Deciding (Sub-Marking) Reachability in $O(P^2 + T^2)$ for Sound Acyclic Free-Choice Workflow Nets

Thomas M. Prinz[1], Christopher T. Schwanen[2], and Wil M. P. van der Aalst[2]

[1] Course Evaluation Service, Friedrich Schiller University Jena, Jena, Germany
Thomas.Prinz@uni-jena.de
[2] Chair of Process and Data Science (PADS), RWTH Aachen University, Aachen, Germany
{schwanen,wvdaalst}@pads.rwth-aachen.de

Abstract. Reachability is a central decision problem in Petri net theory deciding whether a given marking can be reached from the initial marking. Sub-marking reachability (the covering problem) asks whether there is a reachable marking, which consists of at least the tokens in the given marking. The current state of the art describes the computational complexity of both problems as polynomial for live and bounded free-choice nets as well as for sound free-choice workflow nets. This paper refines this complexity on the class of sound acyclic (simple) free-choice workflow nets to $O(P^2 + T^2)$. The presented approach uses three new concepts: admissibility, maximum admissibility, and diverging transitions. *Admissibility* requires that all places in a given marking are pairwise concurrent. *Maximum admissibility* states that adding a place to an admissible marking would make it inadmissible. A *diverging transition* is a transition which originally "produces" the concurrent tokens that lead to a given marking. All three concepts can additionally provide explanations why a (sub-)marking is not reachable.

Keywords: Workflow Nets · Reachability · Covering problem · Soundness · Free-choice

1 Introduction

A central problem in Petri net theory is *reachability*: deciding if a given marking can be reached from the initial marking. Reachability of a given marking is crucial to show whether a given system, which shall be analyzed, fulfills a given property, or not. Such properties can be, for example, the absence of deadlocks, livelocks, or undesired states during conformance checking. As it forms the basis of many verification approaches, deciding reachability and deciding it computationally effective is important. The general reachability problem for Petri nets falls within the `NONELEMENTARY` complexity class [7]. Cheng et al. [4] gives a `PSPACE` complexity class for general safe nets. Esparza [11] stated that the reachability problem in safe and live free-choice nets can be reduced to the *CNF-SAT* problem; thus, deciding reachability for this class is NP-complete. However, Desel and Esparza [8] already showed earlier that reachability in cyclic free-choice nets, a restricted subclass of live and bounded free-choice nets where the initial marking is a home marking, can be decided in polynomial time. Regarding

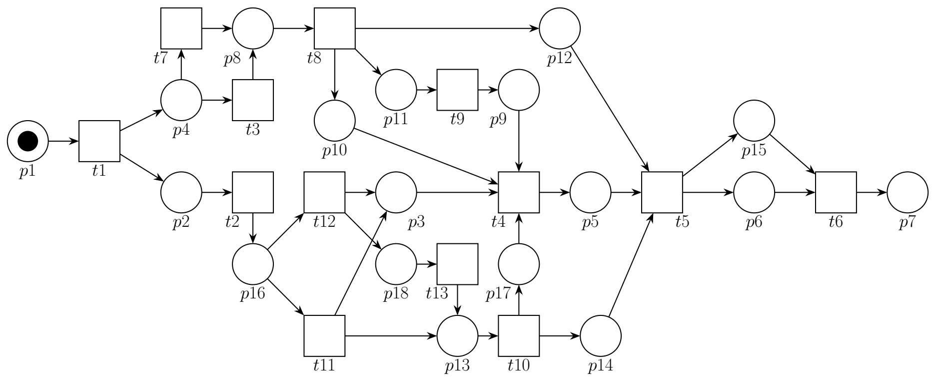


Fig. 1. A sound acyclic free-choice workflow net.

sound extended free-choice workflow nets, Yamaguchi [22] confirms a polynomial time complexity of deciding reachability, but the concrete polynomial is unknown.

In this paper, we show that reachability for *sound acyclic (simple) free-choice workflow nets* can be solved in quadratic time, $O(P^2 + T^2)$. We further show that *sub-marking reachability* [12] (the *covering problem*, i. e., if a partial marking is reachable) can be solved in the same computational complexity.

Figure 1 shows an example of a sound acyclic free-choice workflow net. A question could be: Is a marking reachable in which $p9$, $p12$, and $p16$ have tokens? To answer such questions, instead of focusing on *concrete occurrence sequences* (traces) to the desired marking, the presented algorithm demonstrates that most decisions on whether a marking is reachable can be decided by *concurrency*. Thereby, we introduce the concept of *admissible* and *maximum admissible* markings. A marking M is *admissible* if all pairs of marked places are in a concurrency relation. M is further *maximum admissible* if it is not possible to add a further token to the marking without destroying admissibility. We show that each reachable marking in a sound acyclic free-choice workflow net must be maximum admissible and that neither computing the concurrency relation [14, 19] nor checking maximality requires knowledge of concrete occurrence sequences.

Although maximum admissibility is a necessary condition and good heuristics of deciding reachability, it is unfortunately not sufficient. There are nets with markings that are maximum admissible but *not* reachable. For this reason, we argue that concurrency is always introduced by diverging transitions. This manifests in occurrence nets of sound acyclic free-choice workflow nets as subgraphs which only diverge in transitions. Hence, the identification of these diverging transitions implies the existence of an occurrence net that leads to the (maximum) admissible marking. For this reason, deciding reachability can be achieved by checking the given marking for maximum admissibility and determining whether a diverging transition exists that leads to the marking — examining concrete occurrence sequences is not necessary. Similarly, a sub-marking is reachable if the marking is just admissible and there is such a diverging transition. The overall approach has a quadratic computational complexity in the worst case. As a further benefit, the approach can also explain why a marking is *not* reachable rather than just deciding reachability.

Of course, this paper examines a quite limited class of Petri nets regarding (sub-marking) reachability. There are two main reasons why investigating this class of nets is still important: (1) Industrial business process models strongly correlate with free-choice workflow nets [13]. Soundness is seen as an important quality criterion [10] of process models and can be checked in cubical computational time complexity with detailed diagnostic information [18]. Finally, many industrial process models are acyclic. For this reason, for an industrial setting, sound acyclic free-choice workflow nets are an interesting class of nets. (2) There is an interesting trend investigating *free-choice nets with a home cluster* [3], which showed that this class of nets correlates strongly with *perpetual nets* finally correlating with sound free-choice workflow nets. Mapping the new method of *loop decomposition* for industrial workflow graphs [17–19] to free-choice workflow nets should eventually fill the gap to investigate the class of free-choice nets with a home cluster with the class of sound acyclic free-choice workflow nets. For this reason, this paper can be interpreted as a further step to achieve a low computational polynomial time complexity to decide (sub-marking) reachability while providing diagnostic information at the same moment.

The remainder of this paper is structured as follows: Section 2 introduces basic concepts of Petri nets, markings, reachability, paths, and soundness. Maximum admissible markings and their application are discussed in Section 3 which are then used in Section 4 to finally decide reachability in sound acyclic free-choice workflow nets. Eventually, Section 5 concludes this paper.

2 Preliminaries

This paper uses standard notions for Petri nets, which are provided in the following. We also recall the *Path-to-End Theorem* for simple free-choice nets.

2.1 Multisets, Petri Nets, and Paths

$\mathscr{B}(A)$ is the set of all *multisets* over some set A. For a multiset $b \in \mathscr{B}(A)$, $b(a)$ denotes the number of times element $a \in A$ appears in b. For example, $b_1 = [\,] = \emptyset$, $b_2 = [x,x,y]$, $b_3 = [x,y,z]$, $b_4 = [x,x,y,x,y,z]$, and $b_5 = [x^3,y^2,z]$ are multisets over the set $A = \{x,y,z\}$. b_1 is the empty multiset, b_2 and b_3 consist of three elements, and $b_4 = b_5$, i. e., the ordering of elements is irrelevant and b_5 uses a more compact notation for repeating elements. The standard set operators can be extended to multisets, e. g., $x \in b_2$, $b_2 \uplus b_3 = b_4$, $b_5 \setminus b_2 = b_3$, etc.

Definition 1 (Petri Nets). *A* Petri net *(or simply a* net*) N is a triple (P,T,F) with P and T are finite, disjoint sets of* places *and* transitions*, and $F \subseteq (P \times T) \cup (T \times P)$ is the* flow *relation.* ⌟

$P \cup T$ can be interpreted as *nodes* and F as *edges* between those nodes. For $x \in P \cup T$, $\bullet x := \{p \mid (p,x) \in F\}$ is the *preset* of x (all directly preceding nodes) and $x \bullet := \{s \mid (x,s) \in F\}$ is the *postset* of x (all directly succeeding nodes). Each node in $\bullet x$ is an *input* of x and each node in $x\bullet$ is an *output* of x. The preset and postset of a set of nodes $X \subseteq P \cup T$ is defined as $\bullet X := \bigcup_{x \in X} \bullet x$ and $X\bullet := \bigcup_{x \in X} x\bullet$, respectively. N is *proper* iff $\forall t \in T\colon \bullet t \neq \varnothing \wedge t\bullet \neq \varnothing$. N is *(extended) free-choice* iff $\forall t_1, t_2 \in T\colon \bullet t_1 \cap \bullet t_2 \neq \varnothing \implies \bullet t_1 = \bullet t_2$. N is *simple* free-choice iff $\forall p \in P\colon |p\bullet| \geqslant 2 \implies \bullet(p\bullet) = \{p\}$, i.e., $\forall t_1, t_2 \in T\colon \bullet t_1 \cap \bullet t_2 \neq \varnothing \implies \{p\} = \bullet t_1 = \bullet t_2$ [9].

