The complexity of soundness in workflow nets

Michael Blondin
Université de Sherbrooke
Sherbrooke, Canada
michael.blondin@usherbrooke.ca

Filip Mazowiecki
Max Planck Institute for Software Systems
Saarbrücken, Germany
filipm@mpi-sws.org

Philip Offtermatt
Max Planck Institute for Software Systems
Saarbrücken, Germany
Université de Sherbrooke
Sherbrooke, Canada
philip.offtermatt@usherbrooke.ca

Abstract

Workflow nets are a popular variant of Petri nets that allow for the algorithmic formal analysis of business processes. The central decision problems concerning workflow nets deal with soundness, where the initial and final configurations are specified. Intuitively, soundness states that from every reachable configuration one can reach the final configuration. We settle the widely open complexity of the three main variants of soundness: classical, structural and generalised soundness. The first two are EXPSPACE-complete, and, surprisingly, the latter is PSPACE-complete, thus computationally simpler.

CCS Concepts • **Software and its engineering → Petri nets**; • **Theory of computation** → Computational complexity and cryptography;

Keywords Workflow nets, Petri nets, soundness, generalised soundness, structural soundness, complexity

ACM Reference Format:
Michael Blondin, Filip Mazowiecki, and Philip Offtermatt. 2022. The complexity of soundness in workflow nets. In *37th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (LICS '22), August 2–5, 2022, Haifa, Israel,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 39, 14 pages. https://doi.org/10.1145/3531130.3533341

1 Introduction

Workflow nets are a formalism that allows for the modeling of business processes. Specifically, they allow to formally represent workflow procedures in Workflow Management Systems (WFMSs) (see *e.g.* [23, Section 4], where Figure 6 shows a workflow net for the processing of complaints; and [22, Section 3] for details on modeling procedures). Such a mathematical representation enables the algorithmic formal analysis of their behaviour. This is particularly relevant for large organisations that seek to manage the workflow of complex business processes. Such challenges have received, and continue to receive, intense academic attention, *e.g.* through the foundations track of the Business Process Management Conference (BPM), and via a discipline coined as *process mining* and pioneered prolifically by Wil van der Aalst.[1] In particular, many tools, such as those integrated in the ProM framework [26], can extract events from logs, *e.g.* of enterprise resource planning (ERP) systems, from which they synthesize workflow nets (and other models) to be formally analyzed (see [24] for a book on the topic).

More formally, workflow nets form a subset of (standard) Petri nets. They consist of *places* that can contain resources (called *tokens*) which can be consumed and produced via *transitions* in a nondeterministic and concurrent fashion. Two designated places, namely the *initial place* i and the *final place* f, respectively model the initialisation and termination of a business process. No token can be produced in the initial place, and no token can be consumed from the final place.

A central property studied since the inception of workflow nets is 1-*soundness* [22, 23]. Informally, quoting [22], it states that "*For any case, the procedure will terminate eventually [...]*". More formally, from the configuration with a single token in the initial place i, every reachable configuration can reach the configuration with a single token in the final place f. For readers familiar with computation temporal logic (CTL), 1-soundness can be loosely rephrased as $\mathsf{i} \models \forall \mathsf{G}\, \exists \mathsf{F}\, \mathsf{f}$. More generally, k-soundness states the same but for k tokens, *i.e.* $(k \cdot \mathsf{i}) \models \forall \mathsf{G}\, \exists \mathsf{F}\, (k \cdot \mathsf{f})$.

