On Free Choice and Wil(l): From Petri Net Theory to Process Mining

Christopher T. Schwanen[1(✉)] and Wied Pakusa[2]

[1] Chair of Process and Data Science (PADS), RWTH Aachen University, Aachen, Germany
schwanen@pads.rwth-aachen.de
[2] Faculty of Mathematics, Informatics and Technology, Koblenz University of Applied Sciences, Koblenz, Germany
pakusa@hs-koblenz.de

Abstract. Workflow nets were introduced by Wil in the mid-90s as a subclass of Petri nets tailored towards modelling business processes. They are now the de-facto standard in process mining. We study the complexity of the reachability problem on workflow net classes with the varying properties of quasi-liveness, free-choice and the option to complete. This yields eight classes in total. A workflow net has the option to complete if the final marking is reachable from every state and it is quasi-live if every transition can be fired in some reachable marking. Our main result is a strong dichotomy: on safe workflow nets, the reachability problem is either in PTIME or PSPACE-complete (assuming PTIME and PSPACE are different). This dichotomy becomes all the more apparent when we drop safeness: then, the reachability problem is either in PTIME or becomes at least PSPACE-hard or even Ackermann-complete. More specifically, we show that the properties of being free-choice and having the option to complete are necessary and sufficient to obtain PTIME-complexity. Our proofs make strong use of several of Wil's results on the structure theory of Petri nets, in particular of his work on lucency.

Keywords: Process Mining · Petri Net Theory · Free-Choice Petri Nets · Workflow Nets · Reachability · Wil van der Aalst

1 Introduction

Algorithms are the technological backbone of process mining. In order to extract insights from huge amounts of event data, efficient and scalable approaches are required to make real-world process analyses possible. In process mining, the de-facto standard for process modelling are *workflow nets*, which Wil introduced in the mid-90s [1–4]. They are intended to describe business procedures by modelling the behaviour of a single case with a clear start and termination point. Thereby, the behavioural concept of *soundness* arises naturally because every case of the business procedure should (i) eventually terminate (*option to complete* (OC)), (ii) when terminating all other places of the workflow net should

J. Mendling et al. (Eds.): Wil van der Aalst Festschrift, LNCS 16480, pp. 298–311, 2026.
https://doi.org/10.1007/978-3-032-17618-9_22

be empty (*proper completion*), and (iii) every task of the procedure should be executable (*quasi-liveness* (QL)).

Workflow nets yield a strong process modelling framework. In particular, their ability to express concurrency allows to succinctly encode exponential-sized state spaces, often referred to as the *state space explosion problem*. However, it has been observed that real-world processes do not require the full expressive power of workflow nets. In many cases, real-world processes bring extra structure that can be exploited to simplify the algorithmic analyses. Probably, the most well-studied of these restrictions is the so-called *free-choice* (FC) property. The intuition behind free-choice nets is that decisions are non-conflicting in the following sense: if two tasks require a common condition or resource, then they actually require precisely the same conditions or resources. In other words, whenever a condition precedes or a resource can be consumed by different tasks (i.e., we observe a choice), then such tasks can be performed in precisely the same situations (i.e., we can *freely choose* one among them). First heavily explored in the Petri net community, Wil recognised its huge potential for process mining applications very early, e.g., see [3]. Since then, the structure theory of free-choice workflow nets has been applied successfully to reduce the (algorithmic) complexity of many problems in process mining. As many real-world process models enjoy the free-choice property, this connection has a strong impact on everyday process mining applications.

Sound free-choice workflow nets enjoy a further structural property which Wil coined *lucency* [6–9]. A process model is *lucent* if its state is uniquely determined by the set of enabled transitions. In other words, if two reachable markings enable the same set of transitions, then they are identical. Here, the intuition is that a different state always comes with different actions enabled to perform, implying that states are fully observable by the possible behaviour. Wil also introduced the dual concept of *translucency* [7] for event logs, which requires that an event carries information about all enabled activities at the time it was recorded. Both properties are interesting concepts for academic investigation but, more importantly, they closely align with how processes behave in reality. It is surprising that Wil was the first one to discover and study lucency and translucency in depth and, from our point of view, this structural property deserves much more attention in the process mining and workflow management community.

In fact, Wil himself gave strong support for the applicability of these notions. He already proved in [6] that so-called *perpetual* free-choice nets are lucent. Here, a perpetual net is a live and bounded free-choice net which has a *home cluster*, i.e., a cluster such that in every reachable state it is possible to eventually mark the places of the home cluster and nothing more. Perpetual nets are a direct generalisation of sound workflow nets: the final marking of a sound workflow net induces a home cluster by the requirement of having the *option to complete* in combination with *proper completion*. While this earlier work mostly relied on the theory of live and bounded systems (most importantly, the Blocking Theorem [20]), Wil was later able to show that lucency holds for a much larger class of free-choice nets: he proved that a home cluster alone already implies lucency of

Table 1. Overview of complexity results for the reachability problem. Structural and behavioural properties of the considered workflow net classes are: quasi-liveness (QL), the option to complete (OC), free-choiceness (FC), and safeness (safe). Assumed properties are marked by •, thereby implied properties are marked by ○.

