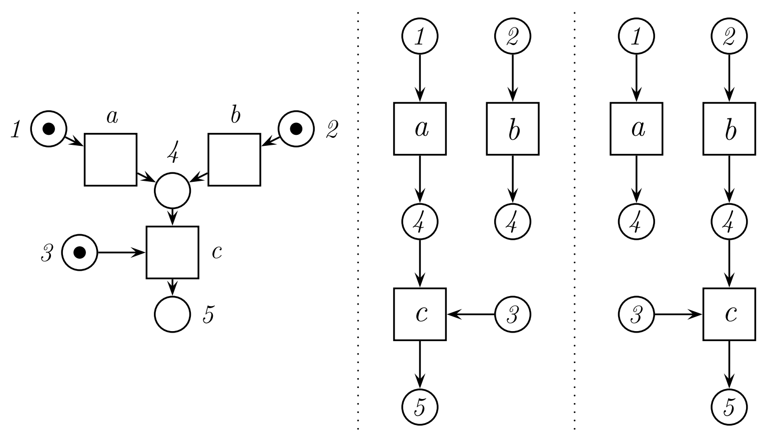

Fig. 1. A net N with its two maximal GR-processes. The correspondence between elements of the net and their occurrences in the processes is indicated by labels.

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, in the sense that there are no choices to resolve, it should intuitively have only one complete run, yet it may have multiple maximal GR-processes [Gol86]. This phenomenon 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 choice, as in the interpretation underlying GR-processes, 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 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 [HKT95] 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]".

As observed by Vogler [Vog90] (as a consequence of Corollary 5.6 therein), it can be argued that BD-processes are not fully satisfying as abstract processes for general P/T systems. To illustrate this result, we recall in Fig. 2 an example due to Ochmański [Och89] — 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 conforms to the collective token interpretation and meets the requirement that for this net there is more than one 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

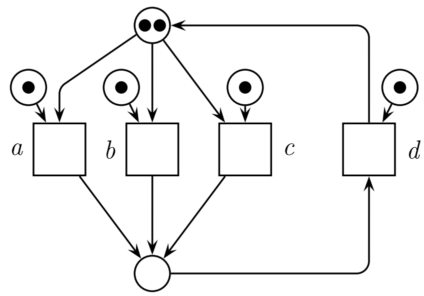

Fig. 2. A net with only a single process up to swapping equivalence.

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. However, we show that BD-processes are completely adequate on a subclass of P/T-systems — proposed in [GGS11a] — 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 in [GGS11a] that, for this subclass of P/T systems, we obtain some finite BD-processes without a common extension whenever the net contains a conflict. The proof of this result is quite involved; it was achieved by using an alternative characterisation of BD-processes via firing sequences from [BD87]. As we point out in Section 5, this result implies that a structural conflict net with a largest BD-process must be conflict-free.

In this paper, we will show the reverse direction of this result, namely that a structural conflict net has a largest BD-process if the net is conflict-free. We then have established that there is a largest 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 BD-processes, as well as a natural partial order on BD-processes that gives rise to the notion of a largest BD-process. Section 4 recalls the concept of conflict in P/T systems and defines structural conflict nets.[1] In Section 5 we recall a result from [GGS11a] that implies that a structural conflict net featuring any conflict cannot have a largest BD-process. In Section 6 we prove the converse, that a conflict-free structural conflict net does have a largest BD-process. Finally, Section 7 reformulates (and slightly strengthens) our result in the terminology of [GGS11a], where we did not employ a partial order on BD-processes, and hence no canonical notion of a largest BD-process. We show that a structural conflict net is conflict-free iff it has a unique maximal GR-process up to swapping equivalence.

[1] The material in Sections 2, 3.1 and 4 follows closely the presentation in [GGS11a], but needs to be included to make the paper self-contained.

The results of this paper, together with a slightly extended overview on the existing approaches on semantics of Petri nets, were previously announced in [GGS11b], with proofs in an accompanying technical report. Our current proofs are conceptually much simpler, as they are carried out directly on BD-processes, rather than via the auxiliary concepts of BD-runs and FS-runs. This became possible after turning swapping equivalence into a preorder on BD-processes, simply by employing only one of the two symmetric clauses defining this relation. That idea stems from Walter Vogler [personal communication, 20-11-2012], whom we gratefully acknowledge.

Additionally, we thank the referees for their very detailed and helpful comments.

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 $A\colon X \rightarrow \mathbb{N}$, 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 $(k \cdot A)(x) := k \cdot A(x)$.
- The function $\emptyset : X \rightarrow \mathbb{N}$, 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.
- A function $\pi : X \rightarrow Y$ extends to multisets $A \in \mathbb{N}^X$ by $\pi(A)(y) = \sum_{y=\pi(x)} A(x)$. In this paper, this sum will always turn out to be finite.

Two multisets $A : X \rightarrow \mathbb{N}$ and $B : Y \rightarrow \mathbb{N}$ 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*),
- $F : ((S{\times}T) \cup (T{\times}S)) \rightarrow \mathbb{N}$ (the *flow relation* including *arc weights*), and
- $M_0 : S \rightarrow \mathbb{N}$ (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.[2] 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.

Definition 3. Let $N = (S, T, F, M_0)$ be a net and $x \in S \cup T$.

The multisets ${}^\bullet x,\ x^\bullet : S \cup T \rightarrow \mathbb{N}$ 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 $X : S \cup T \rightarrow \mathbb{N}$ as usual, by ${}^\bullet X := \sum_{x \in S \cup T} X(x) \cdot {}^\bullet x$ and $X^\bullet := \sum_{x \in S \cup T} X(x) \cdot x^\bullet$.

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 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
- $M' = (M - {}^\bullet G) + G^\bullet$.

[2] This is a presentational alternative for the common approach of having at most one arc from x to y, labelled with the *arcweight* $F(x, y) \in \mathbb{N}$.

We may leave out the subscript N if clear from context. For a word $\sigma = t_1 t_2 \ldots t_n \in T^*$ we write $M \xrightarrow{\sigma} M'$ for

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

When omitting σ or M' we always mean it to be existentially quantified.

Note that steps are (finite) multisets, thus allowing self-concurrency. Also note that $M \xrightarrow{\{t,u\}}$ implies $M \xrightarrow{tu}$ and $M \xrightarrow{ut}$.

3 Processes of place/transition systems

We now define two notions of a process of a net, modelling a run of the represented system on two levels of abstraction.

3.1 GR-processes

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. A conflict present in the original net is represented by the existence of multiple processes, each representing one possible way to decide the conflict.

