On the Step Branching Time Closure of Free-Choice Petri Nets*

Stephan Mennicke, Jens-Wolfhard Schicke-Uffmann, and Ursula Goltz

`mennicke@ips.cs.tu-bs.de`   `drahflow@gmx.de`   `goltz@ips.cs.tu-bs.de`

TU Braunschweig, Germany

Abstract. Free-choice Petri nets constitute a non-trivial subclass of Petri nets, excelling in simplicity as well as in analyzability. Extensions of free-choice nets have been investigated and shown to be translatable back to interleaving-equivalent free-choice nets. In this paper, we investigate extensions of free-choice Petri nets up to step branching time equivalences. For extended free-choice nets, we achieve a generalization of the equivalence result by showing that an existing construction respects weak step bisimulation equivalence. The known translation for behavioral free-choice does not respect step branching time equivalences, which turns out to be a property inherent to all transformation functions from this net class into (extended) free-choice Petri nets. By analyzing the critical structures, we find two subsets of behavioral free-choice nets that are step branching time equivalent to free-choice nets. Finally, we provide a discussion concerning the actual closure of free-choice Petri nets up to step branching time equivalences.

1   Introduction

A well-known abstraction from distributed systems is the model of Petri nets, arranging the aspects of causally dependent and concurrent actions in a neat graphical representation. Petri nets stem from the ideas given in the PhD thesis of C. A. Petri [12]. Petri forced system states to be structured by means of local states, meeting the concepts of distributed systems, where components are distributed over several locations. That actions of a single component depend on local state information only is naturally captured by the firing rule of Petri nets. In addition, Petri nets offer an intuitive notion of concurrent computation by the *step firing rule*, a generalization of the firing rule allowing more than one transition to fire in parallel (or unordered). Hence, system analysis of distributed systems specified by Petri nets may actually be aware of uncertainties in the order in which actions occur. Although decidable, many interesting properties such as *liveness* or *boundedness* of Petri nets are hard in terms of computational complexity [5].

The Petri net class of *free-choice Petri nets* (FC-nets) is non-trivial w. r. t. expressive power, while efficient analysis algorithms are provided [4, 5]. The core
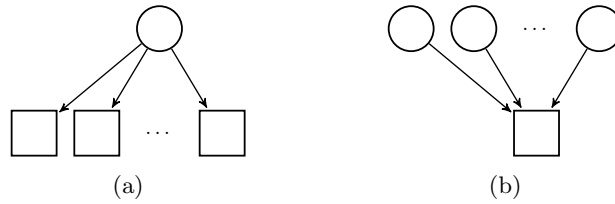

Fig. 1: The two net principles allowed in FC-nets.

idea of *free-choice* is that if the system sometimes has a choice between multiple actions, it always has this choice. It is never the case that only some of these actions are available. Choices between actions may be resolved independently of the system state. The simplest implementation in terms of a Petri net class is provided by FC-nets, comprising only those nets in which conflicting transitions have one condition at most. The two basic subnets following from this restriction are depicted in Fig. 1.

The fundamental results leading to efficient analysis algorithms of FC-nets are achieved in the field of *structure theory* [1], where semantic properties shall follow from syntactical ones. For instance, the *Commoner/Hack Criterion* for FC-nets implies liveness, reducing the liveness problem to a semi-structural property, the basis for efficient analysis. For a comprehensive overview on structure theory, we refer to Best and Wimmel [3]. Esparza provides a survey on computational complexity of analysis algorithms for (free-choice) Petri nets [5].

E. Best states that most of the fundamental results from structure theory do not persist, when using a more expressive net class than FC-nets [1]. However, there are Petri net classes exhibiting the basic concept of free-choice to some degree, provoking the desire to relate them to the original FC-nets. *Extended free-choice nets* exhibit free-choice in that transitions sharing one precondition, share all preconditions. Once again, the set of transitions to choose from does not depend on the system state. In contrast to the aforementioned classes, free-choice may also be described by behavioral notions. A Petri net is a *behavioral free-choice net* if in any system state (reachable from the initial state), two conflicting transitions are either enabled or disabled. Hence, whenever the system has the choice of some of these transitions, it can choose between all of them, again independent of the system state. Best and Shields prove all these net classes equivalent by showing transformations from behavioral free-choice nets to extended free-choice nets and from extended free-choice nets to FC-nets to respect a form of interleaving simulation equivalence [2]. However, it is unclear whether the results from structure theory transfer from FC-nets to free-choice related net classes when considering step semantics of Petri net behavior. Such a transfer would help greatly in applying analysis algorithms for FC-nets to distributed systems where step semantics are relevant.

We analyze the class of FC-nets up to step branching time equivalences. The *Linear Time – Branching Time Spectrum* by van Glabbeek provides a lattice of equivalence notions for linear time behavior such as traces, and branching time

behavior such as bisimulation [6]. We aim at covering the branching behavior of a system, i. e., we take the decision-making process into account. *Failure equivalence* is one of the weakest branching time equivalences, serving as a separator equivalence. If a transformation from one net class into another is impossible, we show that any transformation respecting at least failure equivalence must fail. On the contrary, we use bisimulation equivalence to prove that a particular transformation is correct. For Petri nets, there are two different approaches in interpreting the equivalence notions by van Glabbeek, (i) interleaving, i. e., one action at a time, and (ii) steps, i. e., independent actions may occur unordered or even in parallel. As distributed systems operate in an asynchronous fashion, the step interpretation is appropriate here. Transformations implementing a form of *busy waiting protocol* are not tolerated. Thus, we incorporate divergence-sensitiveness in the equivalences.

Extended free-choice nets remain equivalent to FC-nets under step branching time equivalences. We generalize the proof by Best and Shields to concurrency-aware equivalences. Unfortunately, the well-known construction from behavioral free-choice nets to extended free-choice nets does not respect step branching time behavior, in general. This property turns out to be inherent to all transformation functions from behavioral to (extended) FC-nets. We tackle this problem by analyzing two subsets of behavioral free-choice nets, both being step branching time equivalent to FC-nets. Finally, we discuss the step branching time closure of FC-nets by *symmetrically asynchronous Petri nets*, a net class occurring in the literature discussion of the difference between synchronous and asynchronous behavior [7]. We prove that FC-nets and symmetrically asynchronous Petri nets are equivalent up to step branching time, linking the closure of the class of FC-nets to the closure of symmetrically asynchronous Petri nets.

In Sect. 2, we introduce the basic notions for this paper. The relation between FC-nets and extended free-choice nets is established in Sect. 3 whereas Sect. 4 discusses relations to behavioral free choice net classes. Sect. 5 discusses symmetrically asynchronous Petri nets and their relation to FC-nets. By Sect. 6, we conclude the paper.

2 Basic Notions

In this section, we provide basic notions necessary to follow the employed theories and results concerning free-choice Petri nets. The state notion of Petri nets is based on multisets, which are sets possibly containing the same element multiple times.

Definition 1 (Multiset). *Let S, S' be non-empty sets. A multiset M over S ($M \in \mathbb{N}^S$) is a function $M : S \to \mathbb{N}$. For two multisets A and B over S, we define (i) $A + B$ to be the multiset over S with $(A + B)(s) = A(s) + B(s)$ for all $s \in S$, (ii) $A - B$ to be the multiset over S with $(A - B)(s) = \max\{0, A(s) - B(s)\}$ for all $s \in S$, (iii) $(A \cap B)$ likewise via $(A \cap B)(s) = \min\{A(s), B(s)\}$, and (iv) $A \le B$ iff $A(s) \le B(s)$ for all $s \in S$. The operations (i)–(iii) are generalized to multisets A over S and B over S', resulting in a multiset $(A \, op \, B)$ over $S \cup S'$.*

By $\mathbf{0}_S$ we denote the empty multiset over S. Let $S \supseteq A = \{a_1, \ldots, a_n\}$. Then A denotes the multiset with $A(a_i) = 1$ $(i = 1, \ldots, n)$.