Workflow Net Properties				Complexity of the Reachability Problem	
QL	OC	FC	safe		
				Ackermann-complete	[16,21]
•		•		Ackermann-complete	Corollary 7
	•			∈ EXPSPACE	Corollary 6
•	•			PSPACE-hard	Theorem 6
			•	PSPACE-complete	[14,15]
•	•		•	PSPACE-complete	Theorem 6
•		•	•	PSPACE-complete	Corollary 8
○	•	•	○	∈ PTIME	Corollary 5

proper free-choice systems and that the additional assumptions of liveness and boundedness are not needed [8]. In particular, this implies the uniqueness (but not the existence) of blocking markings in free-choice nets with a home cluster. This result is remarkable also because Wil could not rely on the theory of live and bounded systems anymore, but had to develop completely new techniques and concepts which are of independent interest. Again, we believe that this line of work deserves much more attention and further investigations.

In this article, we celebrate the success story of free-choice workflow nets and Wil! Specifically, we take up Wil's idea from almost 30 years ago and use the structure theory of Petri nets in order to reduce the algorithmic complexity of the *reachability problem* on workflow nets. Asking whether two given markings are reachable from one another, the reachability problem is one of the classic algorithmic benchmarks in Petri net theory. Almost all non-trivial questions about the behaviour of Petri nets are algorithmically at least as hard as reachability and in many cases exhibit equivalent complexity. Our main result is a dichotomy for sound free-choice workflow nets: we show that as soon as we drop one of the soundness criteria 'option to complete (OC)' (which implies proper completion) or the 'free-choice property (FC)', deciding reachability becomes at least PSPACE-hard. Only 'quasi-liveliness (QL)' does not affect the complexity of the reachability problem. Table 1 highlights the main results of this paper.

2 Preliminaries

Let $\mathbb{N} := \{0, 1, 2, \dots\}$ denote the natural numbers.

Definition 1 (Multiset). A *multiset* M over a set A is a function $M: A \to \mathbb{N}$; thus, for any $a \in A$, $M(a)$ indicates how often a is contained in the multiset M. The set of all multisets over A is given by $\mathbb{N}^A$. We also use $[a^{M(a)} | a \in A]$ as notation for a multiset $M \in \mathbb{N}^A$. For multisets $M, M' \in \mathbb{N}^A$, we use the standard notation for functions, e.g., $M + M'$, $M \leq M'$, etc.

Definition 2 (Sequence). *Sequences* with index set I over a set A are denoted by $\sigma = \langle a_i \rangle_{i \in I} \in A^I$. The *length* of a sequence σ is written as $|\sigma|$ and the set of all finite sequences over A is denoted by A^*.

Definition 3 (Petri Net). A *Petri net* N is a bipartite directed graph $N = (P, T, F)$ where P and T, $P \cap T = \emptyset$ are disjoint finite sets of vertices and $F \subseteq (P \times T) \cup (T \times P)$ is the set of arcs. In a Petri net, P is called the set of *places*, T the set of *transitions*, and F the *flow relation*. Given a vertex $v \in P \cup T$, its *pre-set* $\bullet v$ and *post-set* $v\bullet$ are defined by $\bullet v := \{u \in P \cup T | (u, v) \in F\}$ and $v\bullet := \{u \in P \cup T | (v, u) \in F\}$. With regard to a place (transition), its pre- and post-set are also called *input* and *output transitions* (*places*).

Definition 4 (Proper Petri Net). A Petri net $N = (P, T, F)$ is *proper* if all transitions have input and output places, i.e., $\forall t \in T\colon \bullet t \neq \emptyset \neq t\bullet$.

Definition 5 (Free-Choice Petri Net). A Petri net $N = (P, T, F)$ is *free-choice* if any two transitions either share all or none of their input places, i.e., $\forall t, t' \in T\colon \bullet t = \bullet t' \vee \bullet t \cap \bullet t' = \emptyset$.

Definition 6 (Marking, System, Firing Rule). Given a Petri net $N = (P, T, F)$, a *marking* $M \in \mathbb{N}^P$ is a multiset where $M(p)$ is the number of *tokens* at place $p \in P$. A place $p \in P$ is *marked* at M if $M(p) > 0$. The pair (N, M) of a Petri net $N = (P, T, F)$ and a marking $M \in \mathbb{N}^P$ is called a *system*. A transition $t \in T$ is *enabled* in M, denoted by $(N, M)[t\rangle$, if and only if each of its input places $p \in \bullet t$ is marked, i.e., $\forall p \in \bullet t\colon M(p) > 0$. An enabled transition may *fire*, denoted by $(N, M)[t\rangle(N, M')$, and firing results in a new marking M':

$$M'(p) = \begin{cases} M(p) - 1 & \text{if } p \in \bullet t \wedge p \notin t\bullet, \\ M(p) + 1 & \text{if } p \notin \bullet t \wedge p \in t\bullet, \\ M(p) & \text{otherwise.} \end{cases}$$