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 $\{(x, y) \mid \mathcal{F}(x, y) > 0\}$,
  - and $\{t \in \mathcal{T} \mid (t, u) \in \mathcal{F}^+\}$ is finite for all $u \in \mathcal{T}$.
- $\pi : \mathcal{S} \cup \mathcal{T} \rightarrow S \cup T$ 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
  - $\forall t \in \mathcal{T}, s \in S.\ F(s, \pi(t)) = |\pi^{-1}(s) \cap {}^\bullet t| \wedge F(\pi(t), s) = |\pi^{-1}(s) \cap t^\bullet|$, i.e. $\forall t \in \mathcal{T}.\ \pi({}^\bullet t) = {}^\bullet \pi(t) \wedge \pi(t^\bullet) = \pi(t)^\bullet$.

P is called *finite* if $\mathcal{T}$ is finite. The *end of* P is defined as $P^\circ = \{s \in \mathcal{S} \mid s^\bullet = \emptyset\}$.

Let $\mathrm{GR}(N)$ (resp. $\mathrm{GR}_{\mathit{fin}}(N)$) denote the collection of (finite) GR-processes of N.

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 $\pi' = \pi \restriction (\mathcal{S}' \cup \mathcal{T}')$.
- 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, denoted $P \restriction \mathcal{T}'$.

In [Pet77,GSW80,GR83] processes were defined without requiring the third condition on $\mathcal{N}$ from Definition 5. Goltz and Reisig [GR83] observed that certain processes did not correspond with runs of systems, and proposed to restrict the notion of a process to those that can be approximated by finite processes [GR83, end of Section 3]. This is the role of the third condition on $\mathcal{N}$ in Definition 5; it is equivalent to requiring that each transition occurs in a finite prefix. In [Pet77,GSW80,GR83] only processes of finite nets were considered. For those processes, the requirement of *discreteness* proposed in [GR83] is equivalent with imposing the third condition on $\mathcal{N}$ in Definition 5 [GR83, Theorem 2.14].

Two processes $P = (\mathcal{N}, \pi)$ and $P' = (\mathcal{N}', \pi')$ are *isomorphic*, notation $P \cong P'$, 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 $x, y \in \mathcal{S} \cup \mathcal{T}$.

3.2 BD-processes

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

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

If P is the first process depicted in Fig. 1, with p and q the two places that are mapped to place 4 of the underlying net N, then $\mathrm{swap}(P, p, q)$ is the second process of Fig. 1. The transformation simply swaps the arcs leaving p and q.

Definition 8.

- Two processes P and Q of the same net are *one step swapping equivalent* ($P \equiv_1 Q$) iff $\mathrm{swap}(P, p, q)$ is isomorphic to Q for some places p and q.
- We write $\equiv_1^*$ for the reflexive and transitive closure of $\equiv_1$.

By taking $p{=}q$ in Definition 8 one finds that $P{\equiv_1}P$ for any non-empty process P. In [BD87, Definition 7.8] swapping equivalence—there denoted $\equiv_1^\infty$—is defined in terms of *reachable B-cuts*. Using [BD87, Definition 3.14] this definition can be rephrased as follows:

Definition 9. Let N be a net, and $P, Q \in \mathrm{GR}(N)$.

Then $P \equiv_1^\infty Q$ iff

$$\forall P'' \in \mathrm{GR}_{\mathit{fin}}(N), P'' \leq P.\ \exists Q' \equiv_1^* Q,\ Q'' \leq Q'.\ P'' \cong Q'' \tag{1}$$

and, vice versa,

$$\forall Q'' \in \mathrm{GR}_{\mathit{fin}}(N), Q'' \leq Q.\ \exists P' \equiv_1^* P,\ P'' \leq P'.\ P'' \cong Q''\ .$$

In [BD87, Theorem 7.9] (as well as below) it is shown that $\equiv_1^\infty$ is an equivalence relation on GR-processes. Trivially, $\equiv_1^*$ is included in $\equiv_1^\infty$.

Definition 10.

We call a $\equiv_1^\infty$-equivalence class of GR-processes a *BD-process*, and write $[\![P]\!]_\infty$.

To support the idea that $\equiv_1^\infty$ is a natural equivalence relation on GR-processes, an alternative characterisation of $\equiv_1^\infty$ is presented in Section 3.3 below.

In order to establish concepts of a maximal and of a largest BD-process, we turn $\equiv_1^\infty$ into a preorder by focusing on only one of the two clauses of Definition 9 (formulated differently).

Definition 11. Let N be a net, and $P, Q \in \mathrm{GR}(N)$.

Then $P \sqsubseteq_1^\infty Q$ iff

$$\forall P'' \in \mathrm{GR}_{\mathit{fin}}(N), P'' \leq P.\ \exists P', Q' \in \mathrm{GR}_{\mathit{fin}}(N).\ P'' \leq P' \equiv_1^* Q' \leq Q. \tag{2}$$

We proceed to show that (2) is equivalent to (1) and that $\sqsubseteq_1^\infty$ is a preorder.

For $P, Q \in \mathrm{GR}(N)$ write $P \sim_S Q$ if there are places p and q such that $\mathrm{swap}(P, p, q) = Q$ (or equivalently $\mathrm{swap}(Q, p, q) = P$). Clearly, the operations of swapping two places in a process, and of bijectively renaming all places and transitions, commute:

Observation 1 $\exists P' \in \mathrm{GR}(N).\ P \cong P' \sim_S Q \ \Leftrightarrow\ \exists Q' \in \mathrm{GR}(N).\ P \sim_S Q' \cong Q.$

The same holds for the operations of taking a prefix and of bijectively renaming all places and transitions:

Observation 2 $\exists P' \in \mathrm{GR}(N).\ P \cong P' \leq Q \ \Leftrightarrow\ \exists Q' \in \mathrm{GR}(N).\ P \leq Q' \cong Q.$

Moreover, instead of first swapping two places p and q in a process and then extending the resulting process, we can just as well first extend and then swap:

Observation 3 $\exists P' \in \mathrm{GR}(N).\ P \sim_S P' \leq Q \ \Rightarrow\ \exists Q' \in \mathrm{GR}(N).\ P \leq Q' \sim_S Q.$

This implication can in general not be reversed, since it could be that out of two swapped places p and q occurring in Q and Q' only one occurs in P.

Lemma 1. If $P \leq Q' \sim_S Q$ for some $P \in \mathrm{GR}_{\mathit{fin}}(N)$ and $Q', Q \in \mathrm{GR}(N)$, then there are $P', P'' \in \mathrm{GR}_{\mathit{fin}}(N)$ with $P \leq P' \sim_S P'' \leq Q$.