2.1 Petri Nets

In this paper, we consider finite labeled place/transition nets, consisting of two disjoint finite sets P and T, the set of places and the set of transitions, a flow relation F, relating places and transitions, an initial marking, i.e., a multiset over the set of places, and a labeling function for transitions.

Definition 2 (Petri Net). *Let* Act *with* $\tau \notin$ Act *be a finite set of actions. A Petri net is a 5-tuple $N = (P, T, F, M_0, l)$ where P and T are disjoint finite sets, $F \subseteq (P \times T) \cup (T \times P)$, M_0 a multiset over P, and $l : T \to$ Act $\cup \{\tau\}$ a function. If l is injective and $l(T) \subseteq$ Act, then N is* plain. *If l is injective except for τ, i.e., $\forall t, t' \in T : t \neq t' \wedge l(t) = l(t') \Rightarrow l(t) = \tau$, then N is* plain-labeled. *For a node $n \in P \cup T$, by $^\bullet n := \{x \in P \cup T \mid (x, n) \in F\}$ we denote the* preset *of n and, analogously, by $n^\bullet := \{y \in P \cup T \mid (n, y) \in F\}$ the* postset *of n. The notions for pre- and postset canonically extend to (multi-)sets of nodes.*

We assume plain Petri nets as input models for the transformations we describe throughout the paper, because τ-transitions are reserved for implementation purposes. Transformed nets shall preserve the original net transitions but may make use of additional invisible ones. Thus, we expect transformed nets to be plain-labeled.

Petri nets offer a graphical notation, where places are drawn as circles, transitions as boxes, and the flow relation is represented by directed edges, called *arcs*. A marking M is graphically represented by drawing $M(p)$ black dots in the center of each place p.

Definition 3 (Firing Rule). *Let $N = (P, T, F, M_0, l)$ be a Petri net, $M, M' \in \mathbb{N}^P$, $G \in \mathbb{N}^T$, G non-empty and finite. G is called a* step *from M to M', denoted $M[G\rangle_N M'$ iff (i) G is* enabled *under M, denoted $M[G\rangle_N$, meaning that $^\bullet G \leq M$, and (ii) $M' = (M - {}^\bullet G) + G^\bullet$.*

If N is clear from the context, we write $M[G\rangle$ instead of $M[G\rangle_N$ (likewise for $M[G\rangle_N M'$). A marking M is *reachable* in $N = (P, T, F, M_0, l)$ if there are markings $M_1, \ldots, M_n$ and a sequence of steps $G_1, \ldots, G_n$ such that $M_{i-1}[G_i\rangle_N M_i$ $(0 < i \leq n)$ and $M = M_n$. The set of all reachable markings together with the step relation $[G\rangle_N$ induces the state graph of a Petri net, called *reachability graph*.

A common semantic requirement on Petri nets is that each place holds at most one token at each reachable marking. Such nets are called *1-safe nets*. For finite Petri nets, this property implies that the reachability graph of a net is finite, thus exhibiting the expressive power of finite automata. For the generality of our results, we use a more expressive class of Petri nets, but maintain an important aspect of safe Petri nets: Two conflicting transitions, i.e., two transitions sharing a place in their preset, may not fire concurrently, i.e., in the same step. This class of nets is called *structural conflict nets* and was first introduced by van Glabeek et al. [9].

Definition 4 (Reachable Markings, 1-safety, Structural Conflict Net).
Let $N = (P, T, F, M_0, l)$ be a Petri net. The set $[N\rangle$ of reachable markings of N is defined as the smallest set containing M_0, closed under $[G\rangle_N$, meaning that if $M \in [N\rangle$ and $M[G\rangle M'$ for any G, then $M' \in [N\rangle$. N is 1-safe iff $M \in [N\rangle$ implies that for all $p \in P : M(p) \leq 1$. The concurrency relation $\smile \subseteq T^2$ is given by $t \smile u \Leftrightarrow \exists M \in [N\rangle : M[\{t\} + \{u\}\rangle_N$. N is a structural conflict net *iff for all $t, u \in T$ with $t \smile u$ we have $^\bullet t \cap {}^\bullet u = \emptyset$.*

For the remainder of this paper, we assume all nets to be structural conflict nets.

2.2 Semantic Equivalences

The comparison between classes of Petri nets may be performed either by structural equivalences or equivalences capturing certain aspects of the behavior of the net classes. We concentrate on behavioral equivalences in this paper. For a comprehensive overview on comparative equivalence notions, see, e.g., [6].

As a basis for the equivalence notions, we introduce three transition relations for visible and invisible steps, as well as a trace relation. To this end, we divide the set of transitions of a Petri net N into a set of *observable transitions* $\mathcal{O}(N) := \{t \in T \mid l(t) \in \mathsf{Act}\}$ and a set of *invisble transitions* $\mathcal{U}(N) := \{t \in T \mid l(t) = \tau\}$.

Definition 5 (Labeled Transition Relation). *Let $N = (P, T, F, M_0, l)$ be a Petri net over* Act. *(i)* $M \xrightarrow{A}_N M'$ *iff* $\exists G \in \mathbb{N}^{\mathcal{O}(N)} : M[G\rangle M' \wedge l(G) = A$, *(ii)* $M \xrightarrow{\tau}_N M'$ *iff* $\exists t \in \mathcal{U}(N) : M[\{t\}\rangle M'$, *and (iii)* $M \stackrel{\sigma}{\Longrightarrow} M'$ *iff* $\sigma = a_1 a_2 \ldots a_n \in \mathsf{Act}^*$ *and*

$$M \xrightarrow{\tau}{}^*_N \xrightarrow{\{a_1\}}_N \xrightarrow{\tau}{}^*_N \xrightarrow{\{a_2\}}_N \xrightarrow{\tau}{}^*_N \ldots \xrightarrow{\tau}{}^*_N \xrightarrow{\{a_n\}}_N \xrightarrow{\tau}{}^*_N M'$$

where $\xrightarrow{\tau}{}^*_N$ *denotes the reflexive and transitive closure of* $\xrightarrow{\tau}_N$.

Again, we drop the subscripted net identifier N when appropriate.

Based on the previously defined transition relation, we are now able to define our step branching time equivalences. The first equivalence is *step failure equivalence* being the weakest step branching time equivalence we are aware of. It comprises the trace behavior together with a set of steps, the net is not capable of after a given trace.

Definition 6 (Step Failure Equivalence). *Let $N = (P, T, F, M_0, l)$ be a Petri net labeled over* Act, $\sigma \in \mathsf{Act}^*$, *and* $X \subseteq \mathbb{N}^{\mathsf{Act}}$. (σ, X) *is a* step failure pair *of N iff* $\exists M : M_0 \stackrel{\sigma}{\Longrightarrow} M \wedge M \xrightarrow{\tau}{\not\rightarrow} \wedge \forall A \in X : M \xrightarrow{A}{\not\rightarrow}$. $\mathcal{F}(N)$ *denotes the set of all step failure pairs of N. Two Petri nets N_1, N_2 are* step failure equivalent, $N_1 \approx_{\mathcal{F}} N_2$ *iff* $\mathcal{F}(N_1) = \mathcal{F}(N_2)$.