A sequence of transitions $\sigma = \langle t_i \rangle_{i=1}^n \in T^*$ is called a *firing sequence* of (N, M) if for every transition t_i of the sequence holds that $(N, M_{i-1})[t_i\rangle$ and $(N, M_{i-1})[t_i\rangle(N, M_i)$ where $M_0 = M$ and $M_n = M'$. Firing such a sequence is denoted by $(N, M)[\sigma\rangle(N, M')$. The empty sequence $\langle\rangle$ is always enabled and firing the empty sequence leaves the marking unchanged, i.e., $(N, M)[\langle\rangle\rangle(N, M)$. A marking M' is *reachable* if a firing sequence $\sigma \in T^*$ exists such that M' is the resulting marking, i.e., $(N, M)[\sigma\rangle(N, M')$. The set of all reachable markings of (N, M) is denoted by $[N, M\rangle := \big\{M' \in \mathbb{N}^P | \, \exists \sigma \in T^*\colon (N, M)[\sigma\rangle(N, M')\big\}$.

Definition 7 (Boundedness, Safeness). Given some $k \in \mathbb{N}$, a system (N, M_0) with $N = (P, T, F)$ is *k-bounded* if k is a bound for any reachable marking, i.e., $\forall M \in [N, M_0\rangle\colon \forall p \in P\colon M(p) \leq k$. (N, M_0) is *safe* if it is 1-bounded.

Definition 8 ((Quasi-)Liveness). Let (N, M_0) with $N = (P, T, F)$ be a system. A transition $t \in T$ is *quasi-live* if there exists a reachable marking $M' \in [N, M_0\rangle$ which enables t. A transition $t \in T$ is *live* if it is quasi-live for every reachable marking $M \in [N, M_0\rangle$, i.e., $\forall M \in [N, M_0\rangle \colon \exists M' \in [N, M\rangle \colon (N, M')[t\rangle$. A system (N, M_0) is *(quasi-)live* if every transition $t \in T$ is (quasi-)live.

Definition 9 (Home Marking, Cyclic System). Let (N, M_0) be a system. A marking $M \in [N, M_0\rangle$ is a *home marking* if it can be reached from any reachable marking, i.e., $\forall M' \in [N, M_0\rangle \colon M \in [N, M'\rangle$. A system (N, M_0) is *cyclic* if its initial marking M_0 is a home marking.

Definition 10 (Cluster). Let $N = (P, T, F)$ be a Petri net and let $v \in P \cup T$. The *cluster* of v, denoted by $[v]_C$, is the minimal set of places and transitions that includes v, i.e., $v \in [v]_C$, all output transitions of contained places, i.e., $\forall p \in [v]_C \cap P \colon p\bullet \in [v]_C$, and all input places of contained transitions, i.e., $\forall t \in [v]_C \cap T \colon \bullet t \in [v]_C$.

3 Workflow Nets and the Soundness Property

Wil introduced workflow nets [3] to describe business procedures and model the behaviour of a single case with a clear start and termination point.

Definition 11 (Workflow Net). A *workflow net* $N = (P, T, F, i, o)$ is a Petri net (P, T, F) where $i, o \in P$ are special places, namely the single *source place* i with $\bullet i = \emptyset$ and the single *sink place* o with $o\bullet = \emptyset$, and the Petri net is strongly connected if a transition $\bar{t}$ with $\bullet\bar{t} = \{o\}$ and $\bar{t}\bullet = \{i\}$ is added, i.e., every vertex $v \in P \cup T$ of the Petri net is on a path from i to o. Implicitly, in workflow nets, the initial marking is $[i]$ and the final marking is $[o]$.

As such, workflow nets serve as acceptors. This is a notable difference to standard Petri nets, which are indeed commonly considered with an initial marking, but rarely with any final states. Alongside workflow nets, Wil also introduced the soundness property [3] to describe their 'well-behavedness' as business procedures should terminate eventually with leaving no tokens behind and each task in the procedure should potentially be executable.

Definition 12 (Soundness). A workflow net $N = (P, T, F, i, o)$ is *sound* if and only if it satisfies the following requirements:

- option to complete, i.e., $\forall M \in [N, [i]\rangle \colon [o] \in [N, M\rangle$,
- proper completion, i.e., $\forall M \in [N, [i]\rangle \colon M \geq [o] \implies M = [o]$, and
- no dead transitions, i.e., $\forall t \in T \colon \exists M \in [N, [i]\rangle \colon M[t\rangle$.

The requirement of having no dead transitions is the same as quasi-liveness, and in this definition, the option to complete also already implies proper completion. Although this is easy to see, we provide the corresponding proof which shows that there are no dominating markings in workflow nets with the option to complete.