Without loss of generality, this paper focuses on simple *free-choice nets as Murata [15] presents a linear time transformation algorithm of extended to simple and simple to extended free-choice nets.*

A *path* $(n_1, \ldots, n_m)$ is a sequence of nodes $n_1, \ldots, n_m \in P \cup T$ with $m \geqslant 1$ and $\forall i \in \{1, \ldots, m-1\}\colon\ n_i \in \bullet n_{i+1}$. Note that places and transitions alternate on paths. $[(n_1, \ldots, n_m)]$ depicts the set of all nodes on the path. If all nodes of a path are pairwise different, the path is *acyclic*; otherwise, it is *cyclic*. $\mathit{Paths}(x, y)$ denotes the set of all paths between nodes x and y, where $x, y \in P \cup T$. N is *acyclic* if all its paths are acyclic. In the nets shown here, circles represent places, rectangles transitions, and directed edges represent flows as done in Fig. 1.

Definition 2 (Workflow Nets, FC-WF-Nets, and AFC-WF-Nets). *A* workflow net *$N = (P, T, F, i, o)$ is a net (P, T, F) with $i, o \in P$, $\bullet i = o\bullet = \varnothing$. i is the* source *and o is the* sink *of N. All nodes are on a path from i to o. If N is (simple) free-choice, then N is called a* FC-WF-net. *If N is acyclic (simple) free-choice, N is called an* AFC-WF-net. ⌟

This paper focuses on AFC-WF-nets.

2.2 Markings, Reachability, Properties, and Soundness

The behavior of nets is defined via *markings*, which describe the number of *tokens* on places in a specific state.

Definition 3 (Marking). *A* marking *M of a net $N = (P, T, F)$ is a multiset of places, $M \in \mathscr{B}(P)$. (N, M) is a* marked net. *$\langle M \rangle := \{x \in M\}$ depicts the set of places of M.* ⌟

Transitions whose input places all have tokens are *enabled* in a marking and can be fired, leading to the net's semantics:

Definition 4 (Enabledness, Firing, and Reachability). *Let (N, M) be a marked net $N = (P, T, F)$. A transition $t \in T$ is* enabled *in M, denoted as $(N, M)[t\rangle$, iff every place $\bullet t$ contains at least one token in M, $\bullet t \subseteq M$. $en(N, M) := \{t \in T \mid (N, M)[t\rangle\}$ is the set of enabled transitions in (N, M).*

If t is enabled in M, then t may fire*, which removes one token from each of t's input places and adds one token to each of t's output places. $M' = (M \setminus \bullet t) \uplus t\bullet$ is the marking resulting from firing t in (N, M). $(N, M)[t\rangle(N, M')$ denotes that t is enabled in (N, M) and firing t would result in (N, M').*

A sequence $\sigma = \langle t_1, t_2, \ldots, t_n \rangle \in T^*$, $n \geqslant 1$, is an occurrence sequence *if there are markings $M_1, M_2, \ldots, M_{n+1}$ such that $\forall 1 \leqslant i \leqslant n\colon (N, M_i)[t_i\rangle(N, M_{i+1})$. For applying such a sequence σ, we write $(N, M_1)[\sigma\rangle(N, M_{n+1})$. We say that $t_1, t_2, \ldots, t_n$* occur *in σ.*

A marking M' is reachable *from M if there is an occurrence sequence σ such that $(N, M)[\sigma\rangle(N, M')$. $R(N, M) := \{M' \in \mathscr{B}(P) \mid \exists \sigma \in T^*\colon (N, M)[\sigma\rangle(N, M')\}$ denotes the set of all reachable markings of (N, M).*

If $N = (P, T, F, i, o)$ *is a workflow net,* $[i]$ *is its* initial marking *and* $[o]$ *is its* final marking. ⌟

Next, we list some behavioral properties important for analysis:

Definition 5 (Live, Bounded, Safe, and Dead). *Let* (N, M) *be a marked net* $N = (P, T, F)$. (N, M) *is* live *iff for every reachable marking* $M' \in R(N, M)$ *and for every transition* $t \in T$*, there is a reachable marking* $M'' \in R(N, M')$*, which enables* t. (N, M) *is* k-bounded *iff for every reachable marking* $M' \in R(N, M)$ *and for every place* $p \in P$: $M'(p) \leqslant k$. (N, M) *is* bounded *iff there is a* k *such that* (N, M) *is* k*-bounded.* (N, M) *is* safe *iff* (N, M) *is 1-bounded.*

A place $p \in P$ *is* dead *in* (N, M) *iff* $\forall M' \in R(N, M): M'(p) = 0$. *A transition* $t \in T$ *is* dead *in* (N, M) *iff* $\forall M' \in R(N, M): t \notin en(N, M')$. ⌟

Workflow nets can be *sound* [1]:

Definition 6 (Soundness). *A workflow net* $N = (P, T, F, i, o)$ *with its initial marking* $[i]$ *and its final marking* $[o]$ *is* sound *iff*

(1) $\forall M \in R(N, [i]): [o] \in R(N, M)$,
(2) $\forall M \in R(N, [i]): (o \in M \implies M = [o])$, *and*
(3) there is no dead *transition in N:* $\forall t \in T\ \exists M, M' \in R(N, [i]): (N, M)[t\rangle(N, M')$.

An equivalent definition of soundness is that the marked short-circuited net $(N', [i])$ *of* N*, where* N' *is defined as*

$$(P, T \cup \{t\}, F \cup \{(o, t), (t, i)\})$$

for a transition $t \notin T$*, is live and bounded [1].* ⌟

Sound free-choice workflow nets are further *safe*:

Lemma 1 (Safeness). *Sound free-choice workflow nets are safe.* ⌟

Proof (Lemma 1). See Verbeek et al. [21]. □

Sound simple free-choice workflow nets ensure that each path from an arbitrary place to the sink place o contains at most one token [19]:

Theorem 1 (Path-to-End Theorem). *Let* $W = (P, T, F, i, o)$ *be a simple FC-WF-net. It holds: On all paths* ρ *from any place* p *to* o *in any reachable marking* $M \in R(W, [i])$*, the sum of all tokens on all places of* ρ *is at most 1, i. e.,*

$$\forall p \in P\ \ \forall \rho \in Paths(p, o)\ \ \forall M \in R(W, [i]): |[\rho] \cap M| \leqslant 1.$$ ⌟

Proof (Theorem 1). See Theorem 2 in Prinz et al. [19].

Note that this is a special case of Lemma 5.11 in Van der Aalst [2]. □

3 (Maximum) Admissible Markings

In the following, we describe, argue, and prove why reachability and *concurrency* strongly interact in sound AFC-WF-nets. We will show that a given marking can only be reachable if all its places are pairwise concurrent. Two places are concurrent if there is a reachable marking with tokens on both places:

Definition 7 (Concurrency). *Let $N = (P, T, F, i, o)$ be a sound AFC-WF-net. Two places $x, y \in P$ are* concurrent *in N, denoted $x \parallel y$, iff $\exists M_\parallel \in R(N, [i])\colon [x, y] \subseteq M_\parallel$. $\parallel(x) := \{y \in P \mid x \parallel y\}$ denotes the set of all places to which x is concurrent. Due to safeness of sound AFC-WF-nets, a place x is not concurrent to itself.* ⌟

For example, in Fig. 1, $p9 \parallel p10$ and $p5 \parallel p12$ as well as $\parallel(p15) = \{p6\}$.

Sound AFC-WF-nets have benefits regarding their complexity during analysis. One of them is that two concurrent places must not have a path between them:

Lemma 2. Let $N = (P, T, F, i, o)$ be a sound AFC-WF-net with two places $x, y \in P$, $x \neq y$.

$$x \parallel y \quad \Longrightarrow \quad \mathit{Paths}(x, y) = \varnothing \,\wedge\, \mathit{Paths}(y, x) = \varnothing$$ ⌟

Proof (Lemma 2). See Prinz et al. [19] (Cor. 4). The interested reader can also use the Path-to-End Theorem 1 to confirm the path absence of concurrent nodes. □

The interested reader can check that, e. g., $p9$ in Fig. 1 has no path to $p10$ and vice versa.

By Lemma 1, sound AFC-WF-nets are safe. For this reason, all concurrent places x and y, $x \parallel y$, must be joined by a transition on all pairs of paths from x and y to o:

Lemma 3 (Two Concurrent Places are Joined by Transitions). *Let $N = (P, T, F, i, o)$ be a sound AFC-WF-net. Furthermore, let $x, y \in P$. It holds:*

$$\begin{gathered} x \parallel y \\ \Longrightarrow \\ \forall \rho_x = (x_1, \ldots, x_n, o) \in \mathit{Paths}(x, o), n \geqslant 1 \\ \forall \rho_y = (y_1, \ldots, y_m, o) \in \mathit{Paths}(y, o), m \geqslant 1 \\ \exists i \in \{1, \ldots, n\}\ \exists j \in \{1, \ldots, m\}\colon \\ \{x_1, \ldots, x_i\} \cap \{y_1, \ldots, y_j\} = \{x_i\} = \{y_j\} \subseteq T \end{gathered}$$ ⌟

Proof (Lemma 3). Constructive proof. Let $x, y \in P$ with $x \parallel y$. By Definition 7 of concurrency and from $x \parallel y$, it follows that $\exists M \in R(N, [i])\colon [x, y] \subseteq M$. By Lemma 1, N is safe and, thus, $x \neq y$.