Proof. Let $Q = \text{swap}(Q', p, q)$ for certain places p and q in Q'. Take a finite prefix P' of Q' that includes P as well as p and q. Then $P \leq P' \leq Q'$. Let $P'' := \text{swap}(P', p, q)$. Then $P' \sim_S P'' \leq Q$. □

$P \equiv_1 Q$ is defined by $\exists Q'.\ P \sim_S Q' \cong Q$. Using that $\sim_S$ is reflexive on nonempty processes we have:

Observation 4 $\cong \subseteq \equiv_1^*$.

So, using Observation 2, (1) can be reformulated as

$$\forall P'' \in \text{GR}_{\mathit{fin}}(N), P'' \leq P.\ \exists Q' \equiv_1^* Q.\ P'' \leq Q'.$$

From Observations 2 and 3 we obtain:

Corollary 1. $\exists P' \in \text{GR}(N).\ P \equiv_1^* P' \leq Q \Rightarrow \exists Q' \in \text{GR}(N).\ P \leq Q' \equiv_1^* Q.$ □

Likewise, from Observation 2 and 3 and Lemma 1 we obtain:

Corollary 2. If $P \leq Q' \equiv_1^* Q$ for some $P \in \text{GR}_{\mathit{fin}}(N)$ and $Q', Q \in \text{GR}(N)$, then there are $P', P'' \in \text{GR}_{\mathit{fin}}(N)$ with $P \leq P' \equiv_1^* P'' \leq Q$. □

Together, Corollaries 1 and 2 imply the equivalence of (1) and (2). Corollary 1, in combination with the transitivity of $\leq$ and $\equiv_1^*$, implies the transitivity of $\sqsubseteq_1^\infty$. Moreover, by definition $\sqsubseteq_1^\infty$ is reflexive.

Corollary 3. $\sqsubseteq_1^\infty$ is a preorder on $\text{GR}(N)$. Hence $\equiv_1^\infty$ is an equivalence relation.

It follows that $\sqsubseteq_1^\infty$ induces a partial order on BD-processes, and thereby concepts of a *maximal* and a *largest* BD-process.[3]

3.3 An alternative characterisation of swapping equivalence

Let $P \in \text{GR}(N)$. The set $BD(P)$ of *finite BD-approximations* of P is the smallest set of finite GR-processes that contains all finite prefixes of P and is closed under $\equiv_1$ and taking prefixes. (By Observation 4 it therefore is also closed under $\cong$.) Thus, it is the smallest subset of $\text{GR}(N)$ satisfying

- if $P' \in \text{GR}_{\mathit{fin}}(N)$ and $P' \leq P$ then $P' \in BD(P)$,
- if $P' \equiv_1 Q \in BD(P)$ then $P' \in BD(P)$,
- if $P' \leq Q \in BD(P)$ then $P' \in BD(P)$.

Proposition 1. $P \sqsubseteq_1^\infty Q \Leftrightarrow BD(P) \subseteq BD(Q)$. So $P \equiv_1^\infty Q \Leftrightarrow BD(P) = BD(Q)$.

Proof. $BD(P) = \{P''' \in \text{GR}_{\mathit{fin}}(N) \mid \exists P'', P' \in \text{GR}_{\mathit{fin}}(N).\ P''' \leq P'' \equiv_1^* P' \leq P\}$ by Corollary 1. Using this, the result follows directly from Definition 11. □

[3] A *preorder* is a relation that is reflexive and transitive; it is an *equivalence relation* if it moreover is symmetric, and a *partial order* if it moreover is antisymmetric. Given a preorder $\sqsubseteq$, its *kernel* is the equivalence relation $\equiv$ defined by $P \equiv Q$ iff $P \sqsubseteq Q \wedge Q \sqsubseteq P$. Moreover, the *induced* partial order $\leq$ on the $\equiv$-equivalence classes is given by $[P] \leq [Q]$ iff $P \sqsubseteq Q$; it is easy to check that this is independent of the choices of representatives P and Q within the $\equiv$-equivalences classes $[P]$ and $[Q]$.

Here $\equiv_1^\infty$ is the kernel of $\sqsubseteq_1^\infty$, and "maximal" or "largest" refers to the induced partial order on BD-processes.

4 Conflicts in place/transition systems

We recall the notion of conflict introduced in [Gol86]. It formalises the notion of conflict alluded to in [Rei85, p. 23].

Definition 12. 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 $\neg M \xrightarrow{G} \quad \wedge \quad \forall t \in G.\ M \xrightarrow{G \restriction \{t\}}$.
- N is *(semantically) 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 $M_0 \longrightarrow M$.

Thus, N is binary-conflict–free iff whenever two different transitions t and u are enabled at a reachable marking M, then also the step $\{t, u\}$ is enabled at M. The above concept of (semantic) conflict-freeness formalises the intuitive notion that there are no choices to resolve.

Remark: In a net such as displayed in Fig. 3, the multiset $\{t, t\}$ is never enabled. For this reason the multiset $\{t, t, u\}$ does not count as being in conflict, even though it is never enabled. However, its subset $\{t, u\}$ is in conflict.

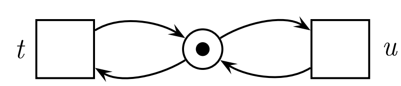

Fig. 3. A net which is persistent but not binary-conflict–free

A number of alternative concepts of conflict and conflict-freeness have been contemplated in the Petri net community.

A Petri net N is called *persistent* [KM69,LR78] if for every marking M with $M_0 \longrightarrow M$ and every $t, u \in T$ with $t \neq u$, $M \xrightarrow{t}$ and $M \xrightarrow{u}$, we have $M \xrightarrow{tu}$; in other words, if any transition u that is enabled in a reachable marking will still be enabled after firing any other transition t. Trivially, a net that is binary-conflict–free is also persistent. The net of Fig. 3, on the other hand, is persistent but not binary-conflict–free.

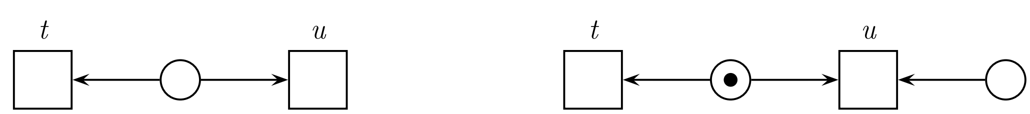

Fig. 4. Two nets with structural conflict, but no choices to resolve.

A pair of different transitions in a net that share a preplace can be called a *structural conflict*. As illustrated in Fig. 4, the presence of a structural conflict does not imply that there are choices to resolve. A net that is free of structural conflicts is certainly conflict-free, but Fig. 4 shows that the reverse does not hold.

A triple (M,t,u) of a reachable marking M and two different transitions t and u with $M \xrightarrow{t}$, $M \xrightarrow{u}$ and ${}^\bullet t \cap {}^\bullet u \neq \emptyset$ could be called a *reachable structural conflict*. This constitutes a middle ground between semantic and structural conflict. The nets of Fig. 4 do not have a reachable structural conflict. However, the net of Fig. 5, although semantically conflict-free, does have a reachable structural conflict.