Note that we use a weak (abstracting from τs) version here, as the equivalences we employ have to handle visible as well as invisible transitions. Furthermore, step failure equivalence is divergence-sensitive in that any form of divergence caused by τ-transitions results in exclusion of the corresponding trace from the set of failure pairs. This is ensured by the requirement that from the reachable marking M, no τ-transition may be enabled. We will use step failure equivalence

to obtain separation results. Towards the other end of the step branching time spectrum, there is *step bisimilarity*. Again, we employ the weak version enabling us to abstract from τ-transitions.

Definition 7 (Weak Step Bisimilarity). *Let* $N_i = (P_i, T_i, F_i, M_i, l_i)$ $(i = 1, 2)$ *be two Petri nets. A* weak step bisimulation *between* N_1 *and* N_2 *is a relation* $R \subseteq \mathbb{N}^{P_1} \times \mathbb{N}^{P_2}$ *such that*

1. $(M_1, M_2) \in R$ *and*
2. *if* $(M_1', M_2') \in R$, *then*
   (a) $M_1' \xrightarrow{\tau} M_1'' \Rightarrow \exists M_2'' \in \mathbb{N}^{P_2} : M_2' \xrightarrow{\tau}^* M_2'' \wedge (M_1'', M_2'') \in R,$
   (b) $M_1' \xrightarrow{A} M_1'' \Rightarrow \exists M_2'' \in \mathbb{N}^{P_2} : M_2' \xrightarrow{\tau}^* \xrightarrow{A} \xrightarrow{\tau}^* M_2'' \wedge (M_1'', M_2'') \in R,$
   (c) $M_2' \xrightarrow{\tau} M_2'' \Rightarrow \exists M_1'' \in \mathbb{N}^{P_1} : M_1' \xrightarrow{\tau}^* M_1'' \wedge (M_1'', M_2'') \in R,$
   (d) $M_2' \xrightarrow{A} M_2'' \Rightarrow \exists M_1'' \in \mathbb{N}^{P_1} : M_1' \xrightarrow{\tau}^* \xrightarrow{A} \xrightarrow{\tau}^* M_1'' \wedge (M_1'', M_2'') \in R.$

N_1 *and* N_2 *are* weak step bisimilar, $N_1 \approx_{\mathcal{B}} N_2$ *iff there exists a weak step bisimulation between* N_1 *and* N_2.

If two nets are weak step bisimilar, then they are step failure equivalent. The converse does not hold, in general. Hence, step failure equivalence is coarser than weak step bisimulation thus $N_1 \not\approx_{\mathcal{F}} N_2$ implies $N_1 \not\approx_{\mathcal{B}} N_2$. Therefore, we use step failure equivalence for separation results.

3 Free-Choice and Extended Free-Choice

Free-choice Petri nets (FC-nets) are generalizations of *S-nets* and *T-nets*, both representing net classes providing efficient analysis algorithms, e. g., for liveness or boundedness [1]. FC-nets allow choices between transitions, but only if there is at most one place in the preset of the conflicting transitions (cf. Fig. 1 (a)), and synchronization of places, but only in the way that there is at most one transition in the postset of the synchronized places (cf. Fig. 1 (b)). The conflict pattern refers to restrictions imposed on S-nets, while the synchronization pattern follows restrictions as imposed on T-nets. Hence, every time a conflict occurs, it occurs between the same set of transitions and it is not influenced by the rest of the net.

Definition 8 (Free-Choice Petri Net). *A Petri net* $N = (P, T, F, M_0, l)$ *is a* free-choice Petri net *(FC-net) iff*
$$\forall p \in P \; \forall t \in T : (p, t) \in F \Rightarrow p^{\bullet} = \{t\} \vee {}^{\bullet}t = \{p\}.$$

FC-nets come with a natural notion of distributed component, called *free-choice cluster*. Given an FC-net $N = (P, T, F, M_0, l)$, the free-choice cluster of a transition $t \in T$ is the *conflict cluster* of t, i. e., $[t] = ({}^{\bullet}t)^{\bullet}$. The set of all conflict clusters and thus free-choice clusters of N, denoted by $\mathcal{C}(N)$, imposes a partitioning on the set of transitions if N is an FC-net. Thus, a free-choice cluster may be seen as a component of a system N. Note that each component has at most one role, either to resolve a conflict between actions or to synchronize
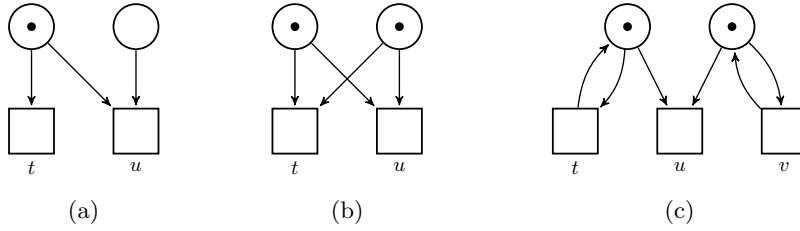
Fig. 2: Three nets that are not FC-nets.

parallel components. Each component $C \in \mathcal{C}(N)$ is equipped with interfaces $^\bullet C$ (the set of input places) and $C^\bullet$ (the set of output places). If some of the places of a component belong to the initial marking, these places are initially marked in the component. Hence, FC-nets are expressive enough to describe non-trivial distributed behavior by a combination of locally restricted choices and synchronization. As already motivated, FC-nets excel in a wide range of applications, mostly due to their structural properties implying semantic properties. For a comprehensive overview, we refer to [1, 3]. Unfortunately, not every Petri net is an FC-net (cf. Fig. 2). However, if a Petri net is behavioral equivalent to an FC-net, analysis results on the transformed net enables us to (partially) reason about semantic properties of the original net.

The Petri net depicted in Fig. 2 (b) belongs to the class of *extended free-choice nets* (EFC-nets). EFC-nets are characterized by the property that if two transitions share any place in their preset, then they share all preplaces. Thus, these nets exhibit the free-choice property to a certain extent, as conflicting transitions always depend on the same set of places.

Definition 9 (Extended Free-Choice Net). *A Petri net* $N = (P, T, F, M_0, l)$ *is an* extended free-choice net *iff* $\forall t_1, t_2 \in T : {}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset \Rightarrow {}^\bullet t_1 = {}^\bullet t_2$.

Furthermore, we observe that the clustering described for FC-net transitions also works in EFC-nets, i.e., $\mathcal{C}(N)$ of an EFC-net N is a partition on the set of transitions of N. Therefore, we call a conflict cluster of an EFC-net *EFC-cluster*. Note that every FC-net is also an EFC-net, but the converse does not hold in general. However, Best and Shields proved that there is a behavioral correspondence by showing that EFC-nets may be transformed into FC-nets, respecting a form of *interleaving simulation* [2].

The core idea of that transformation is to split an EFC-cluster $C \in \mathcal{C}(N)$ into two operational parts, (i) a synchronization of all input places of C and (ii) a choice between all transitions in C. For (i), an unobservable transition is added, consuming one token from each input place of C and producing a token to an additional place p_C. In part (ii), all the transitions in C consume from p_C. An example transformation, namely from the EFC-net in Fig. 2 (b) into the corresponding FC-net is depicted in Fig. 3 (a). Formally, the transformation is defined as follows [2, 1].
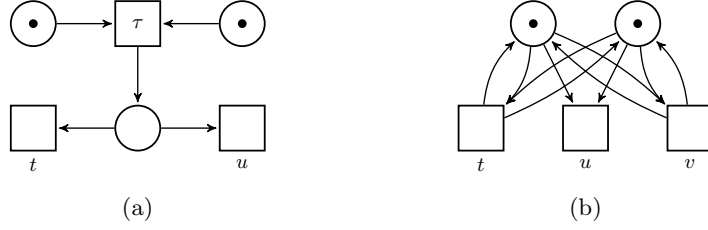
Fig. 3: FC-net and EFC-net Constructions.