By Definition 2 of workflow nets, there are at least two paths $\rho_x = (x_1, \ldots, x_n, o) \in \mathit{Paths}(x, o)$ and $\rho_y = (y_1, \ldots, y_m, o) \in \mathit{Paths}(y, o)$ from $x = x_1$ and $y = y_1$ to the sink o, respectively. It follows also: $[\rho_x] \cap [\rho_y] \neq \varnothing$. Therefore, ρ_x and ρ_y must have a *first* common node $x_i = y_j$ with $x_i \in [\rho_x] \cap [\rho_y]$ and $\{x_1, \ldots, x_{i-1}\} \cap \{y_1, \ldots, y_{j-1}\} = \varnothing$ and $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$.

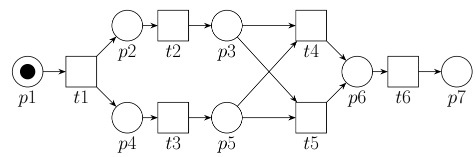


Fig. 2. In sound extended free-choice workflow nets, Lemma 3 does not hold without modifications as $p6$ is *not* a joining transition.

According to the proof of the Path-to-End Theorem 1 in Prinz et al. [19], p. 136, Eq. (4), it holds for each transition on a path ρ to the sink o in a sound (simple) AFC-WF-net because of simple free-choiceness:

$$\forall t \in ([\rho] \cap T): \quad |\bullet t \cap [\rho]| = 1 \quad \wedge \quad |t \bullet \cap [\rho]| \geqslant 1$$

For this reason, removing a token from ρ_x or ρ_y starting from M, respectively, can only be achieved by a place p with $|p\bullet| \geqslant 2$ (a decision). Since N is simple free-choice, $\bullet(p\bullet) = \{p\}$, any output transition of such p can be fired in each reachable marking $M' \in R(N,M)$, $p \in M'$ — thus, also output transitions on ρ_x ($p\bullet \cap [\rho_x]$) and ρ_y ($p\bullet \cap [\rho_y]$). As a consequence, we can treat the tokens on both paths ρ_x and ρ_y starting from x and y, respectively, to "remain" on ρ_x and ρ_y, i. e., if a transition t_x on ρ_x fires, it puts a token back on ρ_x, and if a transition t_y on ρ_y fires, it puts a token back on ρ_y. Since N is sound, this treatment of tokens to "remain" on ρ_x and ρ_y cannot lead to a dead transition. Furthermore, since $x \parallel y$ and Lemma 2, there is no path from x to y and from y to x, i. e., no token can get from x to y and vice versa.

Now, there are two possibilities for the first common node $x_i = y_j$:

$x_i \in P$: Without loss of generality, once the token of ρ_x "reaches" x_i before the token of ρ_y in a reachable marking $M' \in R(N,M)$, then in M' are at least two tokens on path ρ_y to o. This contradicts the Path-to-End Theorem 1. ↯ Therefore, $x_i \notin P$.

$x_i \in T$: This is the only remaining possibility. ✓

Both cases state that all pairs of paths ρ_x and ρ_y have a transition as a first common node. For this reason, this lemma holds. □

Fig. 2 shows an example of a sound *extended* free-choice workflow net, for which Lemma 3 does not hold. However, since this paper focuses on *simple* free-choiceness, such cases are not possible as $t4$ and $t5$ would be merged into a single transition.

The concurrency relation is symmetric by definition. The relationship is crucial for checking reachability since all places in a reachable marking must be in a concurrency relation by Definition 7. We call a marking where all places are pairwise concurrent *admissible*:

Definition 8 (Admissible Markings). *Let N be a sound AFC-WF-net. A marking $M_a \in \mathscr{B}(P)$ is* admissible *if all different places in M_a are pairwise concurrent:*

$$\forall x, y \in P, x \neq y: \quad [x,y] \subseteq M_s \implies x \parallel y$$ ⌟

Each marking containing a single place is admissible. Following from Definition 8, admissibility of a marking is a necessary condition for its (sub-marking) reachability. If a marking is *not* admissible, we can quickly decline its reachability. In addition, admissibility of markings limits the "size" of reachable markings by the concurrency relation. There must be *maximum* admissible markings:

Definition 9 (Maximum Admissible Markings). *Let $N = (P, T, F, i, o)$ be a sound AFC-WF-net. A marking $M \in \mathscr{B}(P)$ is* maximum admissible *iff*

1. $\forall x, y \in M, x \neq y \colon x \parallel y$ *(M is admissible) and*
2. $\bigcap_{x \in M} \parallel(x) = \varnothing$ *(adding places of $P \backslash \langle M \rangle$ to M would make M inadmissible).* ⌟

A marking M composed of concurrent places can only be *not* reachable by Definition 8 for three reasons: (1) M contains not enough places, (2) M contains too much places, or (3) M has a "correct" number of places but all places cannot be concurrent at the same time. Fortunately, we can show that each *reachable* marking in a sound AFC-WF-net is *maximum admissible*, i. e., removes reasons (1) and (2):

Theorem 2 (Reachable Markings are Maximum Admissible). *Let N be a sound AFC-WF-net. All reachable markings M_r from the initial marking $[i]$ are maximum admissible.* ⌟

Proof (Theorem 2). Let $N = (P, T, F, i, o)$ be a sound AFC-WF-net. Furthermore, let $M_r = [p_1, \dots, p_m]$, $m \geqslant 1$, be a reachable marking $M_r \in R(N, [i])$. The proof is done by contradiction: M_r is *not* maximum admissible. As M_r is admissible by Definition 8 but M_r is not *maximum* admissible, by Definition 9, there must be a place $p \in (P \backslash M_r)$ not in M_r being concurrent to all places in M_r:

$$\exists p \in (P \backslash M_r)\ \forall p' \in M_r \colon p' \parallel p \tag{1}$$

Let p be such a place in the following.

For each $p_i \in M_r$, $1 \leqslant i \leqslant m$, there is a path $\rho_i \in \mathit{Paths}(p_i, o)$ to the sink o by Def. 2 of workflow nets. In addition, let $\rho_p \in \mathit{Paths}(p, o)$ be a path to o, which contains p by Def. 2. By Theorem 1, for each ρ_i there is exact one token ρ_i on p_i in M_r (ρ_i is safe). There are exactly two cases for ρ_p:

Case 1: $\exists p' \in \big([\rho_p] \cap P\big) \colon p' \in M_r$ (there is a token on ρ_p in M_r). Thus, there is a path from p to p', $\mathit{Paths}(p, p') \neq \varnothing$. Since $p \parallel p'$ by (1), this contradicts Lemma 2 that concurrency requires the absence of paths. ↯ This case does not hold.

Case 2: $\forall p' \in \big([\rho_p] \cap P\big) \colon p' \notin M_r$ (there is no token on ρ_p in M_r). The entire situation is abstractly illustrated in Fig. 3. By Lemma 3, all such two paths ρ_x and ρ_y, $\rho_x, \rho_y \in \{\rho_1, \dots, \rho_m, \rho_p\}$, $\rho_x \neq \rho_y$, contain a transition t with at least two input places $in_x \in ([\rho_x] \cap \bullet t)$ and $in_y \in ([\rho_y] \cap \bullet t)$, $in_x \neq in_y$; even for ρ_p with any other path ρ_i, $1 \leqslant i \leqslant m$. For the moment, we say ρ_x and ρ_y "converge" in t. By this case, ρ_p is without any token in M_r, i. e., the transition(s) $t_1, \dots, t_m$, in which ρ_p converges with any of the other paths ρ_i, $1 \leqslant i \leqslant m$, are dead when the token(s) on those paths reach any of $t_1, \dots, t_k$ in a marking $M' \in R(N, M_r)$. This contradicts Definition 6 of soundness. ↯ This case does not hold.

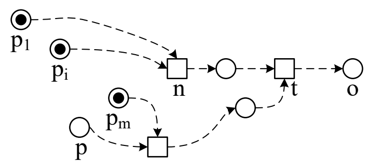


Fig. 3. For each p_i with a token in M_r, there is a path ρ_i to the sink o.

Since both cases do not hold, the contradiction does not hold. For this reason, the theorem holds that each reachable marking is maximum admissible. □

For example, the marking $[p9, p10]$ in Fig. 1 is *not* maximum admissible since $p9$ and $p10$ have $p3$, $p17$, etc. as common concurrent places. The marking $[p3, p5]$ is *not* maximum admissible because it is not admissible at all: $p3 \nparallel p5$. However, the marking $[p5, p12, p14]$ is maximum admissible as all places, to which $p12$ is concurrent (e. g., $p9$, $p11$, $p18$, etc.), are not concurrent to $p5$ and $p14$.

It follows for checking if a marking is reachable in a sound AFC-WF-net to investigate first its maximum admissibility. By Definition 9, checking maximum admissibility of a marking depends on two simple rules. Following these two rules, the computation whether a given marking is (maximum) admissible or not, can be decided in $O(|P|^2 + |T|^2)$:

Theorem 3 (Computational Complexity of (Maximum) Admissibility). *Let $N = (P, T, F, i, o)$ be a sound AFC-WF-net. Deciding, whether a given marking M_r is maximum admissible in N or not, can be achieved in $O(|P|^2 + |T|^2)$.* ⌟

Proof (Theorem 3). It is possible with the algorithm of Prinz et al. [19] to determine the concurrency relation by Definition 7 in $O(|P|^2 + |T|^2)$ for sound AFC-WF-nets.