Theorem 1. *Let $N = (P, T, F, i, o)$ be a workflow net with the option to complete, i.e., $\forall M \in [N, [i]\rangle \colon [o] \in [N, M\rangle$. There is no reachable marking dominating another reachable marking, i.e., $\forall M, M' \in [N, [i]\rangle \colon M \leq M' \implies M = M'$.*

Proof. Let $N = (P, T, F, i, o)$ be a workflow net with the option to complete. Now, let us assume that we have two reachable markings $M, M' \in [N, [i]\rangle$ with $M \leq M'$ and $M \neq M'$. Because of the option to complete, a firing sequence σ with $(N, M)[\sigma\rangle(N, [o])$ exists. It is easy to see that σ is also enabled in M', i.e., $(N, M')[\sigma\rangle$, but firing results in a marking dominating the final marking $[o]$, i.e., $(N, M')[\sigma\rangle(N, [o] + M' - M)$. Again, because of the option to complete and the fact that $o\bullet = \emptyset$, there must exist a firing sequence σ' removing all tokens in $M' - M$ from the net, i.e., $(N, M' - M)[\sigma'\rangle(N, [\,])$. This, however, is impossible because every transition has at least one output place and therefore some token always remains. Hence, $\forall M, M' \in [N, [i]\rangle \colon M \leq M' \implies M = M'$. $\square$

Corollary 1. *A workflow net with the option to complete has proper completion.*

It is a classic result from Petri net theory that reaching a dominating marking is equivalent to unboundedness, hence we further get:

Corollary 2. *A workflow net with the option to complete is bounded.*

The strongly-connected Petri net that we obtain by adding the transition $\bar{t}$ in Definition 11 which connects the sink place to the source place is now commonly referred to as the *short-circuited net*.

Definition 13 (Short-Circuited Net). Let $N = (P, T, F, i, o)$ be a workflow net. Its *short-circuited net* $\bar{N}$ is obtained by adding a transition $\bar{t} \notin T$ with $\bullet\bar{t} = \{o\}$ and $\bar{t}\bullet = \{i\}$, i.e., $\bar{N} := (P, T \cup \{\bar{t}\}, F \cup \{(o, \bar{t}), (\bar{t}, i)\})$. The system $(\bar{N}, [i])$ defines the *short-circuited system*.

Wil used the short-circuited net as an important tool to transfer well-known behavioural characteristics of Petri nets to workflow nets.

Theorem 2 ([3, Theorem 11])**.** *A workflow net N is sound if and only if the short-circuited system $(\bar{N}, [i])$ is live and bounded.*

Verifying soundness is equivalent to verifying whether the short-circuited system is live and bounded [3,4]. For safe workflow nets, this is equivalent to checking liveness of the short-circuited net and can be shown to be PSPACE-complete using a construction similar to the one in Theorem 6 based on [14,15]. Without safeness, deciding soundness on workflow nets has recently been shown to be EXPSPACE-complete [11]. Wil also co-authored a study on the complexity of deciding other soundness notions [10], which will not be elaborated further here.

We can also use the short-circuited system to draw a connection between the option to complete and cyclicity.

Lemma 1. *The short-circuited system of a workflow net with the option to complete is cyclic.*

Proof. Let $N = (P, T, F, i, o)$ be a workflow net. Adding a transition $\bar{t}$ with $\bullet\bar{t} = \{o\}$ and $\bar{t}\bullet = \{i\}$ to obtain the short-circuited system $(\bar{N}, [i])$ does not restrict any behaviour; thus, $[\bar{N}, [i]\rangle \supseteq [N, [i]\rangle$. Because of the option to complete, there is exactly one reachable marking enabling $\bar{t}$, namely $[o]$. Since $(\bar{N}, [o])[\bar{t}\rangle(\bar{N}, [i])$ and $[i] \in [N, [i]\rangle$, $[N, [i]\rangle = [\bar{N}, [i]\rangle$ follows. Hence, $(\bar{N}, [i])$ is cyclic. □

But this shows that the general independence of boundedness, liveness, and cyclicity in Petri nets does not hold up in workflow nets.

Corollary 3. *A live and bounded short-circuited system is cyclic.*

Later on, Wil refined the structural and behavioural implications of soundness by introducing the notion of *perpetual* systems [6], i.e., systems that are live, bounded, and have a *home cluster.*

Definition 14 (Home Cluster). Let (N, M_0) with $N = (P, T, F)$ be a system. A cluster $C \subseteq P \cup T$ of N is a *home cluster* of (N, M_0) if $[p|p \in C \cap P]$ is a home marking.

Obviously, the option to complete already implies the existence of a home cluster. While a home cluster itself does not imply boundedness, the option to complete does (cf. Theorem 1). Perpetual systems are considered a generalisation of sound workflow nets as their short-circuited systems clearly are perpetual [6, Lemma 2].

Wil additionally provided soundness-preserving transformation and composition rules for workflow nets [3,5,9]. In fact, every transformation that preserves liveness and boundedness would suffice. Hence, also well-known Petri net transformations (e.g., as shown in [22, p. 553]) are applicable to sound workflow nets.

4 Sound Free-Choice Workflow Nets

Early on, Wil recognised that the free-choice property has important implications in terms of soundness. With the free-choice property in the picture, he was able to use the Rank Theorem [13,17] to show that soundness can be verified efficiently.