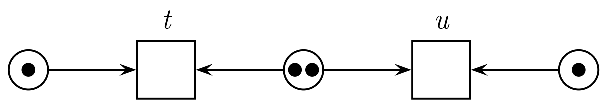

Fig. 5. A net with a reachable structural conflict, but no choices to resolve.

Landweber and Robertson [LR78] define a Petri net to be conflict-free "if every place which is an input of more than one transition is on a self-loop with each such transition." This is an extension of the concept structural conflict-freeness that is closer to persistence. It classifies the net of Fig. 3 as conflict-free and the nets of Fig. 4 as having conflicts. Hence, this notion, just as persistence and structural conflict-freeness, does not formalise the intuitive concept "no choices to resolve".

We proposed in [GGS11a] a class of P/T systems where (semantic) conflict-freeness coincides with the absence of reachable structural conflicts. We called this class of nets *structural conflict nets*. For a net to be a structural conflict net, we required that two transitions sharing a preplace will never occur both in one step.

Definition 13. 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 reachable 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.

Trivially, the class of structural conflict nets includes the class of safe nets, in which no reachable marking assigns multiple tokens to the same place. It also includes the non-safe net of Fig. 1, as well as the *buffer synchronised systems of sequential machines* from [Rei82] and the *locally sequential globally asynchronous nets (LSGA nets)* of [GGS13], in which asynchronous communication is modelled by buffer-places between sequential components that may collect arbitrarily many tokens.

5 A structural conflict net having a largest BD-process is conflict-free

The result announced in this section—that each structural conflict net having a $\sqsubseteq_1^\infty$-largest BD-process must be conflict-free—is in essence obtained in [GGS11a]. However, there we had not defined the order $\sqsubseteq_1^\infty$, and thus neither the corresponding notion of a $\sqsubseteq_1^\infty$-largest BD-process. Instead we used different terminology, and the work in this section merely consists of relating the terminology of [GGS11a] to the one of the present paper.

In [GGS11a] a *partial BD-run* of a net N is defined as a $\equiv_1^*$-equivalence class of finite GR-processes of N.[4] Let $[\![P]\!]$ be the partial BD-run containing P. The prefix/extension relation $\leq$ on $\mathrm{GR}_{\mathit{fin}}(N)$ from Definition 6 is lifted to partial BD-runs by $[\![P']\!] \leq [\![P]\!]$ iff $P' \equiv_1^* Q' \leq Q \equiv_1^* P$ for some $Q', Q \in \mathrm{GR}_{\mathit{fin}}(N)$. By Corollary 1, $\leq$ is a partial order on partial BD-runs, and $[\![P']\!] \leq [\![P]\!]$ iff $P' \leq Q \equiv_1^* P$ for some $Q \in \mathrm{GR}_{\mathit{fin}}(N)$. Moreover, $[\![P']\!] \leq [\![P]\!]$ iff $P' \sqsubseteq_1^\infty P$.

In [GGS11a] a *BD-run* of a net N is defined as a prefix-closed and directed set of partial BD-runs of N. Here we define the notion of a *collapsed BD-run*, or $\mathrm{BD}^\dagger$-run.

Definition 14. A *$BD^\dagger$-run* of a net N is a subset $\mathcal{R}$ of $\mathrm{GR}_{\mathit{fin}}(N)$ that is prefix-closed and closed under $\equiv_1^*$, and satisfies

$$P, Q \in \mathcal{R} \Rightarrow \exists P', Q' \in \mathcal{R}.\ P \leq P' \equiv_1^* Q' \geq Q\ .$$

Note that a $\mathrm{BD}^\dagger$-run is a set of finite GR-processes, whereas a BD-run is a set of sets of finite GR-processes. We proceed to show that the two notions have the same information content. For a BD-run R, let $R^\dagger := \{P \in \mathrm{GR}_{\mathit{fin}}(N) \mid [\![P]\!] \in R\}$. Trivially, $R^\dagger$ is a $\mathrm{BD}^\dagger$-run. Moreover, $R_1 \subseteq R_2$ implies $R_1^\dagger \subseteq R_2^\dagger$.

Conversely, for $\mathcal{R}$ a $\mathrm{BD}^\dagger$-run, let $[\![\mathcal{R}]\!] := \{[\![P]\!] \mid P \in \mathcal{R}\}$. Trivially, $[\![\mathcal{R}]\!]$ is a BD-run. Moreover, $\mathcal{R}_1 \subseteq \mathcal{R}_2$ implies $[\![\mathcal{R}_1]\!] \subseteq [\![\mathcal{R}_2]\!]$. Also note that $[\![R^\dagger]\!] = R$ for any BD-run R, and $[\![\mathcal{R}]\!]^\dagger = \mathcal{R}$ for any $\mathrm{BD}^\dagger$-run $\mathcal{R}$. Thus we have a $\subseteq$-preserving bijective correspondence between BD-runs and $\mathrm{BD}^\dagger$-runs. It follows that a net has a unique maximal BD-run iff it has a unique maximal $\mathrm{BD}^\dagger$-run.[5]

[GGS11a, Section 5] defines the concept of an *FS-run*—an FS run is a certain set of sets of firing sequences—and establishes a $\subseteq$-preserving bijective correspondence between FS-runs and BD-runs. It follows that a net has a unique maximal FS-run iff it has a unique maximal BD-run.

The set of finite prefixes of a GR-process P is directed: for $P_1, P_2 \in \mathrm{GR}_{\mathit{fin}}(N)$ with $P_1 \leq P$ and $P_2 \leq P$, there is a process $P_3 \in \mathrm{GR}_{\mathit{fin}}(N)$ with $P_1 \leq P_3 \leq P$ and $P_2 \leq P_3$. Just take as transitions of P_3 the union of the transitions from P that occur in P_1 or P_2.

[4] It is easy to see that on finite GR-processes the relations $\equiv_1^\infty$ and $\equiv_1^*$ coincide. Hence a partial BD-run is the same as a finite BD-process, i.e., an equivalence class $[\![P]\!]_\infty$ with P a finite GR-process. We do not use this fact further on.

[5] Exactly as in the proof of Lemma 5 in the next section it follows that each BD-run is a prefix of a maximal BD-run. Hence a unique maximal BD-run is the same as largest BD-run. The same applies to $\mathrm{BD}^\dagger$-runs.

Lemma 2. If a net has a $\sqsubseteq_1^\infty$-largest BD-process then it has a largest BD†-run.