The second rule of Definition 9 states:

$$\bigcap_{x \in M_r} \|(x) = \varnothing. \tag{2}$$

Since $x \in M_r$ is concurrent to all other places of M_r except to itself, x is only not in the intersection of all concurrency sets because x is missing in its own concurrency set $\|(x)$. For this reason, temporarily adding x to its own concurrency set leads to a combination of both rules:

$$\bigcap_{x \in M_r} \big(\|(x) \cup \{x\} \big) = M_r. \tag{3}$$

Once the concurrency relation is computed and the concurrency sets of all places are stored within a computationally efficient data structure for mathematical sets being able to compute the intersection in constant time, e. g., by a BitSet in Java, Eq. (3) can be checked in linear time, $O(|P|)$.

In summary, the overall computational complexity is dominated by the computation of the concurrency relation and can, therefore, be achieved in $O(|P|^2 + |T|^2)$. Furthermore, if the concurrency relation is stored, checking (maximum) admissibility can be achieved in linear time, $O(|P|)$. □

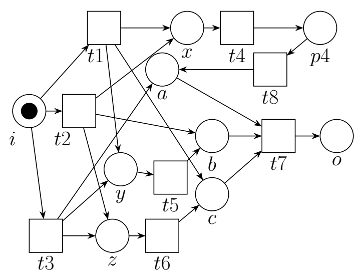


Fig. 4. A net with a maximal marking $[x,y,z]$ that is not reachable.

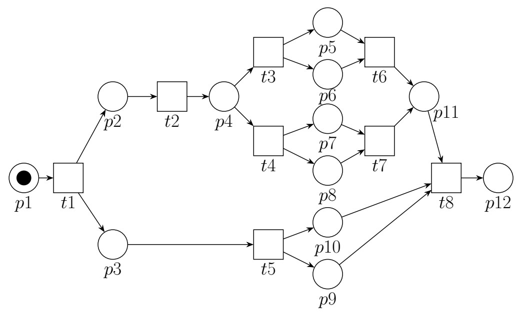


Fig. 5. A sound AFC-WF-net.

Unfortunately, maximum admissibility is only a necessary condition for reachability but *not* sufficient since not each maximum admissible marking is also reachable as the net in Fig. 4 demonstrates. This net contains three places x, y, and z. They are pairwise concurrent, i. e., $x \parallel y$, $x \parallel z$, and $y \parallel z$. The resulting marking $[x,y,z]$ is maximum admissible. However, as the reader can confirm, this marking is not reachable since the places are not concurrent to each other at the same time. As a consequence, not every maximum admissible marking is reachable. We have to confirm maximum admissible markings, which are reachable as well.

4 Reachability

This section will show the sufficient condition of reachability in sound AFC-WF-nets. Besides the absence of paths between concurrent places, another benefit of sound AFC-WF-nets is that their *occurrence nets* [16] (i. e., a net, which represents how a corresponding net was executed) are simplified to *run nets* [20]. In contrast to occurrence nets, run nets are just subgraphs (sub *nets*) of sound AFC-WF-nets and, therefore, each node can only occur once. A run net is defined as follows [20]:

Definition 10 (Run Nets). *Let $N = (P,T,F,i,o)$ be a sound AFC-WF-net. An occurrence sequence $\sigma \in T^*$ is a* run *iff σ leads from the initial marking $[i]$ to the final marking $[o]$. A net $\pi = (P_\sigma, T_\sigma, F_\sigma)$ is a* run net *of a run σ of N iff*

$$P_\sigma = \Big\{ p \in P \;\big|\; p \in \bigcup_{t \in \sigma} (\bullet t \cup t \bullet) \Big\}$$
$$T_\sigma = \{ t \in T \mid t \in \sigma \}$$
$$F_\sigma = F \cap \big((P_\sigma \cup T_\sigma) \times (P_\sigma \cup T_\sigma) \big).$$

$n \in \sigma$ occurs *in π, depicted as $n \in \pi$. $\Pi(N)$ depicts the set of all run nets of N.* ⌟

If a run net $\pi = (P_\sigma, T_\sigma, F_\sigma)$ of a sound AFC-WF-net $N = (P,T,F,i,o)$ contains a transition $t \in T_\sigma$, then it must also contain its input and output places, $\forall t \in T_\sigma : (\bullet t \cup t \bullet) \subseteq P_\sigma$.

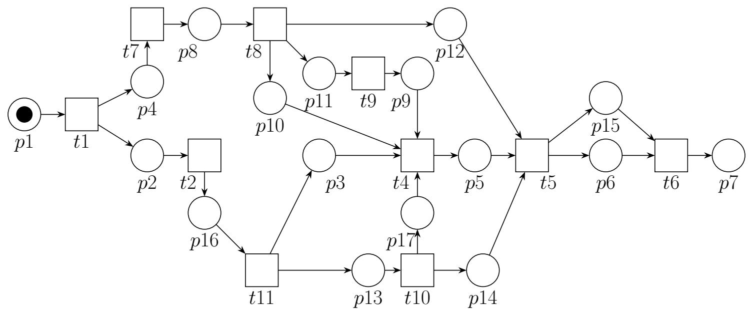


Fig. 6. Example of a run net of the example workflow net in Fig. 1.

If a run net contains a place $p \in P_\sigma$, $p \notin \{i, o\}$, it must also contain exactly one of its input and one of its output transitions, $\forall p \in (P_\sigma \backslash \{i, o\}) \colon |p\bullet \cap T_\sigma| = 1$ and $|\bullet p \cap T_\sigma| = 1$. Figure 6 shows an example of a run net of our net in Fig. 1.

Let M_m be a marking for which we want to decide reachability in a given sound AFC-WF-net N. M_m must be maximum admissible by Theorem 2 and, thus, all places of M_m are pairwise concurrent. Regarding the run nets of N and Lemma 2 about the absence of paths between concurrent places, there cannot be paths between any possibly concurrent places in any run net of N. This fact was also confirmed in Theorem 2 in Prinz et al. [20]. For this reason, deciding whether a maximum admissible or admissible sub-marking M_m is reachable in N corresponds with the existence of a run net that contains all places of M_m:

Corollary 1. *Let $N = (P, T, F, i, o)$ be a sound AFC-WF-net and a marking M_m. It holds:*

$$M_m \in R(N, [i]) \iff M_m \text{ is maximum admissible } \wedge \exists (P_\sigma, T_\sigma, F_\sigma) \in \Pi(N) \colon \langle M_m \rangle \subseteq P_\sigma \tag{4}$$

$$\exists M_s \in R(N, [i]) \colon M_m \subseteq M_s \iff M_m \text{ is admissible } \wedge \exists (P_\sigma, T_\sigma, F_\sigma) \in \Pi(N) \colon \langle M_m \rangle \subseteq P_\sigma \tag{5}$$

Proof (Corollary 1). Constructive proof by focusing on Eq. (4):

$\Longrightarrow$ $M_m \in R(N, [i])$. Since M_m is reachable from $[i]$, there must be a run net by Definition 10 of run nets, which contains all places of M_m. Furthermore, by Theorem 2, each reachable marking is maximum admissible, and, therefore, M_m. ✓

$\Longleftarrow$ M_m is maximum admissible $\wedge$ $\exists (P_\sigma, T_\sigma, F_\sigma) \in \Pi(N) \colon \langle M_m \rangle \subseteq P_\sigma$. Since M_m is maximum admissible, all places of M_m are pairwise concurrent. Furthermore, there is a run net π, which contains all places of M_m. For this reason, and by Lemma 2, all places of M_m do not have paths to each other in N as well as in π. Thus, all places in M_m can have tokens in the same reachable marking from $[i]$ as they can appear in any order from each other. As a consequence, M_m is a reachable marking from $[i]$. ✓

The proof of Eq. (5) can be done similarly. □

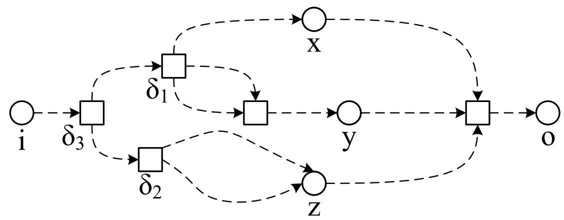


Fig. 7. For the marking $[x,y,z]$ there are diverging points δ_1 (for x and y) and δ_3 (for x and z as well as y and z). The source i is *no* diverging point of $[x,y,z]$ as all paths to them pass the same output transition of i. δ_2 is no diverging point of $[x,y,z]$ as it has only paths to z.

Since we are able to decide the (maximum) admissibility of a marking in $O(|P|^2 + |T|^2)$, we need an approach to check if there is at least one run net, which contains all places of a given marking. Deriving such a run net is not trivial as decisions (places with multiple outgoing flows) can lead to an exponential growth of possible run nets. However, we do not have to derive a concrete run net—we only have to show that such a run net must exist, or not. For this reason, we will focus in the following on transitions, which diverge the control-flow and, therefore, "produce" concurrency.