Classical soundness. Several variants of soundness have been considered in the literature (see [25] for a survey). The best-known is *classical soundness*. It states that a workflow net is 1-sound and that each transition is meaningful, *i.e.* each transition can be fired in at least one execution (often called *quasi-liveness*). It is well-known that deciding classical soundness amounts to checking boundedness and liveness of a slightly modified net. In particular, this means that classical soundness is decidable since boundedness and liveness are decidable problems. However, to the best of our knowledge, the (exact) complexity of classical soundness remains widely open. It has been suggested that classical

LICS '22, August 2–5, 2022, Haifa, Israel

ACM ISBN 978-1-4503-9351-5/22/08.
https://doi.org/10.1145/3531130.3533341

[1] See http://www.processmining.org.

soundness is EXPSPACE-hard. For example, the author of [9] mentions that "*IO-soundness is decidable but also EXPSPACE-hard ([21])*", yet [21] merely states the following:

> *[...] [I]t may be intractable to decide soundness. (For arbitrary [workflow]-nets liveness and boundedness are decidable but also EXPSPACE-hard [...]).*

Further, [23, p. 38] suggests that intractability follows from the fact that "*deciding liveness and boundedness is EXPSPACE-hard*", which is attributed to [5]. However, the latter only mentions liveness to be EXPSPACE-hard (which was known prior to [5]).

The confusion arises from the fact that boundedness and liveness are *independently* EXPSPACE-hard problems, which suggests that classical soundness must naturally be at least as hard. However, this needs not be the case. For example, for a well-studied subclass of Petri nets, called free-choice nets, testing *simultaneously* boundedness and liveness has lower complexity than testing both properties independently[2] [10]. Moreover, since liveness is equivalent to the Petri net reachability problem [13], the only known upper bound is not even primitive recursive [15]. As a first contribution, we show that classical soundness and k-soundness are in fact both EXPSPACE-hard and in EXPSPACE, and hence EXPSPACE-complete. The upper bound is derived with a fortiori surprisingly little effort by invoking known results on coverability and so-called cyclicity. The hardness result is obtained by a careful reduction from the reachability problem for reversible Petri nets [4, 17]. There, we exploit subtle known results in a technically challenging way.

Generalised and structural soundness. Among the variants of soundness catalogued by the survey of van der Aalst et al. [25], *generalised soundness* [28, Def. 3] is the only fundamentally distinct property (in particular, see [25, Fig. 7]). It asks whether a given workflow net is k-sound *for all* $k \geq 1$. Generalised soundness, unlike classical soundness, preserves nice properties like composition [28]. The existential counterpart of generalised soundness, where "for all" is replaced by "for some", is known as *structural soundness* [1].

It is a priori not clear whether generalised and structural soundness are decidable, as the approach for deciding other types of soundness reasons about k-soundness for a given or fixed number k. Nonetheless, both problems have been shown decidable [7, 29]. The two algorithms, and a subsequent one [27], rely on Petri net reachability, which has very recently been shown Ackermann-complete [8, 14, 15].

As for classical soundness, the computational complexity of generalised and structural soundness remains open. In fact, we are not aware of any complexity result. In this work, we prove that generalised and structural soundness have much lower complexity than Petri net reachability: they are respectively PSPACE-complete and EXPSPACE-complete. In particular, the fact that generalised soundness is simpler than classical soundness is arguably surprising: positive instances of both problems require the given workflow net to be bounded, but for generalised soundness, one can avoid explicitly checking this EXPSPACE-complete property.

To derive the PSPACE membership, we introduce the notion of *strong soundness* which is defined in terms of a relaxed reachability relation (sometimes known as $\mathbb{Z}$-*reachability* or *pseudo-reachability*, *e.g.*, see [2]). Through results on integer linear programming and exploiting Steinitz Lemma on reordering vectors [20], we prove that k-unsoundness of a workflow net must occur for a "small" number k. Furthermore, we show that it suffices to witness such a k for so-called $\mathbb{Z}$-bounded nonredundant nets, with $\mathbb{Z}$-boundedness being a more restrictive property than (standard) boundedness. By building upon these results, we establish the EXPSPACE membership of structural soundness, and, in fact, effectively characterise the set of sound numbers of workflow nets, which settles the open problem of [7].

The hardness results for PSPACE and EXPSPACE are respectively obtained via reductions from the reachability problem for conservative Petri nets [18], and from 1-soundness.