Theorem 3 ([3, Theorem 12]). *For a free-choice workflow net, soundness can be decided in polynomial time.*

He further showed that the option to complete alone suffices as a criterion for soundness because it already implies quasi-liveness:

Proposition 1 ([3, Proposition 13]). *A free-choice workflow net with the option to complete is quasi-live.*

Corollary 4. *A free-choice workflow net with the option to complete is sound.*

But this is not the only relevant implication because another important result by Wil shows that free-choice workflow nets with the option to complete are always safe:

Lemma 2 **([5, Lemma 1]).** *A sound free-choice workflow net is safe.*

Therefore, it is not surprising that the reachability problem can be solved efficiently on this class, and although it would suffice to use Wil's results above and directly transfer the results of Desel and Esparza [18,19] on cyclic live and bounded free-choice systems, we take a small detour and investigate the reachability problem on another class, namely *proper free-choice systems with a home cluster* as it allows us to highlight some of Wil's more recent results.

Here, we briefly discuss classes with the structural property of *lucency* [6–9], meaning that each state is uniquely determined by its set of enabled transitions. Wil not only proved that *perpetual* free-choice nets are lucent [6], but the larger class of proper free-choice systems with home cluster in general [8]. This class is a strict generalisation of perpetual free-choice systems and it immediately applies to free-choice workflow nets which are not necessarily live, but guarantee proper completion of each process instance. Finally, we show that reachability can be solved efficiently for proper free-choice systems with a home cluster:

Theorem 4. *The reachability problem for proper free-choice systems with a home cluster is in* PTIME.

Proof. Let (N, M_0) with $N = (P, T, F)$ be a proper free-choice system with a home cluster $C \subseteq P \cup T$. We use the short-circuited cleaned net polynomial-time construction from [8, Definition 6.4] with the added transition t_C to obtain a cyclic live and bounded free-choice system for which reachability can be decided in polynomial time [19]. A small technicality remains: A firing sequence which starts in the initial marking M_0 and reaches a target marking M might make use of the added transition t_C which is not part of the original net. However, because t_C reinitialises M_0, we can simply take the suffix of the firing sequence after t_C's last occurrence to obtain a firing sequence in the original net. $\square$

Corollary 5. *Reachability for sound free-choice workflow nets is in* PTIME.

Dropping quasi-liveness alone as a requirement on free-choice workflow nets does not affect the complexity result for the reachability problem as the option to complete directly implies quasi-liveness (Proposition 1). Therefore, it remains to study classes of workflow nets without the free-choice property (Sect. 5) and without the option to complete (Sect. 6).

5 Classes Without the Free-Choice Property

The technically most challenging part concerns subclasses with missing free-choice property, i.e., sound workflow nets. We are going to show two things: first of all, as an upper bound, we show that the reachability problem is in EXPSPACE on the subclass of workflow nets with the option to complete. However, it remains open whether EXPSPACE also constitutes a lower bound for sound workflow nets. Instead, we are going to establish the weaker lower bound of PSPACE-hardness which positions the reachability problem for sound workflow nets somewhere in between of PSPACE and EXPSPACE.

5.1 Upper Bound for Workflow Nets with Option to Complete

The crucial observation is that workflow nets with the option to complete are cyclic and bounded. We can then apply the following result due to Bouziane and Finkel [12] which establishes an exponential bound on the representation size of reachable markings in bounded cyclic Petri nets.

Lemma 3 ([12, Lemma 4.1]). *Let (N, M_0) be a bounded cyclic Petri net of size$(N, M_0) = n$. Then, for every reachable marking $M \in [N, M_0\rangle$, we have size$(M) \leq 2^{cn \log n}$ for some fixed constant $c > 0$.*

This implies that all reachable markings in workflow nets with the option to complete can be represented in exponential space. This yields a simple non-deterministic EXPSPACE algorithm for the reachability problem on this class: just guess a firing sequence and keep track of the current marking. This is another great example of how structure-theoretic properties of Petri nets can be exploited for algorithmic purposes. Since by Savitch's Theorem EXPSPACE = NEXPSPACE, it follows:

Theorem 5 ([12, Theorem 4.4]). *The reachability problem for cyclic bounded systems is in* EXPSPACE.

Actually the proof in [12] is more general and does not require boundedness, but for our purposes the above formulation suffices. Because (the short-circuited version of) each workflow net with the option to complete is cyclic and bounded, we can conclude:

Corollary 6. *The reachability problem for workflow nets with the option to complete is in* EXPSPACE.

5.2 Lower Bound for Sound Workflow Nets

We now turn our attention towards the PSPACE-lower bound. To this end we adapt a construction due to Wil and the authors which we recently used to establish PSPACE-hardness of the alignment problem for sound workflow nets [27]. It turns out that only minor changes are necessary, so we just sketch the main ideas and refer to the original paper for full details. The proof is based on simulating a PSPACE-Turing machine in terms of a sound workflow net. Depending on whether the Turing machine accepts or rejects, we can reach a designated accepting or rejecting marking respectively which proves PSPACE-hardness of the reachability problem on safe and sound workflow nets.