If a sound AFC-WF-net has a reachable marking M_r with at least two places x and y, then there must be at least one transition t with multiple output places $\{x_t, y_t, \ldots\}$ (as the net has just one source) and x_t and y_t have disjoint paths to x and y so that the tokens are not joint before x and y. We call such t a *diverging point* of $\{x,y\}$ as well as a *diverging point* of the set $M_r = D$. Fig. 7 illustrates the concept of diverging points abstractly for three places x, y, and z. δ_1 is a diverging point of $\{x,y,z\}$ although it only has disjoint paths to x and y. δ_2 is no diverging point of $\{x,y,z\}$ since it only has paths to z. The transition δ_3 is a diverging point of $\{x,y,z\}$ as it has disjoint paths to x and z as well as y and z. The source i is *no* diverging point of $\{x,y,z\}$ since each path from i to them goes through δ_3. The following definition describes that formally:

Definition 11 (Diverging Points). *Let $N = (P,T,F,i,o)$ be an AFC-WF-net. A node $\delta \in P \cup T$ is a* diverging point *of a set $D \subseteq P \cup T$, $|D| \geqslant 2$, of nodes, depicted $\delta \lll D$, if for at least two nodes $d_1, d_2 \in D$, $d_1 \neq d_2$, and for at least two output nodes $o_1, o_2 \in \delta\bullet$, there are two disjoint paths ρ_1 from o_1 to d_1 and ρ_2 from o_2 to d_2.*

$$\exists d_1, d_2 \in D \; \exists o_1, o_2 \in \delta\bullet \; \exists \rho_1 \in Paths(o_1, d_1) \; \exists \rho_2 \in Paths(o_2, d_2) : \quad [\rho_1] \cap [\rho_2] = \varnothing$$

Let $\Delta(D) := \{\delta \in P \cup T \mid \delta \lll D\}$ be the set of all diverging points of D.

For a node $o_1 \in \delta\bullet$ and the set D, the set $\mathscr{R}_D(o_1) = \{d \in D \mid Paths(o_1, d) \neq \varnothing\}$ defines those nodes of $d \in D$ for which there is a path from o_1 to d. ⌟

In the AFC-WF-net of Fig. 4, the set $D = \{x,y,z\} = \langle [x,y,z] \rangle$ of places has the diverging points $\Delta(\{x,y,z\}) = \{t1, t2, t3, i\}$. For example, there are disjoint paths from $t1$ to x and from $t1$ to y. For the outputs $\{x,y,c\}$ of $t1$, the subsets of nodes of $\{x,y,z\}$ they have paths to are $\mathscr{R}_{\{x,y,z\}}(x) = \{x\}$, $\mathscr{R}_{\{x,y,z\}}(y) = \{y\}$, and $\mathscr{R}_{\{x,y,z\}}(c) = \varnothing$. In the example AFC-WF-net in Fig. 5, the set $\{p7, p8, p9\}$ has $t4$ and $t1$ as diverging points although $t1$ has no disjoint paths for $p7$ and $p8$ (as it follows the same output).

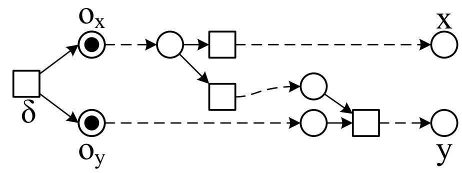


Fig. 8. A diverging transition δ for two places x and y has two output places o_x and o_y with two disjoint paths ρ_x from o_x to x and ρ_y from o_y to y, respectively. Through simple free-choiceness, each decision on ρ_x and ρ_y may lead to a remaining token on ρ_x and ρ_y, respectively. In addition, soundness ensures that transitions on ρ_x and ρ_y, respectively, with multiple input places will be fired. Furthermore, since $x \parallel y$ there are no paths between x and y and vice versa by Lemma 2, i. e., no transition on ρ_x and ρ_y may require the token of y and x, respectively.

Assume a transition δ is a diverging point of two concurrent places x and y, $\delta \lll \{x, y\}$. In this case, the tokens can "travel" from this transition via disjoint paths to x and y, i. e., from all markings in which δ is enabled, there is at least one reachable marking, in which x and y have tokens:

Lemma 4. *Let $N = (P, T, F, i, o)$ be a sound AFC-WF-net and $x, y \in P$. It holds:*

$$x \parallel y \quad \wedge \quad \delta \in \big(\Delta(\{x,y\}) \cap T\big)$$
$$\Longrightarrow$$
$$\forall M_\delta \in R(N, [i]),\ \delta\bullet \subseteq M_\delta\ \exists M_{xy} \in R(N, M_\delta) \colon [x, y] \subseteq M_{xy}$$ ⌟

Proof (Lemma 4). Constructive proof. By $\delta \in \big(\Delta(\{x,y\}) \cap T\big)$ and Definition 11, it follows that there are two disjoint paths $\rho_x \in Paths(o_x, x)$ and $\rho_y \in Paths(o_y, y)$ from two output places $o_x, o_y \in \delta\bullet$ to x and y, $[\rho_x] \cap [\rho_y] = \varnothing$.

Since N is sound, there is at least one reachable marking $M_\delta \in R(N, [i])$ with $\delta\bullet \subseteq M_\delta$ (any marking after firing δ). Let M_δ be such a marking with $o_x, o_y \in M_\delta$.

N is (simple) free-choice, i. e., for each transition t on ρ_x and ρ_y, $t \in \big(([\rho_x] \cup [\rho_x]) \cap T\big)$, with $|\bullet t| \geqslant 2$, it holds $\forall p \in \bullet t \colon |p\bullet| = 1$. Since $[\rho_x] \cap [\rho_y] = \varnothing$ and since $\forall p \in P \colon |p\bullet| \geqslant 2 \implies \bullet(p\bullet) = \{p\}$ (simple free-choiceness), we can force tokens on ρ_x and ρ_y (starting with o_x and o_y) to always put a token on a place on ρ_x and ρ_y, respectively (i. e., transitions on ρ_x and ρ_y fire regardless if they are outputs of diverging places, which could also "take another path"). Since N is sound, there is no dead transition (on ρ_x and ρ_y). Furthermore, since $x \parallel y$ and N is sound there are no paths between x and y and there are no paths between y and x by Lemma 2, i. e., no transition on ρ_x and ρ_y may "require" the token of y and x to fire, respectively. Fig. 8 illustrates the situation. For this reason, at least until reaching x and y, it is possible that the number of tokens on ρ_x and ρ_y will not decrease. Eventually, tokens on ρ_x and ρ_y "arrive" at x and y in a marking M_{xy} that is reachable from M_δ, i. e., $\forall M_\delta \in R(N, [i]),\ \delta\bullet \subseteq M_\delta\ \exists M_{xy} \in R(N, M_\delta) \colon [x, y] \subseteq M_{xy}$. ✓ □

Assume there is a maximum admissible marking M_m and it shall be checked if it is reachable from the initial marking. If all diverging points $\Delta(M_m)$ are transitions

($\Delta(M_m) \subseteq T$), then we can imply that there is a run net, which contains these $\Delta(M_m)$ transitions with all their output places and the paths between them. Thus, by Corollary 1, it directly follows that M_m is reachable. This is the simplest case. Unfortunately, places can also be diverging points as the above examples have already shown. This complicates the situation since it is not sure which output transition of a diverging place we should add to a run net to check the reachability of M_m. Trying each combination of output transition may lead again to an exponential growth of combinations. Fortunately, soundness and admissibility of M_m simplifies matters as with a sound AFC-WF-net and a admissible marking, there is always at least one output transition to take since the output transitions of such diverging places must always share the nodes to which they have paths to:

Lemma 5. *Let $N = (P,T,F,i,o)$ be a sound AFC-WF-net with a marking $M \in \mathscr{B}(P)$, a place as diverging point $\delta \in \big(\Delta(M) \cap P\big)$, and $\{o_x, o_y\} \subseteq \delta\bullet$. It holds:*

$$M \text{ is admissible} \quad \Longrightarrow \quad \big(\mathscr{R}_{\langle M\rangle}(o_x) \subseteq \mathscr{R}_{\langle M\rangle}(o_y)\big) \vee \big(\mathscr{R}_{\langle M\rangle}(o_y) \subseteq \mathscr{R}_{\langle M\rangle}(o_x)\big). \quad \lrcorner$$

Proof (Lemma 5). Proof by contradiction.

$$M \text{ is admissible } \wedge\ \mathscr{R}_{\langle M\rangle}(o_x) \nsubseteq \mathscr{R}_{\langle M\rangle}(o_y)\ \wedge\ \mathscr{R}_{\langle M\rangle}(o_x) \nsubseteq \mathscr{R}_{\langle M\rangle}(o_y) \tag{6}$$

$$\Longrightarrow \quad \exists x \in \mathscr{R}_{\langle M\rangle}(o_x)\ \exists y \in \mathscr{R}_{\langle M\rangle}(o_y): \quad x \notin \mathscr{R}_{\langle M\rangle}(o_y) \wedge y \notin \mathscr{R}_{\langle M\rangle}(o_x) \tag{7}$$

Let $x, y \in M$ be such places. There are exactly two cases:

Case 1: $Paths(o_x,y) \neq \varnothing \vee Paths(o_y,x) \neq \varnothing$. Then, however, $y \in \mathscr{R}_{\langle M\rangle}(o_x)$ or $x \in \mathscr{R}_{\langle M\rangle}(o_y)$ by Definition 11 of diverging points. This contradicts (7) and this case does not hold. ↯

Case 2: $Paths(o_x,y) = \varnothing \wedge Paths(o_y,x) = \varnothing$. Thus:

$$\forall \rho_x \in Paths(o_x,x)\ \forall \rho_y \in Paths(o_y,y): [\rho_x] \cap [\rho_y] = \varnothing. \tag{8}$$

For this reason, concurrency of x and y cannot "start" from $[\delta]$. Fig. 9 illustrates the situation in this case.