Contribution and organisation. In summary, we settle, after around two decades, the exact computational complexity of the central decision problems for workflow nets. This is achieved in the rest of this work, organised as follows. In Section 2, we introduce general notation, Petri nets, workflow nets and soundness. In Section 3, we prove that classical soundness is EXPSPACE-complete. In Section 4, we provide bounds on vector reachability, which in turn allows us to prove PSPACE-completeness of generalised soundness (Section 5), and EXPSPACE-completeness of structural soundness (Section 6). In Section 7, we leverage the previous results to give a characterisation of numbers k for which a workflow net is k-sound. Finally, we conclude in Section 8. Due to space constraints, some proofs are deferred to an appendix.

2 Preliminaries

We denote naturals and integers with the usual font: $n \in \mathbb{N}$ and $z \in \mathbb{Z}$. Given $i, j \in \mathbb{Z}$, we write $[i..j]$ for $\{i, i+1, \ldots, j\}$. We use the bold font for vectors and matrices, *e.g.* $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathbb{Z}^n$ and $\mathbf{A} \in \mathbb{Z}^{m \times n}$. Given $n \in \mathbb{N}$, we write $\boldsymbol{n}^d = (n, \ldots, n) \in \mathbb{N}^d$. We omit the dimension d when it is clear from the context, *e.g.* $\mathbf{0}$ denotes the null vector. We write $\boldsymbol{a}[i] = a_i$ and $\mathbf{A}[i, j]$ for matrix entries where $i \in [1..m]$ and $j \in [1..n]$. We write $\boldsymbol{x} \leq \boldsymbol{y}$ if $\boldsymbol{x}[i] \leq \boldsymbol{y}[i]$ holds for all $i \in [1..n]$. We write $\boldsymbol{x} < \boldsymbol{y}$ if at least one inequality is strict. Given a vector $\boldsymbol{a} \in \mathbb{Z}^n$ or a matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$, we define the norms $\|\boldsymbol{a}\| := \max_{1 \leq i \leq n} |\boldsymbol{a}[i]|$ and $\|\mathbf{A}\| := \max_{1 \leq j \leq m, 1 \leq i \leq n} |\mathbf{A}[j, i]|$.

[2] For free-choice nets: Boundedness is EXPSPACE-complete since any Petri net can trivially be made free-choice while preserving its reachability set up to projection; liveness is coNP-complete [10, Thm. 4.28]; and testing liveness *and* boundedness can be done in polynomial time [10, Cor. 6.18].

2.1 Petri nets

A *Petri net* is a triple $\mathcal{N} = (P, T, F)$ such that:

- P and T are disjoint finite sets whose elements are respectively called *places* and *transitions*,
- $F \colon ((P \times T) \cup (T \times P)) \to \mathbb{N}$ is the *flow function*.

A *marking* is a vector $\boldsymbol{m} \colon P \to \mathbb{N}$ where $\boldsymbol{m}[p]$ indicates how many *tokens* are contained in place p. Informally, $F[p, t]$ and $F[t, p]$ respectively correspond to the amount of tokens to be consumed from and produced in place p. Let ${}^\bullet t, t^\bullet \in \mathbb{N}^P$ respectively denote the vectors such that ${}^\bullet t[p] \coloneqq F[p, t]$ and $t^\bullet[p] \coloneqq F[t, p]$. Let $\Delta(t) \coloneqq t^\bullet - {}^\bullet t$ denote the *effect* of t. We say that a transition $t \in T$ is *enabled* in $\boldsymbol{m}$ if $\boldsymbol{m} \geq {}^\bullet t$. If t is enabled in $\boldsymbol{m}$, then t may be *fired*, which leads to the marking $\boldsymbol{m}' \coloneqq \boldsymbol{m} + \Delta(t)$. The latter is denoted by $\boldsymbol{m} \rightarrow^t \boldsymbol{m}'$, or simply by $\boldsymbol{m} \rightarrow \boldsymbol{m}'$ whenever we do not care about the transition that led to $\boldsymbol{m}'$. We use a standard notation for markings, listing only nonzero values, *e.g.* if $P = \{p_1, p_2\}$, $\boldsymbol{m}[p_1] = 2$ and $\boldsymbol{m}[p_2] = 0$, then $\boldsymbol{m} = \{p_1 \colon 2\}$.