Proof. Let $[\![P]\!]_\infty$ be the $\sqsubseteq_1^\infty$-largest BD-process of a net N. We claim that the set of *all* finite GR-processes of N is a BD†-run. Clearly, it is then also the largest.

Trivially, $\mathrm{GR}_{\mathit{fin}}(N)$ is prefix-closed and closed under $\equiv_1^*$. Now suppose $P_1, Q_1 \in \mathrm{GR}_{\mathit{fin}}(N)$. Since $P_1 \sqsubseteq_1^\infty P$ and $P_1 \leq P_1$ one has $P_1 \leq P_2 \equiv_1^* P_3 \leq P$ for some $P_2, P_3 \in \mathrm{GR}_{\mathit{fin}}(N)$. Likewise $Q_1 \leq Q_2 \equiv_1^* Q_3 \leq P$ for some $Q_2, Q_3 \in \mathrm{GR}_{\mathit{fin}}(N)$. Using that the set of prefixes of P is directed, let $P_4 \in \mathrm{GR}_{\mathit{fin}}(N)$ be such that $P_3 \leq P_4 \leq P$ and $Q_3 \leq P_4$. Now Corollary 1 yields $P_1 \leq\equiv_1^* P_4 \equiv_1^* \geq Q_1$, which needed to be established. □

Theorem 1. Let N be a structural conflict net.

If N has a $\sqsubseteq_1^\infty$-largest BD-process then N is conflict-free.

Proof. [GGS11a, Theorem 6] says that if a structural conflict net N has exactly one maximal FS-run then N is conflict-free.

Now suppose N has a $\sqsubseteq_1^\infty$-largest BD-process. By Lemma 2 it has a unique maximal BD†-run. Hence it has a unique maximal BD-run, and a unique maximal FS-run. It follows that N is conflict-free. □

Note that [GGS11a, Theorem 6] can be reformulated as saying that a structural conflict net N that is not conflict-free fails to have a unique maximal BD-run. This implies that the set of all partial BD-runs of N fails to be a BD-run, and must hence fail to be directed. This in turn implies that there are two finite BD-processes of N without a common extension.

6 A conflict-free structural conflict net has a largest BD-process

In this section we prove the main result of this paper (Theorem 2), namely that each conflict-free structural conflict net has a largest BD-process with respect to the order $\sqsubseteq_1^\infty$. We make use of a labelled transition relation between the processes of a given net. The fact that we are dealing with a structural conflict net is used only at the end of the proof of Theorem 2.

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 GR-processes of a net $N = (S, T, F, M_0)$. Henceforth, we will write $P' \xrightarrow{a} P$ with $a \in T$ a transition of the underlying net, if $P' \leq P$ and $\mathcal{T} = \mathcal{T}' \dot\cup \{t\}$ for some t with $\pi(t) = a$. Let $\mathcal{P}_0(N)$ be the set of *initial processes* of a net N: those with an empty set of transitions. A process $P_0 \in \mathcal{P}_0(N)$ has exactly one place for each token in the initial marking of N; two processes in $\mathcal{P}_0(N)$ differ only in the names of these places. Now for each finite process P of N, having n transitions, there is a sequence $P_0 \xrightarrow{a_1} P_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} P_n$ with $P_0 \in \mathcal{P}_0(N)$ and $P_n = P$.

For $P = ((\mathcal{S}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0), \pi)$ a finite GR-process of a net $N = (S, T, F, M_0)$, we write $\widehat{P}$ for the marking $\pi(P^\circ) \in \mathbb{N}^S$. The following observations describe a *bisimulation* between the above transition relation on the processes of a net, and the one on its markings.

Observation 5 Let $N = (S, T, F, M_0)$ be a net, $a \in T$, and $P, Q \in \mathrm{GR}_{\mathit{fin}}(N)$.

(a) $\mathcal{P}_0(N) \neq \emptyset$ and if $P \in \mathcal{P}_0$ then $\widehat{P} = M_0$.
(b) If $P \xrightarrow{a} Q$ then $\widehat{P} \xrightarrow{a} \widehat{Q}$.
(c) If $\widehat{P} \xrightarrow{a} M$ then there is a Q with $P \xrightarrow{a} Q$ and $\widehat{Q} = M$.
(d) $\widehat{P}$ is *reachable* in the sense that $M_0 \longrightarrow \widehat{P}$. (This follows from (a) and (b).)

Lemma 3. Let $P, P' \in \mathrm{GR}_{\mathit{fin}}(N)$, and a, b transitions of the underlying net N.
If $P \xrightarrow{a} P'$ and $\widehat{P} \xrightarrow{\{a,b\}}$ then $\exists Q, Q'.\ \ P' \xrightarrow{b} Q' \wedge P \xrightarrow{b} Q \xrightarrow{a} Q'$.

Proof. Since $\widehat{P} \xrightarrow{\{a,b\}}$ we have ${}^\bullet a + {}^\bullet b \subseteq \pi(P^\circ)$. Furthermore $\pi(P^\circ \setminus P'^\circ) = {}^\bullet a$. So ${}^\bullet b \subseteq \pi(P^\circ \cap P'^\circ)$. Therefore, there exist Q and Q' as required. □

The following observations are easy to check. For (b) note that $P \equiv_1^* Q$ implies $\widehat{P} = \widehat{Q}$; also compare Corollary 1.

Observation 6 Let P, Q, Q' be finite GR-processes of a net N.

(a) If $P \xrightarrow{a} Q$ and $P \xrightarrow{a} Q'$ then $Q \equiv_1^* Q'$.
(b) If $P \equiv_1^* Q \xrightarrow{a} Q'$ then $P \xrightarrow{a} P' \equiv_1^* Q'$ for some $P' \in \mathrm{GR}_{\mathit{fin}}(N)$.

Lemma 4. Let $N = (S, T, F, M_0)$ be a binary-conflict–free net, $a, b \in T$ with $a \neq b$, and P, P', Q be finite GR-processes of N.
If $P \xrightarrow{a} P'$ and $P \xrightarrow{b} Q$ then $\widehat{P} \xrightarrow{\{a,b\}}$ and $\exists Q'.\ P' \xrightarrow{b} Q' \wedge Q \xrightarrow{a}\equiv_1^* Q'$.

Proof. Suppose $P \xrightarrow{a} P'$ and $P \xrightarrow{b} Q$ with $a \neq b$. We have $M_0 \longrightarrow \widehat{P}$ by Observation 5(d). Moreover, $\widehat{P} \xrightarrow{a} \widehat{P'}$ and $\widehat{P} \xrightarrow{b} \widehat{Q}$ by Observation 5(b). Hence, as N is binary-conflict–free, $\widehat{P} \xrightarrow{\{a,b\}}$. By Lemma 3 there are Q', Q'' with $P' \xrightarrow{b} Q'$ and $P \xrightarrow{b} Q'' \xrightarrow{a} Q'$. By Observation 6(a), $Q \equiv_1^* Q''$, and hence $Q \xrightarrow{a}\equiv_1^* Q'$ by Observation 6(b). □

Lemma 5 ([GGS11a]). Let N be a net.
Every GR-process P of N is a prefix of a maximal GR-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. □

Since the set of GR-processes of N is non-empty by Observation 5(a), this implies that each net has a maximal GR-process.