Theorem 6. *There is a polynomial-time algorithm which transforms a deterministic Turing machine $\mathcal{M}$ with polynomial space bound $p(n)$ and an input w into a safe and sound workflow net $N = (P, T, F, i, o)$ together with a marking M_{acc} such that $\mathcal{M}$ accepts w if and only if $M_{acc} \in [N, [i]\rangle$.*

Proof. We here only sketch a proof that adapts the construction in [27, Theorem 8] which itself makes strong use of the ideas in [14, Theorem 4].

We assume the same Turing machine $\mathcal{M} = (K, \Sigma, \Gamma, \delta, q_0, q_+, q_-, \perp)$ as in [27, Theorem 8] with the same preprocessing to ensure the polynomial space bound, and exactly one accepting and one rejecting state. To encode the computation of $\mathcal{M}$ on input w, we define a workflow net $N = (P, T, F, i, o)$ where the set of places P consists of i, o, and places to represent each possible state (K), each head position ($\{0, \ldots, p(n)\}$), and each possible tape cell content $\{0, \ldots, p(n)\} \times \Gamma$, i.e., for each combination of a valid position and tape symbol. We construct the workflow net N in such a way that reachable markings can be identified with configurations of $\mathcal{M}$ in the above sense. Thus the set of transitions T consists of one initialising transition, one transition completing the computation when the (unique) accepting configuration is reached, one transition completing the computation when the (unique) rejecting configuration is reached, and a distinct transition for each possible local configuration $(q, a, j) \in K \setminus \{q_+, q_-\} \times \Gamma \times \{0, \ldots, p(n)\}$ that simulates the transition function δ.

Because $\mathcal{M}$ is deterministic and the computation is acyclic, at each local configuration precisely one transition representing δ can fire and we eventually reach the unique accepting or rejecting configuration from which we can fire one of the completing transitions to end in the final marking $[o]$.

Finally, to ensure quasi-liveness of the workflow net N, additional transitions are added that can activate and deactivate each transition from above (by producing (consuming) exactly the required input (provided output) tokens). Thereby, additional places indicate whether the net operates in this auxiliary or in the standard mode. The above transitions therefore initialise and clear these places correspondingly.

The construction can be carried out in polynomial time, and the resulting workflow net N is safe, has the option to complete, and is quasi-live. Asking whether the marking for the accepting configuration can be reached from $[i]$ is identical to asking whether $\mathcal{M}$ accepts w. $\square$

6 Classes Without the Option to Complete

If we drop the option to complete, then the remaining restrictions of being a quasi-live and free-choice workflow net do not provide any algorithmically useful structure. In fact, we are going to show that the reachability problem of general Petri nets can be reduced to the reachability problem of quasi-live free-choice workflow nets in polynomial time. Recall that the reachability problem for general Petri nets is decidable but Ackermann-complete [16,21]. This clearly rules out any hope for efficient algorithms on this class and highlights the crucial role which the option to complete plays for process mining applications.

Lemma 4. *There is a polynomial time algorithm which reduces the reachability problem for general Petri nets to the reachability problem for quasi-live free-choice workflow nets.*

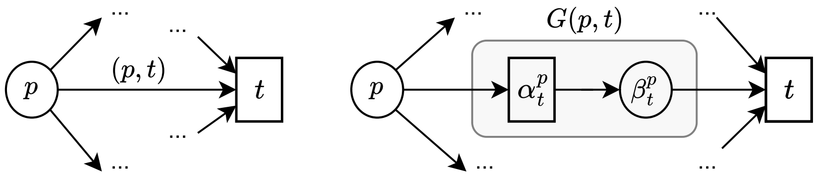

Fig. 1. Gadget $G(p, t)$ that replaces each flow relation $(p, t) \in F \cap (P \times T)$. The gadgets $G(p, t)$ separate choices and synchronisations explicitly.

Proof. Let N be a general Petri net with $N = (P, T, F)$ and two markings $M_1, M_2 \in \mathbb{N}^P$. We can assume that N is connected (otherwise, we can treat the components separately).

As a first step, we transform N into a quasi-live workflow net. Therefore, we add two new places i and o. Next, for each transition $t \in T$ we add two auxiliary transitions t_+ and t_- with $\bullet t_+ = \{i\}$ and $t_+\bullet = \bullet t$, and $\bullet t_- = t\bullet$ and $t_-\bullet = \{o\}$, respectively. This guarantees that every transition $t \in T$ can be fired from the initial marking $[i]$. Moreover, it also guarantees that each vertex is on a path from i to o. Note that if there are other sink or source places (besides i and o) in N we can simply add a loop to make them non-sink or non-source. Hence, the resulting net N' is a quasi-live workflow net. Clearly, this first step of the reduction can be carried out in polynomial time.

It remains to transform N' into a free-choice net (while maintaining quasi-liveness and the worflow net structure). However, since we have no (strong) liveness requirement (due to the absence of the option to complete), this is quite easy. We simply substitute each flow relation $(p, t) \in F \cap (P \times T)$ by a small gadget $G(p, t)$ which is illustrated in Fig. 1.