A *run* is a sequence of transitions $\rho = t_1 \cdots t_n \in T^*$. A run is enabled in a marking $\boldsymbol{m}_0$ if there is a sequence of markings $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_n$ such that $\boldsymbol{m}_i \rightarrow^{t_{i+1}} \boldsymbol{m}_{i+1}$ for all $0 \leq i < n$. If it is the case, then we denote this by $\boldsymbol{m}_0 \rightarrow^\rho \boldsymbol{m}_n$, or $\boldsymbol{m}_0 \rightarrow^* \boldsymbol{m}_n$ if ρ is not important. Given $\ell \in \mathbb{N}$, we say that ρ is *ℓ-bounded* if $\|\boldsymbol{m}_i\| \leq \ell$ for all $0 \leq i \leq n$. The support of a run is the set of transitions occurring in it, denoted $\mathrm{supp}(\rho) \coloneqq \{t_1, \ldots, t_n\}$.

We introduce a semantics where transitions can always be fired, and hence where markings may become negative. Formally, a *$\mathbb{Z}$-marking* is a vector $\boldsymbol{m} \colon P \to \mathbb{Z}$. We write $\boldsymbol{m} \rightarrow^t_{\mathbb{Z}} \boldsymbol{m}'$ (or simply $\boldsymbol{m} \rightarrow_{\mathbb{Z}} \boldsymbol{m}'$) if $\boldsymbol{m}' = \boldsymbol{m} + \Delta(t)$. Given a run ρ, we define in the obvious way $\rightarrow^\rho_{\mathbb{Z}}$ and $\rightarrow^*_{\mathbb{Z}}$. Note that markings are $\mathbb{Z}$-markings (with the domain restricted to $\mathbb{N}$). The definition of $\mathbb{Z}$-markings is mostly needed to use $\rightarrow^*_{\mathbb{Z}}$.

We define the *absolute value* and *norm* of a Petri net $\mathcal{N} = (P, T, F)$ by $|\mathcal{N}| \coloneqq |P| + |T|$ and $\|\mathcal{N}\| \coloneqq \|F\| + 1$, where F is seen as a vector over $(P \times T) \cup (T \times P)$. The *size* of a Petri net is defined as $\mathrm{size}(\mathcal{N}) \coloneqq |\mathcal{N}| \cdot (1 + \log \|\mathcal{N}\|)$. For some complexity problems, we will be given a Petri net and some markings, *e.g.* $\boldsymbol{m}$ and $\boldsymbol{m}'$. By the size of the input, we understand $\mathrm{size}(\mathcal{N}, \boldsymbol{m}, \boldsymbol{m}') \coloneqq \mathrm{size}(\mathcal{N}) + \log(\|\boldsymbol{m}\| + 1) + \log(\|\boldsymbol{m}'\| + 1)$.

A transition t is said to be *quasi-live* from marking $\boldsymbol{m}$ if there exists a marking $\boldsymbol{m}'$ such that $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$ and t is enabled in $\boldsymbol{m}'$. A transition t is said to be *live* from $\boldsymbol{m}$ if t is quasi-live from all $\boldsymbol{m}'$ such that $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$. We say that a Petri net $\mathcal{N}$ is *quasi-live* (resp. *live*) from $\boldsymbol{m}$ if each transition t of $\mathcal{N}$ is quasi-live (resp. live) from $\boldsymbol{m}$. Informally, quasi-liveness states that no transition is useless, and liveness states that transitions can always eventually be fired.