Let $\rho_x \in Paths(o_x,x)$ and $\rho_y \in Paths(o_y,y)$. By the admissibility of M in Eq. (6) and its Definition 8, it follows $x \parallel y$. Therefore, all paths from x and from y to the sink o join at first at converging transitions by Lemma 3. Let $c \in T$ be such a converging transition and $\rho_{cx} \in Paths(x,c_x)$ with $c_x \in \bullet c$ is a path without any other converging transition of x and y (i. e., c is the "first" of such transitions). There is also a disjoint path $\rho_{cy} \in Paths(y,c_y)$ with $c_y \in \bullet c$ by Lemma 3, $[\rho_{cy}] \cap [\rho_{cx}] = \varnothing$. As (a) there is no path from o_x to y, (b) there is no path from o_y to x, (c) c_x is the first converging transition on ρ_{cx}, and (d) N is sound and simple free-choice ($\bullet o_x = \bullet o_y = \{\delta\}$), whether o_x or o_y is fired is independent ("free") from other tokens in any reachable marking M of N. For this reason and N is sound, let $M_\delta \in R(N,[i])$ and $M_{c_y,\delta} \in R(N,M_\delta)$ be two reachable markings with $\delta \in M_\delta$ and $[c_y,\delta] \subseteq M_{c_y,\delta}$ so that c can be fired if δ decides for o_x. However, in any $M_{c_y,\delta}$, there is an "unsafe" path from δ via c_y to the sink o by Theorem 1. This contradicts (6) that N is sound. ↯ This case does not hold.

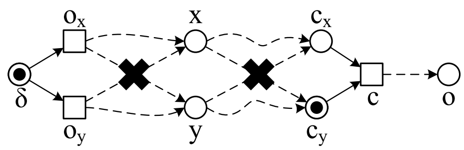


Fig. 9. A diverging place δ has two output transitions o_x and o_y. o_x has a path to x but not to y and o_y has a path to y but not to x. By $x \parallel y$ and Lemma 3, there is a "first" joining transition c. Thus, there is no path from y to c_x and there is no path from x to c_y. If δ decides for o_x, no token "after" δ can reach to c_y enabling c. Thus, since soundness is required, there must be a marking $M_{c_y,\delta}$ with $[c_y,\delta] \subseteq M_{c_y,\delta}$ to avoid a dead c. However, the path from δ via o_y, c_y, and c to o is not safe.

Both cases do not hold. Thus, contradiction (6) does not hold. ↯ Thus, the lemma must hold. □

The above Lemma 5 guarantees that the output transitions $\delta\bullet$ of a diverging place δ of an admissible marking M_m either have paths to the same subset of places of M_m or at least one output transition has paths to more places of M_m. To simplify what matters: there is always one output transition of a diverging place that has paths to the union of all subsets of M_m of all other output transitions. Since we are interested in the (sub-marking) reachability of all places of M_m, it is always the best option to take that output transition with the most reachable places of M_m. In summary, if we derive $\Delta(M_m)$ for an admissible marking M_m, we could take those output transitions of diverging places containing the greatest subset of places of M_m.

The remaining open question is if there is at least one transition in $\Delta(M_m)$, which has paths to all places in M_m. By Definition 11, such a transition $\delta \in \Delta(M_m)$ exists if $\bigcup_{o_\delta \in \delta\bullet} \mathscr{R}_{\langle M_m \rangle}(o_\delta) = M_m$. If such δ exists, then we can imply a partial run net starting by δ and adding all transitions and necessary paths to these transitions as well as to the places of M_m and the necessary paths to those places. Although this partial run net may not be a "full" run net, this partial run net implies the existence of a "full" run net.

Let us consider our example in Fig. 1. Assume, we want to check if the marking $M_m = [p3, p8, p14, p17]$ is reachable. All places are pairwise concurrent and it is not possible to add any further place (M_m is maximum admissible). For this reason, we have to check whether there is a diverging transition, which can cause this marking. The diverging points for $M_m = [p3, p8, p14, p17]$ cover the set $\{t10, t11, t12, p16, t1\}$ by Definition 11. $t10$'s output places $p14$ and $p17$ are part of M_m, thus, $\mathscr{R}_{\langle M_m \rangle}(p14) = \{p14\}$ and $\mathscr{R}_{\langle M_m \rangle}(p17) = \{p17\}$. For $t12$'s output places $p3$ and $p18$, it holds $\mathscr{R}_{\langle M_m \rangle}(p3) = \{p3\}$ and $\mathscr{R}_{\langle M_m \rangle}(p18) = \{p14, p17\}$. For $t11$, the information are similar $\mathscr{R}_{\langle M_m \rangle}(p3) = \{p3\}$ and $\mathscr{R}_{\langle M_m \rangle}(p13) = \{p14, p17\}$. The place $p16$ is a decision with two output transitions $t11$ and $t12$. By Lemma 5, both transitions either have the same information ($\mathscr{R}_{\langle M_m \rangle}(t11) = \mathscr{R}_{\langle M_m \rangle}(t12) = \{p3, p14, p17\}$) or one is a superset of the other. Eventually, transition $t1$ has the output places $p4$ with $\mathscr{R}_{\langle M_m \rangle}(p4) = \{p8\}$ and $p2$ with $\mathscr{R}_{\langle M_m \rangle}(p2) = \{p3, p14, p17\}$. Now, we can induce a partial run net from this information. Fig. 10 illustrates the induction steps. First, we can add the diverging point $t1$, which is a diverging point of all places of M_m. Furthermore, we add its output places $p2$

and $p4$ as well as a path from $p1$ (the source) to $t1$. We further add $p8$ as a place of M_m and a path to it as we know that $p4$ has a disjoint path to $p8$. The diverging place $p16$ with a path to it is added next (all paths from $t1$ to $t11$ or $t12$ pass $p16$). Furthermore, we add that output transition with the most information (i. e., that output transition t for which $\mathscr{R}_{\langle M\rangle}(t)$ is the greatest). In this case, it does not matter, which of $t11$ or $t12$ we add as they have the same information. In the next step, we add the diverging transition $t12$ (which we already have inserted) with its output places $p3$ and $p18$. Eventually, diverging transition $t10$ is added with a path to it as well as its output places. This final partial run net contains all places of $M_m = [p3, p8, p14, p17]$ and implies the existence of a "full" run net containing all places of M_m. Such steps are applied to finally deciding the (sub-marking) reachability in AFC-WF-nets:

Theorem 4 ((Sub-Marking) Reachability). *Let $N = (P, T, F, i, o)$ be a sound AFC-WF-net with a marking $M \in \mathscr{B}(P)$. It holds:*

$$M \in R(N,[i]) \iff M \textit{ is maximum admissible } \wedge \; \exists \delta \in (\Delta(M) \cap T)\colon \bigcup_{o_\delta \in \delta\bullet} \mathscr{R}_{\langle M\rangle}(o_\delta) = M \tag{9}$$

$$\exists M_s \in R(N,[i])\colon M \subseteq M_s \iff M \textit{ is admissible } \wedge \; \exists \delta \in (\Delta(M) \cap T)\colon \bigcup_{o_\delta \in \delta\bullet} \mathscr{R}_{\langle M\rangle}(o_\delta) = M \tag{10}$$

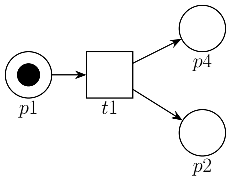
⌟

Proof (Theorem 4). Constructive proof. By Corollary 1 it holds:

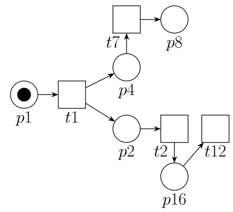
$$M \in R(N,[i]) \iff M \text{ is maximum admissible } \wedge \exists (P_\sigma, T_\sigma, F_\sigma) \in \Pi(N)\colon \langle M\rangle \subseteq P_\sigma.$$

Furthermore, by Lemma 5, it follows for each diverging transition $\delta \in \Delta(M) \cap T$ and for each of its output places $o_\delta \in \delta\bullet$ that there is at least one sub-net of all $\mathscr{R}_{\langle M\rangle}(o_\delta)$ from o_δ to all $x \in \mathscr{R}_{\langle M\rangle}(o_\delta)$, which diverges only in transitions. As a consequence, for each diverging transition $\delta \in \Delta(M) \cap T$, the union $\bigcup_{o_\delta \in \delta\bullet} \mathscr{R}_{\langle M\rangle}(o_\delta)$ describes the subset $M' \subseteq M$ of M to which the transition δ has such a sub-net to, which only diverges in transitions. If there is no diverging transition $\delta \in \Delta(M) \cap T$ with $M = \bigcup_{o_\delta \in \delta\bullet} \mathscr{R}_{\langle M\rangle}(o_\delta)$, then there is no run net that can contain a sub-net, which only diverges at transitions to the places of M. Therefore, there is no run net that contains all places of M. If there is such a diverging transition δ with $M = \bigcup_{o_\delta \in \delta\bullet} \mathscr{R}_{\langle M\rangle}(o_\delta)$, then there must be at least one run net, which contains all places in M. ✓ The proof of sub-marking reachability by (10) follows the same argumentation. ✓ ◻