Definition 10 ($FC(N)$). *Let $N = (P, T, F, M_0, l)$ be a Petri net. We define $FC(N) := (P', T', F', M_0, l')$ where*

$$
\begin{aligned}
P' &= P \cup \{p_{[u]} \mid [u] \in \mathcal{C}(N)\} \\
T' &= T \cup \{\tau_{[u]} \mid [u] \in \mathcal{C}(N)\} \\
F' &= (F \cap (T \times P)) \cup \{(\tau_{[u]}, p_{[u]}) \mid [u] \in \mathcal{C}(N)\} \cup \\
&\quad \{(p, \tau_{[u]}) \mid [u] \in \mathcal{C}(N), p \in {}^\bullet[u]\} \cup \{(p_{[t]}, t) \mid t \in T\} \\
l' &= l \cup \{(\tau_{[u]}, \tau) \mid [u] \in \mathcal{C}(N)\}.
\end{aligned}
$$

By Best and Shields, $FC(N)$ is an FC-net simulating N if N is an EFC-net [2].

Let us now determine if EFC-nets remain equivalent to FC-nets when considering concurrency-aware equivalences. As we employ structural conflict nets, two transitions may only fire in a step if they do not share any preplace. Furthermore, the transformation defined above preserves conflict clusters in the sense that $\mathcal{C}(N) \subseteq \mathcal{C}(FC(N))$. Hence, if two transitions belonged to a step of N, they are still part of a step of $FC(N)$. Formally, the transformation function FC enjoys the property that an EFC-net N and $FC(N)$ are weak step bisimilar. The weak version of the equivalence is necessary, as FC introduces τ-transitions that do not belong to the original plain EFC-net.

Theorem 1. *Let N be a plain EFC-net. Then $N \approx_{\mathcal{B}} FC(N)$.*

The proof of this theorem relies on the observation that the markings of N are also markings of $FC(N)$. The only difference is that the construction introduces some new places, one per EFC-cluster, that only hold a token if their preceding τ-transition has fired. $R = \{(M_1, M_2) \mid M_1 \in [N\rangle \wedge M_1 \xrightarrow{\tau}{}^*_{FC(N)} M_2\}$ is a weak step bisimulation relation between N and $FC(N)$.

Thus, EFC-nets remain equivalent to FC-nets under concurrency-aware equivalences, even by using the well-known transformation. Intuitively, this paves the way for FC-net-based analysis for distributed systems being specified by EFC-nets. In the following section, we try to repeat this result for *behavioral free-choice nets* (BFC-nets). Unfortunately, not all BFC-nets exhibit (step) behavior that may be expressed by an equivalent FC-net.

4 Behavioral Free-Choice

Free-choice may be explored by a behavioral notion. The net in Fig. 2 (c) is neither an FC-net nor an EFC-net. However, in every reachable marking, con-

flicting transitions t and u as well as u and v are either enabled or disabled. Although not all conflicting transitions depend on exactly the same resources, the initial conflict situation remains invariant. Once again, this meets the intuition of free-choice. This observation is the basis for *behavioral free-choice nets* (BFC-nets).

Definition 11 (Behavioral Free-Choice Net). *A* behavioral free-choice net *(BFC-net) is a Petri net $N = (P, T, F, M_0, l)$ where*
$$\forall t, u \in T : {}^\bullet t \cap {}^\bullet u \neq \emptyset \Rightarrow \forall M \in [N\rangle : {}^\bullet t \leq M \Leftrightarrow {}^\bullet u \leq M.$$

Every EFC-net is a BFC-net and thus every FC-net is also a BFC-net. However, for BFC-nets, the conflict clustering does not provide the property that $\mathcal{C}(N)$ partitions the set of transitions, e. g., the net depicted in Fig. 2 (c) has two overlapping conflict clusters, namely $\{t, u\}$ and $\{u, v\}$. By merging those overlapping clusters, we reach a partitioning of the set of transitions, called BFC-clusters where $\langle u \rangle$ denotes the BFC-cluster of u. Thus, transitions in a BFC-cluster do not necessarily share all of their preplaces, e. g., t and v in Fig. 2 (c) belong to the same BFC-cluster but ${}^\bullet t \cap {}^\bullet v = \emptyset$. However, the BFC-net property ensures that, e. g., in Fig. 2 (c), both t and v are enabled as long as u does not fire. In Fig. 4 (a), another BFC-net is depicted. As before, either all transitions of the BFC-cluster $\langle t \rangle$ are enabled or disabled. Thus, without changing the (step) behavior of the BFC-net, we may make each transition t dependent on all places in ${}^\bullet \langle t \rangle$. In order to preserve the markings of the original net, each additional arc is complemented by an arc in the opposite direction. For the net in Fig. 4 (a) we get the EFC-net depicted in Fig. 4 (b). The same implementation idea applied to the BFC-net in Fig. 2 (c) yields the EFC-net depicted in Fig. 3 (b). For BFC-nets, the result of this construction is always an EFC-net that is interleaving branching time equivalent to the original net [1]. Thus, by performing the FC-net construction described in Sect. 3, we get an FC-net that is at least interleaving branching time equivalent to the original BFC-net.

Definition 12 ($\mathit{EFC}(N)$). *Let $N = (P, T, F, M_0, l)$ be a Petri net. We define $EFC(N) := (P, T, F', M_0, l)$ where*
$$F' := F \cup \{(p, u), (u, p) \mid u \in T, p \in {}^\bullet \langle u \rangle, (p, u) \notin F\}.$$

In Fig. 4, we depicted all intermediate translation steps finally yielding the FC-net depicted in Fig. 4 (c), which is indeed weak step bisimilar to the BFC-net in Fig. 4 (a). Unfortunately, the EFC-net depicted in Fig. 3 (b) is not weak step bisimilar to the net in Fig. 2 (c), as the formerly concurrent transitions t and v are now conflicting. Hence, the proposed construction from the literature does not respect weak step bisimilarity.

In general, there is no transformation from the net in Fig. 2 (c) into a step branching time equivalent (E)FC-net. This is due to the pattern, called *pure* M, the net contains, which has been shown to be stable under step branching time equivalences by van Glabbeek et al. [7]. A pure M is characterized by three distinct transitions, t, u, v with ${}^\bullet t \cap {}^\bullet u \neq \emptyset$, ${}^\bullet u \cap {}^\bullet v \neq \emptyset$, and ${}^\bullet t \cap {}^\bullet v = \emptyset$ such that there is a reachable marking under which all three transitions are enabled. Thus,
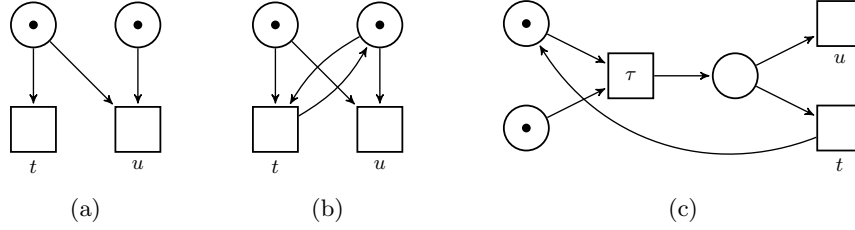
Fig. 4: From BFC-nets to FC-nets.

transitions t and v may fire concurrently while the synchronizing transition u is in conflict with both. In particular, van Glabbeek et al. prove that if a Petri net contains a *pure* M, then every step failure equivalent [8] and thus every *step ready equivalent* [7] Petri net contains a pure M. Hence, every step branching time equivalent Petri net containing the net depicted in Fig. 2 (c) is condemned to contain the same pattern again.