Example 2.1. Consider the Petri net $\mathcal{N}_{\text{middle}} = (P, T, F)$ illustrated in the middle of Figure 1. Places $P = \{\mathsf{i}, q_1, q_2, \mathsf{f}\}$ and transitions $T = \{t_1, t_2, t_3, t_4\}$ are depicted respectively as circles and squares. The flow function F is depicted by arcs, where weight 1 is omitted and arcs with weight 0 are not drawn, *e.g.* $F(\mathsf{i}, t_1) = 1$, $F(t_1, \mathsf{i}) = 0$, $F(t_4, \mathsf{f}) = 2$ and $F(\mathsf{f}, t_4) = 0$. In particular, transitions t_1, t_2 and t_3 are quasi-live from marking $\{\mathsf{i} \colon 1\}$ since

$$\{\mathsf{i} \colon 1\} \rightarrow^{t_1} \{q_1 \colon 1\} \rightarrow^{t_2} \{q_2 \colon 1\} \rightarrow^{t_3} \{q_1 \colon 1\}.$$

However, as no other marking is reachable, transition t_4 is not quasi-live. Note that t_2 and t_3 are both live from $\{\mathsf{i} \colon 1\}$, while t_1 is not live since it can only be fired once.

2.2 Workflow nets and soundness

A *workflow net* $\mathcal{N}$ is a Petri net that satisfies the following:

- there is a dedicated *initial* place i with $t^\bullet[\mathsf{i}] = 0$ for every transition t (cannot produce tokens in i);
- there is a dedicated *final* place $\mathsf{f} \neq \mathsf{i}$ with ${}^\bullet t[\mathsf{f}] = 0$ for every transition t (cannot consume tokens from f);
- each place and transition lies on at least one path from i to f in the underlying graph of $\mathcal{N}$, *i.e.* the graph (V, E) where $V \coloneqq P \cup T$ and $(u, v) \in E$ iff $F[u, v] > 0$.

Given $k \in \mathbb{N}$, we say that $\mathcal{N}$ is k-sound iff for all $\boldsymbol{m}$, $\{\mathsf{i} \colon k\} \rightarrow^* \boldsymbol{m}$ implies $\boldsymbol{m} \rightarrow^* \{\mathsf{f} \colon k\}$, i.e. starting from k tokens in the initial place, it is always possible to move the k tokens into the final place. We say that $\mathcal{N}$ is:

- *classically sound* iff $\mathcal{N}$ is 1-sound and quasi-live from $\{\mathsf{i} : 1\}$;
- *generalised sound* iff $\mathcal{N}$ is k-sound for all $k \geq 0$;
- *structurally sound* iff $\mathcal{N}$ is k-sound for some $k > 0$.

Example 2.2. Consider the workflow nets $\mathcal{N}_{\text{left}}$, $\mathcal{N}_{\text{middle}}$ and $\mathcal{N}_{\text{right}}$ depicted respectively in Figure 1.

Workflow nets $\mathcal{N}_{\text{left}}$ and $\mathcal{N}_{\text{middle}}$ are not 1-sound since their only transition that can mark place f is not quasi-live from $\{\mathsf{i} \colon 1\}$, namely s_2 and t_4. In particular, this means that both workflow nets are neither classically sound, nor generalized sound. Workflow net $\mathcal{N}_{\text{right}}$ is 1-sound, and in fact classically sound, as shown by the reachability graph of Figure 2.

In particular, this means that $\mathcal{N}_{\text{right}}$ is structurally sound. Workflow net $\mathcal{N}_{\text{left}}$ is not structurally sound as no matter the marking $\{\mathsf{i} \colon k\}$ from which it starts, there is no way to empty place p_1 once it is marked. Workflow net $\mathcal{N}_{\text{middle}}$ is 2-sound, and hence structurally sound. Indeed, from $\{\mathsf{i} \colon 2\}$, the two tokens must enter $\{q_1, q_2\}$ from which they can escape via $\{q_1 \colon 1, q_2 \colon 1\}$ by firing t_4, reaching marking $\{\mathsf{f} \colon 2\}$.

Workflow net $\mathcal{N}_{\text{right}}$ is not 2-sound, and hence not generalised sound. Indeed, we have $\{\mathsf{i} \colon 2\} \rightarrow^{u_1 u_2 u_4} \{r_2 \colon 2, \mathsf{f} \colon 1\}$ and no transition is enabled in the latter marking.