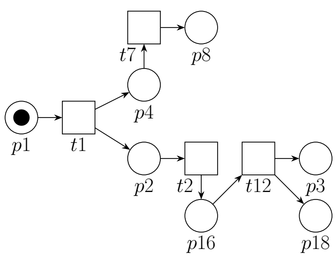
With Theorem 4, we can decide if a (sub-)marking is reachable based on the structure of a sound AFC-WF-net rather than over its concrete behavior in terms of occurrence sequences, state space exploration, or similar techniques. In addition, maximum admissibility and diverging points help to understand why a given specific marking is *not* reachable. For example, the marking $[p8, p5, p14]$ in Fig. 1 is *not* reachable from $[p1]$ since $p8$ is not concurrent to $p5$ (there is a path from $p8$ to $p5$ following Lemma 2). In Fig. 4, the marking $[x, y, z]$ is not reachable from $[i]$ as there is no diverging transition, which leads to simultaneous tokens on x, y, and z — just on x and y, x and z, or y and z. This diagnostic information may support repairing improperly developed nets, in which a desired marking is not reachable or an undesired marking is reachable.



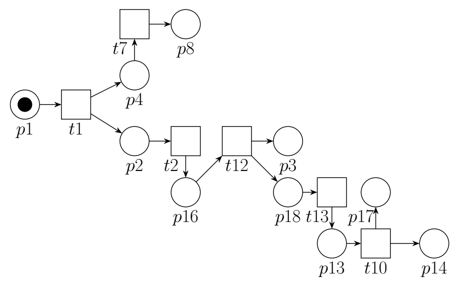
Adding diverging point $t1$ with its output places $p2$ and $p4$.



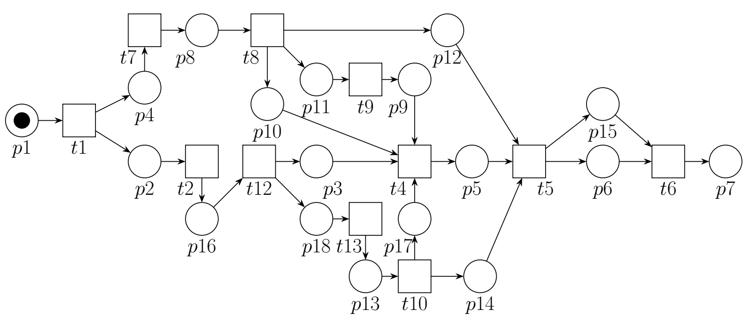
Adding a path from $p4$ to $p8$. Furthermore, adding diverging point $p16$ with its output transition $t12$. It does not matter, which of $t11$ or $t12$ is added as they have the same information ($\mathscr{R}_{\langle M_m \rangle}(t11) = \mathscr{R}_{\langle M_m \rangle}(t12)$).



Adding diverging point $t12$ with its output places $p3$ and $p18$.



Adding a path to the diverging point $t10$ with its output places $p14$ and $p17$. This net contains all places $[p3, p8, p14, p17]$ and …



… implies the existence of a run net.

Fig. 10. Step-wisely inducing a partial run net for the exemplary AFC-WF-net in Fig. 1 out of the diverging point information for the marking $M_m = [p3, p8, p14, p17]$. The final partial run net implies the existence of a run net presented at the end.

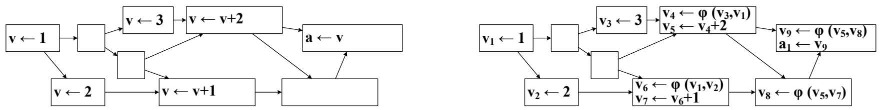


Fig. 11. Transforming a program with program variables v and a (left) into SSA form (right) by inserting new definitions of variables and ϕ-functions at converging nodes.

Algorithmic Derivation: There is a concept in compiler theory to identify diverging points: the placement of so-called ϕ-functions to derive the minimal *Static Single Assignment* (SSA) form [6]. The SSA form requires that each *program variable* in a computer program gets statically only once a value. If there are converging control-flows, a ϕ-function selects that program variable, which was used during execution and assigns it to a new program variable. Figure 11 illustrates an exemplary control-flow graph (left) being transferred into SSA form (right). Cytron et al. [6] derived algorithms to compute the *minimal* SSA form from normal programs. The minimal SSA form only inserts ϕ-functions if necessary. To achieve a SSA form out of a computer program, the transformation inserts a new program variable (which is called a *definition*) for each assignment of this variable. Subsequently, it inserts a ϕ-function of the form $d_n = \phi(d_1, \ldots, d_m)$, $m \geqslant 2$, for each program variable at each program point where different definitions $d_1, \ldots, d_m$ can occur. This placement of ϕ-functions in the minimal SSA assumes fairness, i. e., each path can be followed by a token. The positions of nodes where a ϕ-function is required for a program variable d are positions where two paths from two different definitions of the same variable first meet. As a consequence, for all $d_1, \ldots, d_m$ there are disjoint paths from the position of the ϕ-function to their assignment. Actually, this is the same definition as for diverging points, only in the reverse direction.

We can simply reuse the algorithm of placing ϕ-functions to identify all diverging points $\Delta(M)$ of a given admissible marking M and to compute the sets $\mathscr{R}_{\langle M \rangle}(o_\delta)$ for all $o_\delta \in \bigcup_{\delta \in \Delta(M)} \delta\bullet$. In doing so, we assume that on all places in M there is an assignment on the same single program variable v. Then, we perform the ϕ-function placements, however, on the *reverse* net, i. e., we change the direction of each flow. Thus, the SSA algorithm will replace the assignments on the program variable v with unique definitions of v. On all nodes, from where two different definitions have disjoint paths, the SSA approach has placed a ϕ-function with those definitions as arguments and a new definition where the result of the ϕ-function is assigned. It is important to note that the different unique definitions allow for collecting $\mathscr{R}_{\langle M \rangle}(o)$ for the outputs o of the diverging points and identifying disjoint paths easily.

Algorithm 1 summarizes the steps of checking (sub-marking) reachability of a given marking M_r in a sound AFC-WF-net N. At first, (maximum) admissibility of M_r is checked in line 2. If M_r is *admissible* or *maximum admissible*, then M_r may be (sub-marking) reachable. Otherwise, if M_r is *not admissible*, M_r is stated to be *not reachable*. Line 4 computes the diverging points of M_r with the ϕ-placement algorithm of building the minimal SSA form [6] in the reverse net of N. This placement includes for all outputs o of all diverging points those diverging points and places of M_r, $\mathscr{R}_{\langle M_r \rangle}(o)$, to

Algorithm 1 Checking (sub-marking) reachability of a given marking $M_r \in \mathscr{B}(P)$ for a sound AFC-WF-net N.

1: **function** ISREACHABLE(N, M_r)
2:     $admissibility \leftarrow$ CHECKMAXIMUMADMISSIBILITY(N, M_r)
3:     **if** $admissibility \in \{$ admissible, maximum admissible $\}$ **then**
4:         Compute $\Delta(M_r)$ of N by the ϕ-placement algorithm.
5:         **for all** $\delta \in (\Delta(M_r) \cap T)$ **do**
6:             $M_\delta \leftarrow \bigcup_{o_\delta \in \delta \bullet} \mathscr{R}_{\langle M_r \rangle}(o_\delta)$
7:             **if** $M_r = M_\delta$ **then**
8:                 **if** $admissibility =$ admissible **then**
9:                     **return** sub-marking reachable
10:                 **else**
11:                     **return** reachable
12:     **return** not reachable

which they have paths. Finally, all diverging transitions δ are investigated in lines 5–11. The set M_δ is computed in line 6 regarding Theorem 4. Line 7 checks if M_r is equal to M_δ. If this check holds, the marking is either *sub-marking reachable* (if M_r is *admissible*) or *reachable* (if M_r is maximum admissible). Otherwise, other diverging transitions are investigated until either there is one, for which the check holds, or the marking is stated as *not reachable*. This algorithm does not require checking reachability by investigating the state space, computing occurrence sequences, etc. This explains why the computational complexity is just quadratic in the worst case:

Theorem 5 (Computational Complexity). *Let $N = (P, T, F, i, o)$ be a sound AFC-WF-net and M_r a marking to check. Deciding reachability of M_r can be computed in $O(|P|^2 + |T|^2)$.* ⌟

Proof (Theorem 5). Constructive proof by line-by-line investigation of Algorithm 1.

Line 2: Checking admissibility and maximum admissibility can be achieved in quadratic computational time complexity, $O(|P|^2 + |T|^2)$, by Theorem 3.

Line 4: Following Cytron et al. [6], placing ϕ-functions requires to compute the "dominance frontier", which is feasible in quadratic time, $O(|P|^2 + |T|^2)$, by Cooper et al. after building the "tree of dominance" in the same complexity $O(|P|^2 + |T|^2)$ [5]. The placement of ϕ-functions for M_r can also be achieved in quadratic time $O(|P|^2 + |T|^2)$ for a single variable by Cytron et al. [6] as it is needed in our case. Thus, overall, this line takes $O(|P|^2 + |T|^2)$.