Theorem 2. Let N be a conflict-free structural conflict net.
Then N has a $\sqsubseteq_1^\infty$-largest BD-process.

Proof. Let P be a maximal GR-process of N—it exists by Lemma 5. We show that $[\![P]\!]_\infty$ is the $\sqsubseteq_1^\infty$-largest BD-process of N, i.e., for each GR-process Q of N one has $Q \sqsubseteq_1^\infty P$. This proof is illustrated in Fig. 6.

Let $BD(P)$ be as defined in Section 3.3. As remarked there, using Corollary 1, $BD(P) = \{P''' \in \mathrm{GR}_{\mathit{fin}}(N) \mid \exists P'', P' \in \mathrm{GR}_{\mathit{fin}}(N).\ P''' \leq P'' \equiv_1^* P' \leq P\}$.

Towards a contradiction, suppose $Q \not\sqsubseteq_1^\infty P$ for some $Q \in \mathrm{GR}(N)$. Then, by Definition 11, there is a finite prefix Q'' of Q with $Q'' \notin BD(P)$. Let Q_0 be a minimal such prefix w.r.t. the prefix order $\leq$ of Definition 6. Q_0 can be written as $((\mathcal{S}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0), \pi)$. Since all initial processes of N are isomorphic, each initial process of N is in $BD(P)$. Hence Q_0 must have a transition.

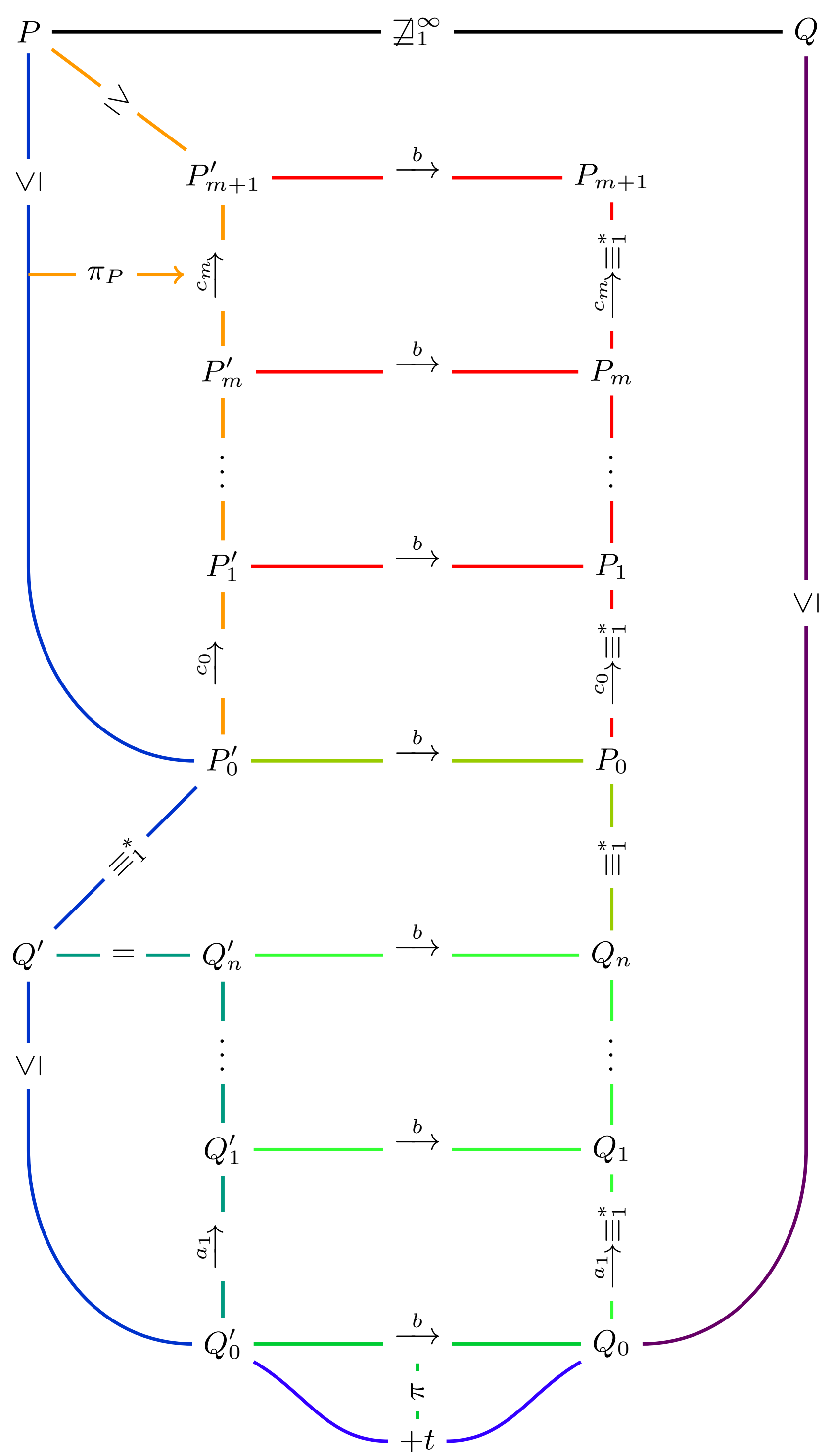

Fig. 6. Illustration of the proof of Theorem 2.

Let t be a maximal element in $\mathcal{T}$ with respect to $\mathcal{F}^+$. Then $Q_0 \upharpoonright (\mathcal{T} \setminus \{t\}) =: Q'_0$ is a process and $Q'_0 \in BD(P)$. Hence there exists finite P'_0, Q' such that $Q'_0 \leq Q' \equiv_1^* P'_0 \leq P$. Moreover, there are $Q'_1, \ldots, Q'_n \in BD(P)$ and transitions $a_1 \ldots, a_n$ of N with $Q'_n = Q'$ and $Q'_{i-1} \xrightarrow{a_i} Q'_i$ for $i = 1, \ldots, n$.