To gain some intuition on why soundness in workflow nets is in general easier than reachability in Petri nets, let us end this section by proving a simple property which lets us conclude k-unsoundness from strict coverability of the final marking $\{\mathsf{f} \colon k\}$.

Lemma 2.3. *Let $\mathcal{N} = (P, T, F)$ be a workflow net and let $k \in \mathbb{N}$. If $\{\mathsf{i} \colon k\} \rightarrow^* \{\mathsf{f} \colon k\} + \boldsymbol{m}$ for some marking $\boldsymbol{m} > \boldsymbol{0}$, then $\mathcal{N}$ is not k-sound.*

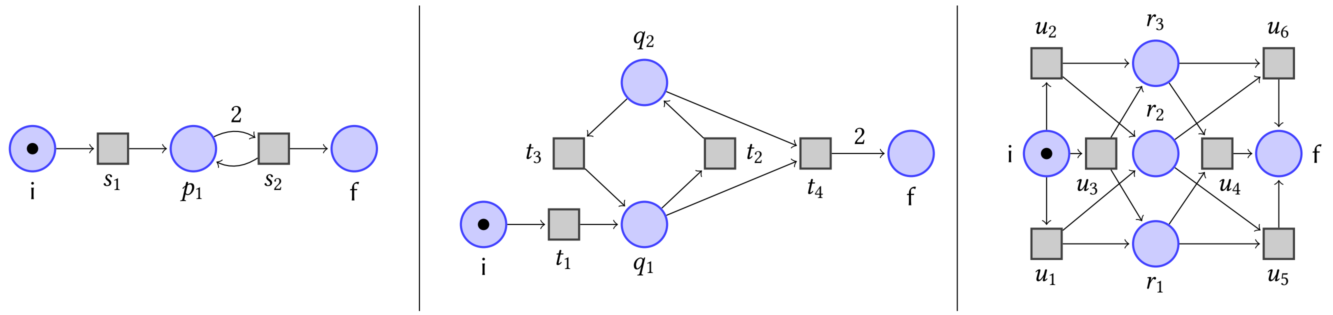

Figure 1. Three workflow nets, each marked with $\{\mathsf{i}\colon 1\}$.

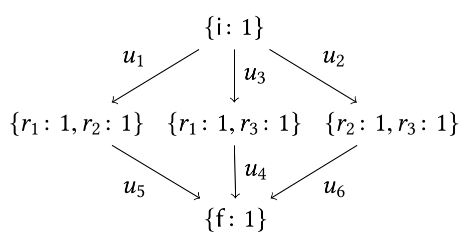

Figure 2. Markings reachable from $\{\mathsf{i}\colon 1\}$ in $\mathcal{N}_{\text{right}}$.

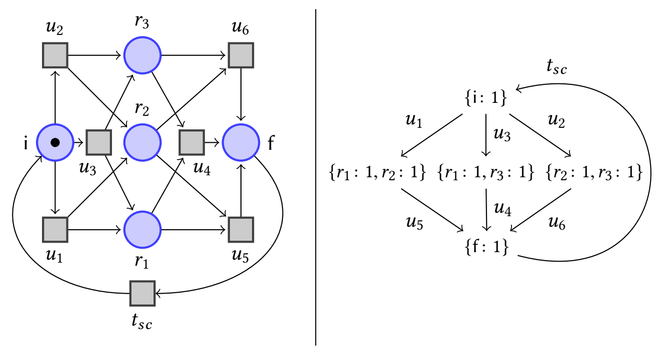

Figure 3. *Left*: Short-circuit net of the rightmost workflow net from Figure 1. *Right*: Its markings reachable from $\{\mathsf{i}\colon 1\}$.

In the proof of Lemma 2.3 we rely on two properties of workflow nets: that there are no outgoing edges from f; and that from a nonzero marking one cannot reach a zero marking (since all nodes are on a path from i to f).