This construction guarantees that synchronisations only occur for places that have precisely one output transition. The corresponding choices have to happen before and are separated from the synchronisations by the gadgets $G(p, t)$. This yields a free-choice net. If we call the resulting net N'', then it is clear that every firing sequence in N' can be extended to a firing sequence in N'' (by pushing the required tokens through the gadgets $G(p, t)$) and that, vice versa, every firing sequence in N'' can be projected down to a firing sequence in N' (by ignoring the transitions and places in the gadgets). In particular, the net N'' remains quasi-live (but the gadgets $G(p, t)$ might introduce new deadlocks). Hence, M_2 is reachable from M_1 in N if and only if M_2 is reachable from M_1 in N''. The construction can be carried out in polynomial time. $\square$

Due to Lemma 4 and the Ackermann-completeness of the reachability problem for general Petri nets [16,21], we can conclude:

Corollary 7. *The reachability problem for quasi-live free-choice workflow nets is not primitive recursive but* Ackermann*-complete.*

It is easy to see that the construction of Lemma 4 preserves safeness. Due to the PSPACE-completeness of the reachability problem for general safe Petri nets [14,15], we can conclude:

Corollary 8. *The reachability problem for safe quasi-live free-choice workflow nets is* PSPACE*-complete.*

7 Beyond Reachability

We demonstrated that structural properties of workflow nets have a huge impact on the algorithmic complexity of the reachability problem. While for some classes standard linear-algebraic techniques combined with simple graph algorithms are sufficient, other classes touch the very limits of decidability and are way beyond any practical applicability. This highlights the great value of Petri net theory in general, and, in particular, of the theory of workflow nets developed by Wil in several seminal and highly influential works starting from [3–5]. As demonstrated, these results do not only provide clear orientation in the jungle of process models, but may even provide algorithmic pathways for seemingly intractable problems in practice.

Of course, reachability is only one of many relevant algorithmic problems in process mining. Along very similar lines, Wil and the authors recently classified the complexity of the alignment problem for several classes of workflow nets [24–27]. The results have a similar spirit in the sense that the FC and OC properties are key for algorithmic tractability. However, some details diverge from the case of the reachability problem as the alignment problem is generally harder to solve. Consequently, the present results immediately serve as a lower bound for the alignment problem.

These examples showcase how process mining algorithms can benefit from Petri net theory and, in particular, from Wil's extensive work on workflow nets. We believe that the potential here is far from being exhausted. First, there are many more algorithmic problems in process mining which could be studied from this perspective. For instance, the algorithmic complexity of model synthesis or model repair problems are completely open, yet these are fundamental problems in process mining. The same holds for model comparison problems, for example. Second, while free-choice is probably the most studied structural property which leads to efficient algorithms, there are many more patterns that we witness in real-life process models and which could be exploited for algorithmic purposes. One example is to add acyclicity: in many processes, actions are never repeated, and while of course still in PTIME, it allows for faster algorithms such as in [23] which decides reachability in quadratic time. Another example is Wil's notion of lucency: again, in many realistic processes, being in different states also implies being able to perform different actions. So one interesting question is the following: what happens to the algorithmic complexity of reachability (and other problems in process mining) if we consider other structural properties or replace the free-choice property by lucency?

This is not the end of the story! We should not forget that Wil's pioneering work on process mining was once rooted in Petri net theory. The huge success of our research field is also due to its beautiful theoretic foundation. Let us continue to foster the fruitful grounds out of which Wil once established a world full of notions that kept us busy and excited for already three decades. And more importantly, let us use our free choice, will and Wil to apply Petri net theory for solving real world problems!

References

1. van der Aalst, W.M.P.: A class of Petri nets for modeling and analyzing business processes. Computing science reports 95/26. Technische Universiteit Eindhoven (1995). https://research.tue.nl/en/publications/76d57856-828b-49cd-ba84-7bfcd7042ee9
2. van der Aalst, W.M.P.: Structural characterizations of sound workflow nets. Computing science reports 96/23. Technische Universiteit Eindhoven (1996). https://research.tue.nl/en/publications/d7ac08f3-457f-449a-97bb-f6dc54cea668
3. van der Aalst, W.M.P.: Verification of workflow nets. In: Azéma, P., Balbo, G. (eds.) ICATPN 1997. LNCS, vol. 1248, pp. 407–426. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63139-9_48
4. van der Aalst, W.M.P.: The application of petri nets to workflow management. J. Circ. Syst. Comput. **8**(1), 21–66 (1998). https://doi.org/10.1142/s0218126698000043
5. van der Aalst, W.M.P.: Workflow verification: finding control-flow errors using petri-net-based techniques. In: van der Aalst, W., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 161–183. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45594-9_11
6. Aalst, W.M.P.: Markings in perpetual free-choice nets are fully characterized by their enabled transitions. In: Khomenko, V., Roux, O.H. (eds.) PETRI NETS 2018. LNCS, vol. 10877, pp. 315–336. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91268-4_16
7. van der Aalst, W.M.P.: Lucent process models and translucent event logs. Fund. Inform. **169**(1–2), 151–177 (2019). https://doi.org/10.3233/FI-2019-1842
8. van der Aalst, W.M.P.: Free-choice nets with home clusters are lucent. Fund. Inform. **181**(4), 273–302 (2021). https://doi.org/10.3233/FI-2021-2059
9. van der Aalst, W.M.P.: Reduction using induced subnets to systematically prove properties for free-choice nets. In: Buchs, D., Carmona, J. (eds.) PETRI NETS 2021. LNCS, vol. 12734, pp. 208–229. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-76983-3_11
10. van der Aalst, W.M.P., et al.: 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
11. Blondin, M., Mazowiecki, F., Offtermatt, P.: The complexity of soundness in workflow nets. In: Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science. LICS 2022. Association for Computing Machinery, New York (2022). https://doi.org/10.1145/3531130.3533341
12. Bouziane, Z., Finkel, A.: Cyclic petri net reachability sets are semilinear effectively constructible. Electron. Notes Theor. Comput. Sci. **9**, 15–24 (1997). https://doi.org/10.1016/S1571-0661(05)80423-2
13. Campos, J., Chiola, G., Silva, M.: Properties and performance bounds for closed free choice synchronized monoclass queueing networks. IEEE Trans. Automatic Control **36**(12), 1368–1382 (1991). https://doi.org/10.1109/9.106153