$\pi(t)$ is some transition b of N, so $Q'_0 \xrightarrow{b} Q_0$. We now show by induction on $i \in \{1, \ldots, n\}$ that there are $Q_1, \ldots, Q_n \in \mathrm{GR}_{\mathit{fin}}(N) \setminus BD(P)$ with $Q'_i \xrightarrow{b} Q_i$ and $Q_{i-1} \xrightarrow{a_i}\equiv_1^* Q_i$ for $i = 1, \ldots, n$. Namely, given Q_{i-1}, as $Q_{i-1} \notin BD(P)$ we have $Q_{i-1} \not\equiv_1^* Q'_i \in BD(P)$. Using that $Q'_{i-1} \xrightarrow{a_i} Q'_i$ and $Q'_{i-1} \xrightarrow{b} Q_{i-1}$, this implies $a_i \neq b$ by Observation 6(a). Now Lemma 4 yields a $Q_i \in \mathrm{GR}_{\mathit{fin}}(N)$ such that $Q'_i \xrightarrow{b} Q_i$ and $Q_{i-1} \xrightarrow{a_i}\equiv_1^* Q_i$. As $BD(P)$ is $\equiv_1^*$- and prefix-closed, we have $Q_i \notin BD(P)$.

Since $Q'_n \equiv_1^* P'_0$ and $Q'_n \xrightarrow{b} Q_n$, there is a $P_0 \in \mathrm{GR}_{\mathit{fin}}(N)$ with $P'_0 \xrightarrow{b} P_0$ and $P_0 \equiv_1^* Q_n$, using Observation 6(b). Hence $P_0 \notin BD(P)$.

Now let u be any transition in $P := (\mathcal{N}, \pi_P)$ that is not included in P'_0. Then there are $P'_1, \ldots, P'_{m+1} \leq P$ with $P'_i \xrightarrow{c_i} P'_{i+1}$ for $i = 0, \ldots, m$ and $c_m = \pi_P(u)$. Exactly as above, by induction on i, $b \neq c_i$ for $i = 0, \ldots, m$ and there are $P_1, \ldots, P_{m+1} \in \mathrm{GR}_{\mathit{fin}}(N) \setminus BD(P)$ with $P'_{i+1} \xrightarrow{b} P_{i+1}$ and $P_i \xrightarrow{c_i}\equiv_1^* P_{i+1}$ for $i = 0, \ldots, m$. Moreover, since $P'_m \xrightarrow{c_m} P'_{m+1}$ and $P'_m \xrightarrow{b}$, we have $\widehat{P}'_m \xrightarrow{\{c_m, b\}}$ by Lemma 4. By Observation 5(d) we furthermore have $M_0 \longrightarrow \widehat{P}'_m$, where $N =: (S, T, F, M_0)$. Hence, as N is a structural conflict net, ${}^\bullet b \cap {}^\bullet c_m = \emptyset$.

Since $\widehat{P}'_0 \supseteq {}^\bullet b$, by Observation 5(b), and the tokens in the preplaces of b cannot be consumed by the π_P-image of any transition of P that fires after P'_0 has been executed, P can be extended with the transition b, and hence is not maximal. This is the required contradiction. $\square$

7 Unique maximal GR-processes up to $\equiv_1^\infty$

Together, Theorems 1 and 2 say that a structural conflict net has a $\sqsubseteq_1^\infty$-largest BD-process iff it is conflict-free. The "only if" direction stems essentially from [GGS11a], and "if" is contributed here.

Since the preorder $\sqsubseteq_1^\infty$ was not employed in [GGS11a], there we did not consider $\sqsubseteq_1^\infty$-largest BD-processes. Instead, we spoke of a "unique maximal GR-process up to $\equiv_1^\infty$", using the notion of maximality from Definition 6, that is, maximality w.r.t. the prefix order $\leq$ between GR-processes. The following propositions compare $\sqsubseteq_1^\infty$-maximality and $\leq$-maximality.

Proposition 2. Let N be a net and P a process thereof.
If $\llbracket P \rrbracket_\infty$ is $\sqsubseteq_1^\infty$-maximal then some $Q \in \llbracket P \rrbracket_\infty$ is maximal.

Proof. Assume $\llbracket P \rrbracket_\infty$ is $\sqsubseteq_1^\infty$-maximal. By Lemma 5 there exists some maximal Q with $P \leq Q$. By Definition 11, $P \leq Q$ implies $P \sqsubseteq_1^\infty Q$. Since $\llbracket P \rrbracket_\infty$ is $\sqsubseteq_1^\infty$-maximal we have $Q \equiv_1^\infty P$ and Q is a maximal process within $\llbracket P \rrbracket_\infty$. $\square$

The reverse of Proposition 2 does not hold. The first process depicted in Fig. 7 cannot be extended, for none of the tokens in place 2 will in the end come to rest. So it is maximal. Yet, it is not $\sqsubseteq_1^\infty$-maximal. For it is swapping equivalent

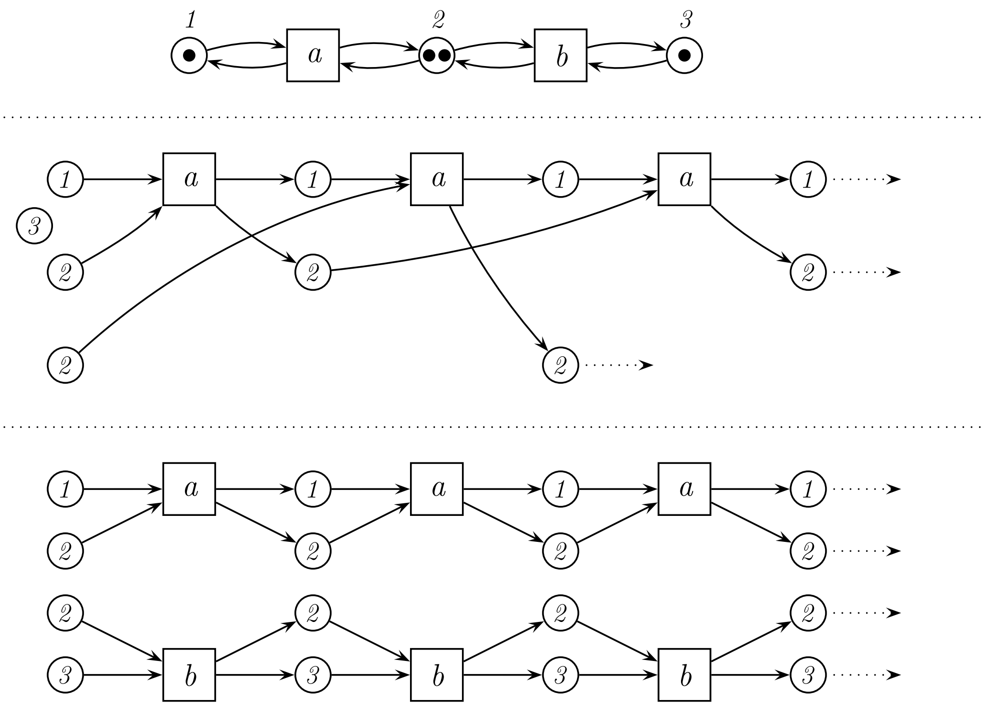

Fig. 7. A net and two maximal GR-processes thereof.