Proof of Lemma 2.3. By definition, each transition lies on a path from i to f. In particular, this means $t^\bullet \neq \mathbf{0}$ for all $t \in T$. So $\{\mathsf{f}\colon k\} + \boldsymbol{m} \not\rightarrow^* \{\mathsf{f}\colon k\}$, and thus $\mathcal{N}$ is not k-sound. □

3 Classical soundness

As mentioned in the introduction, classical soundness is decidable, but its complexity has not yet been established. Let us recall why decidability holds. We say that a Petri net $\mathcal{N}$ is *bounded* from marking $\boldsymbol{m}$ if there exists $b \in \mathbb{N}$ such that $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$ implies $\boldsymbol{m}' \leq \boldsymbol{b}$. Otherwise, $\mathcal{N}$ is *unbounded* from $\boldsymbol{m}$. It is well-known that unboundedness holds iff there exist markings $\boldsymbol{m}' < \boldsymbol{m}''$ such that $\boldsymbol{m} \rightarrow^* \boldsymbol{m}' \rightarrow^* \boldsymbol{m}''$. The *short-circuit net* $\mathcal{N}_{sc}$ of a workflow net $\mathcal{N}$ is $\mathcal{N}$ extended with a transition t_{sc} such that $F[\mathsf{f}, t_{sc}] = F[t_{sc}, \mathsf{i}] = 1$ (and 0 for other entries relating to t_{sc}). Informally, the short-circuit net allows restoring the system upon completion, *i.e.* by moving a token from f to i.

For example, the left side of Figure 3 illustrates a short-circuit net $\mathcal{N}_{sc}$. By inspecting the graph of markings reachable from $\{\mathsf{i}\colon 1\}$ in $\mathcal{N}_{sc}$, we see that $\mathcal{N}_{sc}$ is live and bounded, *i.e.* it is always possible to (re)fire any transition, and each place is bounded by $b := 1$ token. It turns out that liveness and boundedness characterize classical soundness:

Proposition 3.1 ([22, Lemma 8]). *A workflow net $\mathcal{N}$ is classically sound iff $\mathcal{N}_{sc}$ is live and bounded from* $\{\mathsf{i}\colon 1\}$.

Decidability of classical soundness follows from Theorem 3.1. Indeed, boundedness can be tested in EXPSPACE [19], and liveness is decidable since it reduces to reachability [13, Thm 5.1] which is decidable [16]. However, the liveness problem is hard for the reachability problem [13, Thm 5.2], which was recently shown Ackermann-complete [8, 14, 15]. In this section, we first give a slightly different characterization not involving liveness which yields EXPSPACE membership. Then, we show that classical soundness is EXPSPACE-hard, and hence EXPSPACE-complete, via a reduction from the reachability problem for so-called reversible Petri nets.

3.1 EXPSPACE membership

Let us reformulate the characterization of Theorem 3.1 so that it deals with another property than liveness, namely "cyclicity". We say that a Petri net is *cyclic* from a marking $\boldsymbol{m}$ if for any marking $\boldsymbol{m}'$, $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$ implies $\boldsymbol{m}' \rightarrow^* \boldsymbol{m}$, i.e. it is always possible to go back to $\boldsymbol{m}$. For example, the short-circuit net $\mathcal{N}_{sc}$, illustrated on the left of Figure 3, is cyclic since each marking reachable from $\{\mathsf{i}\colon 1\}$ can reach $\{\mathsf{f}\colon 1\}$, which in turn can reach $\{\mathsf{i}\colon 1\}$.

Rather than directly considering classical soundness, we first consider 1-soundness. The characterization of Theorem 3.1 can be adapted to this problem as follows:

Proposition 3.2. *A workflow net $\mathcal{N}$ is 1-sound iff $\mathcal{N}_{sc}$ is bounded and transition t_{sc} is live from* $\{\mathsf{i}\colon 1\}$.

Proof. Let $\mathcal{N} = (P, T, F)$.