Fortunately, the containment of pure Ms is no necessary condition for a Petri net to be a BFC-net, e. g., the BFC-net in Fig. 4 (a) does not even have three transitions. Therefore, we devise two subclasses of BFC-nets that do not have pure Ms in the following. First, we restrict BFC-nets to those nets that do not have self-loops. Second, we look at the intersection of BFC-nets and so called asymmetric choice nets in which self-loops are allowed. See again [1] for an overview on asymmetric choice nets. Both subclasses turn out to be step branching time equivalent to FC-nets.

4.1 BFC-nets without Self-Loops

In this section, we briefly discuss pure Ms in BFC-nets without self-loops. A self-loop is constituted by a transition t and a place p such that t consumes from and produces to p.

Definition 13 (Self-Loop). *Let $N = (P, T, F, M_0, l)$ be a Petri net. N has a self-loop iff $\exists p \in P : {}^\bullet p \cap p^\bullet \neq \emptyset$.*

If we now consider BFC-nets without any self-loops, we need to reassess our argumentation above on why BFC-nets are not equivalent to FC-nets. Without a self-loop, a pure M is not expressible, because if t or v fired without returning the token to their preplaces they share with u, the overall net would not obey the BFC-net property. Indeed, the construction known from the literature works for BFC-nets without self-loops, proving that BFC-nets without self-loops are step-branching equivalent to (E)FC-nets.

Theorem 2. *Let N be a plain BFC-net without self-loops. Then $EFC(N) \approx_\mathcal{B} N$.*

In order to prove that the construction does not change the step behavior of N, it is sufficient to show that $t \smallsmile u$ implies $\langle t \rangle \neq \langle u \rangle$ for all $t, u \in T$. Suppose, there are transitions contradicting this requirement. If there is a reachable marking
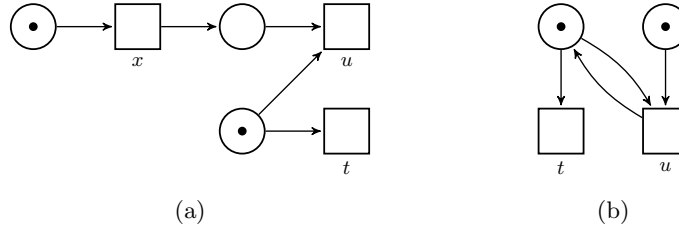
Fig. 5: Two AC-nets having a Partially and Fully Reachable N.

enabling t and u, then after firing t, u is enabled while t is not, contradicting that N is a BFC-net. Therefore, the EFC-net construction does not restrict the set of enabled steps from reachable markings. The remaining proof follows the same principles as the proof of Theorem 3, provided in the appendix. Next, we discuss a subclass that allows self-loops in BFC-nets but restricts the class structurally to not contain Ms.

4.2 BFC-nets and Asymmetric Choice

The condition for a Petri net to be an EFC-net may be reformulated in terms of places, yielding exactly the same net class as by Def. 9 [1]. In this definition, a Petri net is an EFC-net iff for all places p and q it holds that $p^\bullet \cap q^\bullet \neq \emptyset$ implies that $p^\bullet = q^\bullet$. In an *asymmetric choice net* (AC-net), this restriction is weakened to an alternative between the two set-inclusions between $p^\bullet$ and $q^\bullet$. Before discussing the combination of BFC-nets and AC-nets, we clarify the relation between AC-nets and FC-nets.

Definition 14 (Asymmetric Choice Net). *A Petri net $N = (P, T, F, M_0, l)$ is an* asymmetric choice net *(AC-net) iff*
$$\forall p, q \in P : p^\bullet \cap q^\bullet \neq \emptyset \Rightarrow p^\bullet \subseteq q^\bullet \vee q^\bullet \subseteq p^\bullet.$$

In case $p^\bullet \subseteq q^\bullet$ but $q^\bullet \not\subseteq p^\bullet$, all transitions in $q^\bullet$ are enabled if at least p and q hold tokens. This restriction hardly enforces the concepts of free-choice, as it does not require conflicting transitions to be chosen freely. However, the Petri net depicted in Fig. 2 (a) shows an example AC-net, which may be simulated by an FC-net. We discuss this transformation in Sect. 5. Considering the same net in a larger context, it is unclear whether a token eventually occurs on the second input place of u. The AC-net in Fig. 5 (a) shows an example context. The choice between transitions t and u does only occur if transition x fires in advance. In an adversary scenario, transition x might only fire if transition t has fired, which makes u unable to fire at all. In the literature, such a situation is called *(asymmetric) confusion* [13], which may not be simulated by any FC-net, as choices may never occur conditionally. In general, we characterize such conditional choices by the notion of *partially and fully reachable* Ns. Here, the existence of a subnet like the one in Fig. 2 (a) is expected and two distinct markings, one under which t is enabled and one enabling t and u.

Definition 15 (Partially and Fully Reachable N). *Let $N = (P, T, F, M_0, l)$ be a Petri net. $(t, u) \in T^2$ is a partially and fully reachable N iff ${}^\bullet t \cap {}^\bullet u \neq \emptyset$, $|{}^\bullet u| > 1$, and $\exists M_t, M \in [N\rangle : {}^\bullet t \leq M_t \wedge {}^\bullet u \not\leq M_t \wedge {}^\bullet\{t, u\} \leq M$.*

Note that this notion does not only characterize asymmetric confusion patterns, but also nets as the one depicted in Fig. 5 (b). Just as in the case of pure Ms, partially and fully reachable Ns are stable w.r.t. step branching time equivalences.

Proposition 1. *Let N be a plain Petri net with a partially and fully reachable N. Then for all plain-labeled Petri nets N' with $N \approx_{\mathcal{F}} N'$ it holds that N' has a partially and fully reachable N.*

Proof. Let $N = (P, T, F, M_0, l)$ and (t, u) a partially and fully reachable N of N, i.e., there exist reachable markings M_t and M such that M_t enables t, but not u and M enables both. Thus, there is a σ such that $M_0 \overset{\sigma}{\Longrightarrow} M_t$ and since N is plain, and hence deterministic [14], (1) $(\sigma, X) \in \mathcal{F}(N)$ implies that $\{l(t)\} \notin X$ and there exists (σ, X) with $\{l(u)\} \in X$. From the existence of M we deduce that (2) there is a σ' such that $M_0 \overset{\sigma'}{\Longrightarrow} M$ and (3) $(\sigma', X) \in \mathcal{F}(N)$ implies that $\{l(t)\} \notin X \wedge \{l(u)\} \notin X$. As N is a structural conflict net, (4) there exists $(\sigma', X) \in \mathcal{F}(N)$ such that $\{l(t), l(u)\} \in X$.