14. Cheng, A., Esparza, J., Palsberg, J.: Complexity results for 1-safe nets. In: Shyamasundar, R.K. (ed.) FSTTCS 1993. LNCS, vol. 761, pp. 326–337. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57529-4_66
15. Cheng, A., Esparza, J., Palsberg, J.: Complexity results for 1-safe nets. Theoret. Comput. Sci. **147**(1–2), 117–136 (1995). https://doi.org/10.1016/0304-3975(94)00231-7
16. Czerwiński, W., Orlikowski, Ł.: Reachability in vector addition systems is Ackermann-complete. In: 62nd Annual Symposium on Foundations of Computer Science. FOCS 2021, pp. 1229–1240. IEEE (2022). https://doi.org/10.1109/FOCS52979.2021.00120
17. Desel, J.: A proof of the rank theorem for extended free choice nets. In: Jensen, K. (ed.) ICATPN 1992. LNCS, vol. 616, pp. 134–153. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-55676-1_8
18. Desel, J., Esparza, J.: Reachability in reversible free choice systems. In: Choffrut, C., Jantzen, M. (eds.) STACS 1991. LNCS, vol. 480, pp. 384–397. Springer, Heidelberg (1991). https://doi.org/10.1007/BFb0020814
19. Desel, J., Esparza, J.: Reachability in cyclic extended free-choice systems. Theoret. Comput. Sci. **114**(1), 93–118 (1993). https://doi.org/10.1016/0304-3975(93)90154-L
20. Gaujal, B., Haar, S., Mairesse, J.: Blocking a transition in a free choice net and what it tells about its throughput. J. Comput. Syst. Sci. **66**(3), 515–548 (2003). https://doi.org/10.1016/S0022-0000(03)00039-4
21. Leroux, J.: The reachability problem for petri nets is not primitive recursive. In: 62nd Annual Symposium on Foundations of Computer Science. FOCS 2021, pp. 1241–1252. IEEE (2022). https://doi.org/10.1109/FOCS52979.2021.00121
22. Murata, T.: Petri nets: properties, analysis and applications. Proc. IEEE **77**(4), 541–580 (1989). https://doi.org/10.1109/5.24143
23. Prinz, T.M., Schwanen, C.T., van der Aalst, W.M.P.: Deciding (sub-marking) reachability in O(P2 + T2) for sound acyclic free-choice workflow nets. In: Application and Theory of Petri Nets and Concurrency. Petri Nets 2025. LNCS, vol. 15714, pp. 366–387. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-94634-9_18
24. Schwanen, C.T., Pakusa, W., van der Aalst, W.M.P.: A dynamic programming approach for alignments on process trees. In: Process Mining Workshops. ICPM 2024 International Workshops, Lyngby, Denmark, 14–18 October 2024, Revised Selected Papers. ICPM 2024. LNBIP, vol. 533, pp. 84–97. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-82225-4_7
25. Schwanen, C.T., Pakusa, W., van der Aalst, W.M.P.: Alignments meet linear algebra. In: Algorithms & Theories for the Analysis of Event Data (ATAED 2025), vol. 3998, pp. 185–200. CEUR Workshop Proceedings. Aachen: CEURWS. org (2025). https://ceur-ws.org/Vol-3998/paper12.pdf
26. Schwanen, C.T., Pakusa, W., van der Aalst, W.M.P.: Process tree alignments. In: Enterprise Design, Operations, and Computing. EDOC 2024. LNCS, vol. 15409, pp. 300–317. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-78338-8_16
27. Schwanen, C.T., Pakusa, W., van der Aalst, W.M.P.: Complexity of alignments on sound free-choice workflow nets. In: Application and Theory of Petri Nets and Concurrency. Petri Nets 2025. LNCS, vol. 15714, pp. 388–410. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-94634-9_19