Lines 5–11: Lines 6–11 are investigated $O(|T|)$ times by line 5. Computing M_δ is linear $O(|P|)$ with an implementation requiring constant time for set unions (e. g., a BitSet in Java). The remaining lines 7-11 are at most $O(|P|)$. Thus, lines 5–11 can be achieved in $O(|T| \cdot |P|)$ in general.

In summary, checking the (sub-marking) reachability of a given marking is dominated by the term $O(|P|^2 + |T|^2)$. ✓

Termination: All steps in Algorithm 1 are either known to terminate (as placing ϕ-functions) or are performed over finite sets (as over the set of diverging points, M_r, etc.). For this reason, Algorithm 1 terminates. □

5 Conclusion

This paper showed for the Petri net class of sound acyclic free-choice workflow nets that all places of a reachable (sub-)marking must be pairwise concurrent (the marking is admissible). Furthermore, reachable markings must be maximum admissible, i. e., they are admissible and adding a further place to them would destroy their admissibility. Since maximum admissibility is just a necessary condition of (sub-marking) reachability, this paper introduced diverging transitions as sufficient condition, whose existence manifests in subgraphs of sound acyclic free-choice workflow nets representing possible occurrence graphs and, therefore, enabling the reachability of (maximum) admissible markings. The overall approach presented in this paper has a $O(|P|^2 + |T|^2)$ computational complexity. Furthermore, the new concepts of admissibility, maximum admissibility, and diverging transitions allow for explaining *why* a given (sub-)marking is *not* reachable.

The Petri net and business process management communities get a new computationally efficient algorithm to check reachability, especially for process-like and industrial related nets. Since the reachability of markings is central to many other decision problems of Petri nets, the here presented approach empowers to speed up other algorithms. For example, business analysts could request their process models if undesired states of their models could happen to improve quality, check compliance, and to avoid cost-intensive failures.

In the future, the concepts of admissibility and maximum admissibility shall be generalized to the class of proper free-choice nets with a home cluster. Furthermore, the method of loop decomposition [18] shall be introduced for free-choice nets leading to an unfolding of loops increasing a model just quadratically in the worst case. For industrial settings, we want to provide an implementation of our approach, which allows for checking reachability for process models by simply selecting some nodes of the model. Furthermore, it is of interest whether the new concepts introduced in this paper may be applicable to solve other problems in Petri nets.

References

1. van der Aalst, W.M.P.: Verification of workflow nets. In: Azéma, P., Balbo, G. (eds.) Application and Theory of Petri Nets 1997, 18th International Conference, ICATPN '97, Toulouse, France, June 23-27, 1997, Proceedings. Lecture Notes in Computer Science, vol. 1248, pp. 407–426. Springer (1997). `https://doi.org/10.1007/3-540-63139-9_48`
2. van der Aalst, W.M.P.: Free-choice nets with home clusters are lucent. Fundam. Informaticae **181**(4), 273–302 (2021). `https://doi.org/10.3233/FI-2021-2059`
3. van der Aalst, W.M.P., van Hee, K.M., ter Hofstede, A.H.M., Sidorova, N., Verbeek, H.M.W., Voorhoeve, M., Wynn, M.T.: Soundness of workflow nets: classification, decidability, and analysis. Formal Aspects Comput. **23**(3), 333–363 (2011). `https://doi.org/10.1007/s00165-010-0161-4`
4. Cheng, A., Esparza, J., Palsberg, J.: Complexity results for 1-safe nets. Theor. Comput. Sci. **147**(1&2), 117–136 (1995). `https://doi.org/10.1016/0304-3975(94)00231-7`
5. Cooper, K.D., Harvey, T.J., Kennedy, K.: A simple, fast dominance algorithm. Technical Report TR-06-33870, Institut für Informatik, Technische Universität München, Rice University (2001)

6. Cytron, R., Ferrante, J., Rosen, B.K., Wegman, M.N., Zadeck, F.K.: Efficiently computing static single assignment form and the control dependence graph. ACM Trans. Program. Lang. Syst. **13**(4), 451–490 (1991). `https://doi.org/10.1145/115372.115320`, `https://doi.org/10.1145/115372.115320`
7. Czerwinski, W., Lasota, S., Lazic, R., Leroux, J., Mazowiecki, F.: The reachability problem for Petri nets is not elementary. J. ACM **68**(1), 7:1–7:28 (2021)
8. Desel, J., Esparza, J.: Reachability in cyclic extended free-choice systems. Theor. Comput. Sci. **114**(1), 93–118 (1993). `https://doi.org/10.1016/0304-3975(93)90154-L`, `https://doi.org/10.1016/0304-3975(93)90154-L`
9. Desel, J., Esparza, J.: Free Choice Petri Nets. Cambridge University Press, pbk version ed. edition edn. (2008)
10. van Dongen, B.F., Mendling, J., van der Aalst, W.M.P.: Structural patterns for soundness of business process models. In: Tenth IEEE International Enterprise Distributed Object Computing Conference (EDOC 2006), 16-20 October 2006, Hong Kong, China. pp. 116–128. IEEE Computer Society (2006). `https://doi.org/10.1109/EDOC.2006.56`
11. Esparza, J.: Reachability in live and safe free-choice Petri nets is NP-complete. Theor. Comput. Sci. **198**(1-2), 211–224 (1998). `https://doi.org/10.1016/S0304-3975(97)00235-1`, `https://doi.org/10.1016/S0304-3975(97)00235-1`
12. Esparza, J., Nielsen, M.: Decidability issues for Petri nets - a survey. CoRR **abs/2411.01592** (2024). `https://doi.org/10.48550/ARXIV.2411.01592`, `https://doi.org/10.48550/arXiv.2411.01592`
13. Favre, C., Fahland, D., Völzer, H.: The relationship between workflow graphs and free-choice workflow nets. Inf. Syst. **47**, 197–219 (2015)
14. Kovalyov, A., Esparza, J.: A Polynomial Algorithm to Compute the Concurrency Relation of Free-Choice Signal Transition Graphs. Sonderforschungsbereich 342: Methoden und Werkzeuge für die Nutzung paralleler Rechnerarchitekturen TUM-19528, SFB-Bericht Nr. 342/15/95 A, Institut für Informatik, Technische Universität München, München, Germany (Oct 1995)
15. Murata, T.: Petri nets: Properties, analysis and applications. Proc. IEEE **77**(4), 541–580 (1989). `https://doi.org/10.1109/5.24143`, `https://doi.org/10.1109/5.24143`
16. Polyvyanyy, A., Weidlich, M., Conforti, R., Rosa, M.L., ter Hofstede, A.H.M.: The 4C Spectrum of fundamental behavioral relations for concurrent systems. In: Ciardo, G., Kindler, E. (eds.) Application and Theory of Petri Nets and Concurrency - 35th International Conference, PETRI NETS 2014, Tunis, Tunisia, June 23-27, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8489, pp. 210–232. Springer (2014)
17. Prinz, T.M., Choi, Y., Ha, N.L.: Understanding and decomposing control-flow loops in business process models. In: Ciccio, C.D., Dijkman, R.M., del-Río-Ortega, A., Rinderle-Ma, S. (eds.) Business Process Management - 20th International Conference, BPM 2022, Münster, Germany, September 11-16, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13420, pp. 307–323. Springer (2022)
18. Prinz, T.M., Choi, Y., Ha, N.L.: Soundness unknotted: An efficient soundness checking algorithm for arbitrary cyclic process models by loosening loops. Inf. Syst. **128**, 102476 (2025). `https://doi.org/10.1016/J.IS.2024.102476`, `https://doi.org/10.1016/j.is.2024.102476`
19. Prinz, T.M., Klaus, J., van Beest, N.R.T.P.: Pushing the limits: Concurrency detection in acyclic sound free-choice workflow nets in $O(P^2 + T^2)$. In: Köhler-Bussmeier, M., Moldt, D., Rölke, H. (eds.) Proceedings of the International Workshop on Petri Nets and Software Engineering 2024 co-located with the 45th International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS 2024), June 24 - 25, 2024, Geneva, Switzerland. CEUR Workshop Proceedings, vol. 3730, pp. 132–154. CEUR-WS.org (2024), `https://ceur-ws.org/Vol-3730/paper08.pdf`

20. Prinz, T.M., Welsch, T., Ha, N.L.: Recognizing relationships: Detecting the 4C Spectrum in $o(p^2 + t^2)$ for acyclic sound process models. In: Borbinha, J., Sales, T.P., da Silva, M.M., Proper, H.A., Schnellmann, M. (eds.) Enterprise Design, Operations, and Computing - 28th International Conference, EDOC 2024, Vienna, Austria, September 10-13, 2024, Revised Selected Papers. Lecture Notes in Computer Science, vol. 15409, pp. 281–299. Springer (2024). https://doi.org/10.1007/978-3-031-78338-8_15, https://doi.org/10.1007/978-3-031-78338-8_15
21. Verbeek, H.M.W., Basten, T., van der Aalst, W.M.P.: Diagnosing workflow processes using Woflan. Comput. J. **44**(4), 246–279 (2001). https://doi.org/10.1093/comjnl/44.4.246
22. Yamaguchi, S.: Polynomial time verification of reachability in sound extended free-choice workflow nets. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **97-A**(2), 468–475 (2014). https://doi.org/10.1587/TRANSFUN.E97.A.468