   Let $N' = (P', T', F', M'_0, l')$ be a Petri net with $N \approx_{\mathcal{F}} N'$, i.e., $\mathcal{F}(N) = \mathcal{F}(N')$. Therefore, $\mathcal{F}(N')$ obeys (1)–(4). Hence, there exist transitions $t', u' \in T'$ with $l'(t') = l(t)$ and $l'(u') = l(u)$. By (3) and (4), we have that ${}^\bullet t' \cap {}^\bullet u' \neq \emptyset$. By (1), we get that there is a place $p \in {}^\bullet u' \setminus {}^\bullet t'$ thus $|{}^\bullet u'| > 1$. By (1), there is a marking $M_{t'}$ with $M'_0 \overset{\sigma}{\Longrightarrow} M_{t'}$ and ${}^\bullet t' \leq M_{t'} \wedge {}^\bullet u' \not\leq M_{t'}$, as otherwise there would be a failure pair $(\sigma, X) \in \mathcal{F}(N')$ with $\{l'(t')\} \in X$. By (2), there is a marking M' with $M'_0 \overset{\sigma'}{\Longrightarrow} M'$ and by (3) ${}^\bullet t' \leq M' \wedge {}^\bullet u' \leq M'$. By (4) $\{t', u'\}$ is no step from M'. Summarizing, N' has a partially and fully reachable N. $\square$

Corollary 1. *There is no transformation function from plain AC-nets to plain-labeled step branching time equivalent FC-nets.*

However, the BFC-net in Fig. 2 (c) does not respect the restrictions as imposed on AC-nets. In general, such structures may not occur in an AC-net and conversely, partially and fully reachable Ns are ruled out in BFC-nets, which paves the way for the following theorem.

Theorem 3. *Let N be a plain BFC-net and an AC-net. Then $EFC(N) \approx_{\mathcal{B}} N$.*

Note that Best and Shields already prove that *EFC* yields an EFC-net for BFC-nets [2]. Concerning the proof of Theorem 3, we benefit from the construction given in Def. 12, adding no more places to the original net. Thus, every marking of N is also a marking of $EFC(N)$. Furthermore, the construction does not change the enabledness condition for transitions, and it does not even increase or decrease formerly existing steps. Thus, the desired weak step bisimulation between N and $EFC(N)$ is $R = \{(M, M) \,|\, M \in [N\rangle\}$. The proof that R is a weak step bisimulation is provided in the appendix.

5 Symmetric Asynchrony

Throughout the last section, we have seen two subclasses of BFC-nets that are step branching time equivalent to FC-nets. However, BFC-nets do not represent a reasonable basis for the step branching time closure of FC-nets, i.e., there are non-BFC-nets being behaviorally equivalent to FC-nets, e.g., the net depicted in Fig. 2 (a). In this section, we analyze a Petri net class that appears in the context of investigations on synchronous and asynchronous interaction. R. van Glabbeek et al. introduced this class that does not distinguish simultaneous and delayed token removals from more than one place on a behavioral level [7]. Each place of a Petri net is assumed to reside on an individual location. If a transition has only one preplace, it may be assigned to the same location. Two transitions reside on a shared location if they share a preplace that is allowed on the same location. Thus, whenever a transition synchronizes more than one place, i.e., more than one location, it resides on a different location than any of its preplaces. Interaction delays between locations are simulated by an additional τ-transition and place on each arc crossing location borders, an *asynchronous implementation*. A Petri net is symmetrically asynchronous iff it is step failure equivalent to its asynchronous implementation.

Definition 16 (Symmetric Asynchrony [7]). *Let $\mathcal{L}$ be a set of* locations *and $N = (P, T, F, M_0, l)$ be a Petri net. A (distribution) function $\lambda : (P \cup T) \to \mathcal{L}$ is a* symmetric distribution on N *when*

$\quad \lambda(p) = \lambda(q) \quad$ *for $p, q \in P$* $\qquad$ *only if $p = q$,*
$\quad \lambda(t) = \lambda(p) \quad$ *for $t \in T, p \in P$* $\quad$ *only if $^\bullet t = \{p\}$ and*
$\quad \lambda(t) = \lambda(u) \quad$ *for $t, u \in T$* $\qquad$ *only if $t = u$ or $\exists p \in P : \lambda(t) = \lambda(p) = \lambda(u)$,*

$\mathcal{I}_\lambda(N) := (P \cup P_\tau, T \cup T_\tau, F', M_0, l)$ *is the* asynchronous implementation (w.r.t. λ) of N *iff*

$$P_\tau := \{s_t \,|\, t \in T, s \in {}^\bullet t, \lambda(s) \neq \lambda(t)\},$$
$$T_\tau := \{t_s \,|\, t \in T, s \in {}^\bullet t, \lambda(s) \neq \lambda(t)\},$$
$$F' := \{(t, s) \,|\, t \in T, s \in t^\bullet\} \cup \{(s, t) \,|\, s \in {}^\bullet t, \lambda(s) = \lambda(t)\} \cup$$
$$\{(s, t_s), (t_s, s_t), (s_t, t) \,|\, t \in T, s \in {}^\bullet t, \lambda(s) \neq \lambda(t)\},$$
$$l'(t) := \begin{cases} l(t) & \text{if } t \in T \\ \tau & \text{otherwise.} \end{cases}$$

N *is* symmetrically asynchronous *iff there exists a symmetric distribution λ on N such that $N \approx_\mathcal{F} \mathcal{I}_\lambda(N)$.*

R. van Glabbeek et al. prove that a Petri net is symmetrically asynchronous iff there is a symmetric distribution λ and the Petri net has no *distributed conflict* w.r.t. λ [7]. A Petri net is said to have a distributed conflict w.r.t. λ iff there are two distinct transitions t, u with $^\bullet t \cap {}^\bullet u \neq \emptyset$, $\lambda(t) \neq \lambda(u)$, and there is a reachable marking enabling t. Another characterization of symmetrically asynchronous Petri nets is given in terms of a Petri net pattern, called *partially reachable* N [7], which follows the definition given in Def. 15, leaving out the existence of the marking M (full reachability). All the nets depicted in Fig. 2 have partially reachable Ns, obstructing symmetric asynchrony.

R. van Glabbeek et al. conjecture that there is a strong relation between symmetrically asynchronous nets and FC-nets [8], which we finally prove in the following. Given a symmetrically asynchronous Petri net $N = (P, T, F, M_0, l)$, i.e., there is a symmetrical distribution $\lambda : P \cup T \to \mathcal{L}$ such that N has no distributed conflict w.r.t. λ. We transform N under λ by $FC_{sym}(N) = (P, \hat{T}, \hat{F}, M_0, l)$ where

- $\hat{T} := \{t \in T \,|\, \forall p \in {}^\bullet t : \forall u \in p^\bullet : \lambda(p) = \lambda(u) \vee u = t\}$ and
- $\hat{F} := F \cap ((P \times \hat{T}) \cup (\hat{T} \times P))$.

This transformation yields an FC-net by removing transitions obstructing the FC-net properties.