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.

Proposition 3. Let N be a net and P a process thereof.
If P is the only maximal process up to $\equiv_1^\infty$, then it is the $\sqsubseteq_1^\infty$-largest process.

Proof. Let P be the only maximal process of N up to $\equiv_1^\infty$, and Q any other process of N. Let Q' be a maximal process with $Q \leq Q'$—it exists by Lemma 5. Using Definition 11, $Q \leq Q'$ trivially implies $Q \sqsubseteq_1^\infty Q'$. Since P is the only maximal process up to $\equiv_1^\infty$, we have $Q' \equiv_1^\infty P$. Thus $Q \sqsubseteq_1^\infty P$, showing that P is the $\sqsubseteq_1^\infty$-largest process of N. □

[GGS11a, Corollary 1] says that if a structural conflict net N has only one maximal GR-process up to $\equiv_1^\infty$ then N is conflict-free. Using Proposition 3 this is a weakening of Theorem 1. We now establish the converse, that a conflict-free structural conflict net has only one maximal GR-process up to $\equiv_1^\infty$; this is a strengthening of Theorem 2.

Theorem 3. Let N be a conflict-free structural conflict net.
Then N has a unique maximal GR-process up to $\equiv_1^\infty$.

Proof. Let P and Q be two maximal GR-processes of N. The proof of Theorem 2 shows that $[\![P]\!]_\infty$ is the $\sqsubseteq_1^\infty$-largest BD-process of N, and the same holds for Q. So $Q \sqsubseteq_1^\infty P$ and $P \sqsubseteq_1^\infty Q$, i.e., $Q \equiv_1^\infty P$. □

Thus we obtained, for structural conflict nets N, that N is conflict-free iff N has a $\sqsubseteq_1^\infty$-largest BD-process, iff N has a unique maximal GR-process up to $\equiv_1^\infty$. In our technical report [GGS11b] we moreover show that for structural conflict nets the converse of Proposition 2 holds: if $P \in \mathrm{GR}(N)$ is maximal, then $[\![P]\!]_\infty$ is $\sqsubseteq_1^\infty$-maximal—see Lemma 12. Consequently, for structural conflict nets also the converse of Proposition 3 holds. So for structural conflict nets there is no difference between a $\sqsubseteq_1^\infty$-largest BD-process and a unique maximal GR-process up to $\equiv_1^\infty$.

8 Conclusion

We defined a *BD-process* as an equivalence class of Goltz-Reisig processes under the swapping equivalence proposed by Best and Devillers, and argued that on the subclass of structural conflict nets BD-processes constitute a fully satisfactory concept of abstract process of a Petri net under the collective token interpretation. To justify that assessment we showed that a structural conflict net is conflict-free iff it has a largest BD-process.

In the technical report belonging to [GGS11b] we strengthen the result obtained here by showing that each countable net without binary conflict (even if not a structural conflict net) has a largest BD-process. However, proving this is much more complicated than the results presented here.

We leave as an open question to consider also branching time semantics. The notion of a process for condition/event systems was adapted to a branching time semantics of nets through the concept of an *unfolding* of Nielsen, Plotkin and Winskel [NPW81]. Unfolding a net results in an occurrence net with forward branched places that captures all runs of the net, together with the branching structure of choices between them. This work was adapted by Engelfriet in [Eng91] to P/T systems without arc weights, and Meseguer, Sassone and Montanari extended this to cover arc weights as well [MMS97]. The resulting occurrence nets have one branch for every maximal GR-process of the underlying net. It is an open question whether such a construction can be adapted to the collective token interpretation of Petri nets, so that an unfolding of a net has one branch for every BD-process, and thus remains unbranched in case of conflict-free nets.

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.

[DMM89] P. Degano, J. Meseguer & U. Montanari (1989): *Axiomatizing Net Computations and Processes*. In: Proceedings LICS'89, 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, doi:10.1007/BF01463946.

[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 (extended abstract)*. In J.-P. Katoen & B. König, editors: Proceedings CONCUR'11, LNCS 6901, Springer, pp. 43–59, doi:10.1007/978-3-642-23217-6_4. Full version available as Technical Report 2011-06, TU-Braunschweig. Available at http://theory.stanford.edu/~rvg/abstracts.html#90.

[GGS13] R.J. van Glabbeek, U. Goltz & J.-W. Schicke-Uffmann (2013): *On Characterising Distributability*. *Logical Methods in Computer Science* 9(3):17, doi:10.2168/LMCS-9(3:17)2013. Available at http://arxiv.org/abs/1309.3883.

[Gla05] R.J. van Glabbeek (2005): *The Individual and Collective Token Interpretations of Petri Nets*. In M. Abadi & L. de Alfaro, editors: Proceedings CONCUR'05, 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, doi:10.1007/978-3-642-72822-8_15.

[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, doi:10.1016/S0019-9958(83)80040-0.

[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, doi:10.1007/3-540-10001-6_39.

[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, doi:10.1006/inco.1995.1032.

[KM69] R.M. Karp & R.E. Miller (1969): *Parallel Program Schemata*. *Journal of Computer and System Sciences* 3(2), pp. 147–195, doi:10.1016/S0022-0000(69)80011-5.

[LR78] L.H. Landweber & E.L. Robertson (1978): *Properties of Conflict-Free and Persistent Petri Nets*. *Journal of the ACM* 25(3), pp. 352–364, doi:10.1145/322077.322079.

[Maz89] A.W. Mazurkiewicz (1989): *Concurrency, Modularity, and Synchronization*. In: Proceedings MFCS'89*: Mathematical Foundations of Computer Science 1989*, LNCS 379, Springer, pp. 577–598, doi:10.1007/3-540-51486-4_103.

[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 LICS'88, IEEE, pp. 155–164, doi:10.1109/LICS.1988.5114.

[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, doi:10.1017/S0960129597002314.

[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.

[Rei82] Wolfgang Reisig (1982): *Deterministic Buffer Synchronization of Sequential Processes. Acta Informatica* 18, pp. 115–134, doi:10.1007/BF00264434.

[Rei85] W. Reisig (1985): *Petri Nets: An Introduction. EATCS Monographs on Theoretical Computer Science* 4, Springer, doi:10.1007/978-3-642-69968-9.

[Vog90] W. Vogler (1990): *Representation of a Swapping Class by One Net.* In: *Advances in Petri Nets 1989*, LNCS 424, Springer, pp. 467–486, doi:10.1007/3-540-52494-0_42.

[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.