Lemma 1. *Let N be a symmetrically asynchronous Petri net. Then, $FC_{sym}(N)$ is an FC-net.*

Proof. Let $N = (P, T, F, M_0, l)$ be a symmetrically asynchronous Petri net and λ the required symmetrical distribution. N has no distributed conflict w.r.t. λ [7], i.e., $\forall t, u \in T \; \forall p \in {}^\bullet t \cap {}^\bullet u : t = u \vee \lambda(p) = \lambda(u) \vee \nexists M \in [N\rangle : {}^\bullet t \leq M$. Let $\hat{N} = FC_{sym}(N)$ with set of transitions $\hat{T}$ and arc relation $\hat{F}$. We show that for all $p \in P$ and all $t \in \hat{T}$, $(p, t) \in \hat{F}$ implies ${}^\bullet t = \{p\}$ or $p^\bullet = \{t\}$. Assume there is a place p and a transition t with $(p, t) \in \hat{F}$, but ${}^\bullet t \neq \{p\}$ and $p^\bullet \neq \{t\}$. Hence, there exists a place $q \neq p$ with $q \in {}^\bullet t$ and a transition $u \in T$ with $t \neq u$ and $u \in p^\bullet$. As λ is symmetrical, p is not co-located with q and t, q is not co-located with u and t, and hence, t is not co-located with u. There are two cases to consider, (1) $\lambda(p) = \lambda(u)$ and (2) $\lambda(p) \neq \lambda(u)$. If (1), then $u \notin \hat{T}$, because by definition of $\hat{T}$, for all transitions in $p^\bullet$, it must hold that they are co-located with p or they are equal to u. But $u \neq t$, $t \in p^\bullet$ and $\lambda(p) \neq \lambda(t)$. If (2), then $u, t \notin \hat{T}$ by definition of $\hat{T}$. In detail, $t \notin \hat{T}$, as there is a transition $u \in p^\bullet$ with $\lambda(p) \neq \lambda(u)$ (by assumption (2)). $u \notin \hat{T}$, as there is a transition, $t \in p^\bullet$, with $\lambda(t) \neq \lambda(p)$ (by λ being symmetrical). Hence, either $\{t\} = p^\bullet$ or t does not even exist. Thus, $FC_{sym}(N)$ is an FC-net. $\qquad\square$

Fortunately, none of the removed transitions takes part in actual behavior, as all of them are dead transitions, i.e., there is no reachable marking enabling them.

Lemma 2. *Let $N = (P, T, F, M_0, l)$ be a symmetrically asynchronous Petri net and $FC_{sym}(N) = (P, \hat{T}, \hat{F}, M_0, l)$. Then for all $t \in T \setminus \hat{T}$ there is no $M \in [N\rangle$ such that $M[\{t\}\rangle$.*

Proof. Assume there is a $t \in T \setminus \hat{T}$ and a marking $M \in [N\rangle$ such that t is enabled under M. By definition of $\hat{T}$, there is a $p \in {}^\bullet t$ and a $u \in p^\bullet$ such that $\lambda(p) \neq \lambda(u)$ and $u \neq t$. But M enabled t, contradicting the assumption that N is symmetrically asynchronous, i.e., N has no distributed conflict. $\qquad\square$

Hence, FC_{sym} yields step branching time equivalent FC-nets.

Theorem 4. *Let N be a plain symmetrically asynchronous Petri net. Then $N \approx_{\mathcal{B}} FC_{sym}(N)$.*

This theorem directly follows from Lemma 2. As every FC-net is also symmetrically asynchronous, we get the following correspondence between the step branching time closure of FC-nets and of symmetrically asynchronous nets.

Corollary 2. *Let N be a plain Petri net. N is weak step bisimilar to a plain-labeled FC-net iff N is weak step bisimilar to a plain-labeled symmetrically asynchronous Petri net.*

6 Conclusions and Future Work

We provided an overview on well-known free-choice related Petri net classes and their transformations between them. Our goal was to analyze their relations under concurrency-aware equivalences, namely step branching time equivalences. EFC-nets are equivalent to FC-nets while BFC-nets contain critical structures not expressible by any FC-net. We proved that two subclasses of BFC-nets, namely self-loop free BFC-nets and BFC-nets that are also AC-nets, are behaviorally equivalent to FC-nets respecting concurrent steps. Finally, we proved that the closure of step branching time equivalent FC-nets relies on the closure of symmetrically asynchronous Petri nets, a syntactically larger class than FC-nets.

FC-nets come with efficient analysis algorithms that are also useful for the analysis of distributed systems. Positive results in this paper may be used to transport many of the analysis techniques of FC-nets to broader net classes like EFC-nets or BFC-nets. Negative results we pointed to or actually proved, sketch a limit of efficient analyzability, as Best stated that many of the free-choice related results do not hold for net classes beyond free-choice [1].

For future work, we plan to tackle the closure of symmetrically asynchronous Petri nets, i.e., FC-nets (cf. Corollary 2). In [11], partial closure and remaining problematic structures are provided by equating symmetrically asynchronous Petri nets with distributed systems and asking for the class of distributable Petri nets. For another notion of distributed Petri nets related to the aforementioned pure M, such a closure is provided by van Glabbeek et al. in terms of *ST bisimulation*, a strong equivalence taking causality into account [10]. The EFC-net construction presented in this paper adds causality. Thus, a causality-preserving transformation remains to be found if possible.

References

1. Best, E.: Structure theory of petri nets: the free choice hiatus. In: Brauer, W., Reisig, W., Rozenberg, G. (eds.) Petri Nets: Central Models and Their Properties, LNCS, vol. 254, pp. 168–205. Springer (1987)
2. Best, E., Shields, M.W.: Some equivalence results for free choice nets and simple nets and on the periodicity of live free choice nets. In: Ausiello, G., Protasi, M. (eds.) CAAP'83, LNCS, vol. 159, pp. 141–154. Springer (1983)
3. Best, E., Wimmel, H.: Structure theory of petri nets. In: Jensen, K., Aalst, W.M.P., Balbo, G., Koutny, M., Wolf, K. (eds.) Transactions on Petri Nets and Other Models of Concurrency VII, LNCS, vol. 7480, pp. 162–224. Springer (2013)

4. Desel, J., Esparza, J.: Free Choice Petri Nets. Cambridge University Press, New York, NY, USA (1995)
5. Esparza, J.: Decidability and complexity of petri net problems – an introduction. In: Reisig, W., Rozenberg, G. (eds.) Lectures on Petri Nets I: Basic Models, LNCS, vol. 1491, pp. 374–428. Springer (1998)
6. van Glabbeek, R.J.: The linear time - branching time spectrum. In: Baeten, J., Klop, J. (eds.) CONCUR '90 Theories of Concurrency: Unification and Extension, LNCS, vol. 458, pp. 278–297. Springer (1990)
7. van Glabbeek, R.J., Goltz, U., Schicke, J.W.: On synchronous and asynchronous interaction in distributed systems. In: Ochmaski, E., Tyszkiewicz, J. (eds.) MFCS 2008, LNCS, vol. 5162, pp. 16–35. Springer (2008)
8. van Glabbeek, R.J., Goltz, U., Schicke, J.W.: Symmetric and asymmetric asynchronous interaction. In: ICE 2008, Satellite Workshop ICALP 2008. entcs, vol. Vol. 229, pp. 77–95. elsevier (2009)
9. van Glabbeek, R.J., Goltz, U., Schicke, J.W.: Abstract processes of place/transition systems. Information Processing Letters 111(13), 626 – 633 (2011)
10. van Glabbeek, R.J., Goltz, U., Schicke-Uffmann, J.W.: On characterising distributability. LMCS 9(3) (2013)
11. Mennicke, S.: Strong Distributability Criteria for Petri Nets. Master's thesis, TU Braunschweig, Germany (may 2013)
12. Petri, C.A.: Kommunikation mit Automaten. Ph.D. thesis, TU Darmstadt (1962)
13. Rozenberg, G., Thiagarajan, P.: Petri nets: Basic notions, structure, behaviour. In: Bakker, J., Roever, W.P., Rozenberg, G. (eds.) Current Trends in Concurrency, LNCS, vol. 224, pp. 585–668. Springer (1986)
14. Vidal-Naquet, G.: Deterministic languages of petri nets. In: Girault, C., Reisig, W. (eds.) Application and Theory of Petri Nets, Informatik-Fachberichte, vol. 52, pp. 198–202. Springer Berlin Heidelberg (1982)

A Proof of Theorem 1

Let $N = (P, T, F, M_0, l)$ be a plain EFC-net and $N' = FC(N) = (P', T', F', M_0', l')$. We prove that $R = \{(M_1, M_2) \mid M_1 \in [N\rangle, M_1 \xrightarrow{\tau}{}_{N'}^{*} M_2\}$ is a bisimulation. Note that $P \subseteq P'$ and $T \subseteq T'$. $(M_0, M_0') \in R$, as $M_0 = M_0'$. Let $(M_1, M_2) \in R$. As N is τ-free, there are three cases to consider:

1. $M_1 \xrightarrow{A}_N M_1'$, i.e., $M_1[G\rangle_N M_1'$ and $l(G) = A$. Note that ${}^{\bullet}t \cap {}^{\bullet}t' = \emptyset$ for all $t, t' \in G$ ($t \neq t'$), due to N being a structural conflict net. Existence of marking M_2' with $M_2 \xrightarrow{\tau}{}_{N'}^{*} \xrightarrow{A}{}_{N'}^{*} \xrightarrow{\tau}{}_{N'}^{*} M_2'$ and $(M_1', M_2') \in R$ is shown in two cases, (i) $M_2 = M_1$ and (ii) $M_2 \neq M_1$.

   In case (i), $\forall t \in G : \forall p \in {}^{\bullet}[t] : M_1(p) > 0$ and thus $M_2(p) > 0$. By construction, there are transitions $\tau_{[t]}$ enabled in N' for each $t \in G$, i.e., $M_2[\{\tau_{[t]} \mid t \in G\}\rangle_{N'}$ Hence, $M_2 \xrightarrow{\tau}{}_{N'}^{*} M_2''$ and $M_2''[G\rangle_{N'}$. Therefore, $M_2 \xrightarrow{\tau}{}_{N'}^{*} \xrightarrow{A}_{N'} M_2'$. Furthermore, $M_2' = M_1'$, as all outgoing arcs of transitions are preserved. By definition, $(M_1', M_2') \in R$. In case (ii), $M_1 \xrightarrow{\tau}{}_{N'}^{*} M_2$. Hence, for each $t \in G$ either $M_2[\{t\}\rangle_{N'}$ or $M_2[\tau_{[t]}\rangle_{N'}$. Let $G' = \{\tau_{[t]} \mid t \in G \wedge M_2[\tau_{[t]}\rangle_{N'}\}$. G' is a step from M_2 to M_2'' and by construction, $M_2''[G\rangle_{N'}$. Hence, $M_2'' \xrightarrow{A}_{N'} M_2'$

and $M_2 \xrightarrow{\tau}{}^*_{N'} \xrightarrow{A}_{N'} M_2'$. If $M_2' = M_1'$, $(M_1', M_2') \in R$ is implied. If not, then $M_1' \xrightarrow{\tau}{}^*_{N'} M_2'$, as otherwise, G was no step from M_2'' to M_2'. Thus, $(M_1', M_2') \in R$.

2. $M_2 \xrightarrow{\tau}_{N'} M_2'$. Then $M_1 \xrightarrow{\tau}{}^*_N M_1$ and $(M_1, M_2') \in R$.

3. $M_2 \xrightarrow{A}_{N'} M_2'$, $M_2[G\rangle_{N'} M_2'$ and $l(G) = A$. We need to show that $M_1 \xrightarrow{A}_N M_1'$ and $(M_1', M_2') \in R$. We construct M_1 by reversing τ-transitions in N'. As $(M_1, M_2) \in R$, $M_1 \xrightarrow{\tau}{}^*_{N'} M_2$. By construction, G is enabled under M_2 iff G is enabled under M_1. Hence, there is a marking M_1' such that $M_1 \xrightarrow{A}_N M_1'$. If $M_2' = M_1'$, then the claim is implied. Otherwise, M_2' evolves from M_1' by firing the τ firing sequences we rolled back earlier. By definition, $(M_1', M_2') \in R$.

Hence, R is a weak step bisimulation between N and $FC(N)$. $\qquad\square$

B Proof of Theorem 3

Let $N = (P, T, F, M_0, l)$ be a BFC-AC-net specification and $EFC(N) = N' = (P, T, F', M_0, \hat{l})$. We prove that $R = \{(M, M) \mid M \in [N\rangle\}$ is a weak step bisimulation between N and N'. By definition, $(M_0, M_0) \in R$. Let $(M_1, M_2) \in R$. The cases for τ-steps are obsolete, as N and N' are plain.

1. $M_1 \xrightarrow{A}_N M_1'$, i.e., there is a step G from M_1 to M_1' and $l(G) = A$. As N' is τ-free, we have to give a marking M_2' such that $M_2 \xrightarrow{A}_{N'} M_2'$ and $(M_1', M_2') \in R$. If G is enabled by M_2, then $M_2' = M_1'$ where $M_2[G\rangle M_2'$, as $M_2 = M_1$ and tokens that are consumed due to the construction of N' are reproduced. Additionally, the original outgoing arcs of transitions are preserved. As N is a structural conflict net and due to the construction, for transitions $t_1, t_2 \in G$ it holds that ${}^\bullet t_1 \cap {}^\bullet t_2 = \emptyset$. It remains to be shown that $\hat{G} = G$ is enabled by M_2, i.e., ${}^\bullet\hat{G} \le M_2$. Let $\hat{t} \in \hat{G}$ with $\hat{t} = t$. The $\hat{t}/\hat{G}$ notation just helps distinguishing between elements of N and N'. As in N', no arcs were removed, it holds that $|{}^\bullet\hat{t}| \ge |{}^\bullet t|$. In case of equality, the claim holds and G may fire under M_2. If $|{}^\bullet\hat{t}| > |{}^\bullet t|$, then there are places $p, q \in P$ and a transition $u \in T$ such that $p \in {}^\bullet t \cap {}^\bullet u$, $q \notin {}^\bullet t$ but $q \in {}^\bullet u$ — due to the fact that N is an AC-net. Hence, in N' there is an additional arc from q to $\hat{t}$. As N is a BFC-net, t is enabled iff u is enabled and hence, $q \in M_1$. Thus, $q \in M_2$. In conclusion ${}^\bullet\hat{t} \le M_2$ thus ${}^\bullet\hat{G} \le M_2$, i.e., $\hat{G}$ is enabled by M_2 and may fire, producing a new marking M_2' with the properties described above.

2. $M_2 \xrightarrow{A}_{N'} M_2'$, i.e., there is a step G from M_2 to M_2' and $l(G) = A$. N is τ-free and therefore, we need to give a marking M_1' such that $M_1 \xrightarrow{A}_N M_1'$ and $(M_1', M_2') \in R$. As before, a t in N has less or equal incoming arcs than t in N'. As, $M_2 = M_1$ by definition of R, G is a step from M_1 to a marking M_1'. Every token that is consumed by some t in N', due to the construction of N', is reproduced in N'. In N such tokens are not even consumed by t. The original postsets of transitions are preserved and hence, $M_1[G\rangle_N M_1'$ and $M_2' = M_1'$ thus $(M_1', M_2') \in R$. $\qquad\square$