$\Rightarrow$) For the sake of contradiction, suppose that $\mathcal{N}_{sc}$ is unbounded. There exist markings $\boldsymbol{m} < \boldsymbol{m}'$ such that $\{\mathsf{i}\colon i\} \rightarrow^{\pi} \boldsymbol{m} \rightarrow^{\pi'} \boldsymbol{m}'$ in $\mathcal{N}_{sc}$. Let us assume, without loss of generality, that no marking repeats along the run. There are two cases to consider: either $\pi\pi'$ contains t_{sc}, or not.

Let us argue that the first case cannot hold. For the sake of contradiction, assume it does. Let σt_{sc} be the shortest prefix of $\pi\pi'$ such that $\{\mathsf{i}\colon 1\} \rightarrow^{\sigma} \{\mathsf{f}\colon 1\} + \boldsymbol{n} \rightarrow^{t_{sc}} \{\mathsf{i}\colon 1\} + \boldsymbol{n}$ in $\mathcal{N}_{sc}$. Note that by minimality, $\{\mathsf{i}\colon 1\} \rightarrow^{\sigma} \{\mathsf{f}\colon 1\} + \boldsymbol{n}$ holds in $\mathcal{N}$. If $\boldsymbol{n} = \boldsymbol{0}$, then we obtain a contradiction as no marking repeats. Otherwise $\mathcal{N}$ is unsound by Theorem 2.3, which contradicts 1-soundness.

Thus, $\pi\pi'$ only contains transitions from T, which means we can reason about $\mathcal{N}$ (rather than $\mathcal{N}_{sc}$). By 1-soundness, we have $\{\mathsf{i}\colon 1\} \rightarrow^{*} \boldsymbol{m} \rightarrow^{*} \{\mathsf{f}\colon 1\}$ in $\mathcal{N}$. Since $\{\mathsf{i}\colon 1\} \rightarrow^{*} \boldsymbol{m}'$ in $\mathcal{N}$, altogether this yields

$$\{\mathsf{i}\colon 1\} \rightarrow^{*} \boldsymbol{m}' = \boldsymbol{m} + (\boldsymbol{m}' - \boldsymbol{m}) \rightarrow^{*} \{\mathsf{f}\colon 1\} + (\boldsymbol{m}' - \boldsymbol{m}).$$

By 1-soundness, this means that $(\boldsymbol{m}' - \boldsymbol{m}) \rightarrow^{*} \boldsymbol{0}$, which is impossible. Consequently, $\mathcal{N}_{sc}$ is bounded from $\{\mathsf{i}\colon 1\}$.

It remains to argue that t_{sc} is live from $\{\mathsf{i}\colon 1\}$. Let $\{\mathsf{i}\colon 1\} \rightarrow^{\rho} \boldsymbol{m}$ in $\mathcal{N}_{sc}$, where no marking repeats. We can assume that t_{sc} does not appear in ρ as it would mean that $\{\mathsf{i}\colon 1\}$ is repeated. Hence, $\{\mathsf{i}\colon 1\} \rightarrow^{\rho} \boldsymbol{m}$ in $\mathcal{N}_{sc}$. By 1-soundness, we have $\boldsymbol{m} \rightarrow^{*} \{\mathsf{f}\colon 1\}$, from which t_{sc} is enabled as desired.

$\Leftarrow$) Let $\{\mathsf{i}\colon 1\} \rightarrow^{*} \boldsymbol{m}$ in $\mathcal{N}$ (and so in $\mathcal{N}_{sc}$). Since t_{sc} is live from $\{\mathsf{i}\colon 1\}$, we have $\boldsymbol{m} \rightarrow^{*} \{\mathsf{f}\colon 1\} + \boldsymbol{n}$ for some $\boldsymbol{n} \in \mathbb{N}^P$. If $\boldsymbol{n} > \boldsymbol{0}$, then we obtain $\{\mathsf{i}\colon 1\} \rightarrow^{*} \{\mathsf{f}\colon 1\} + \boldsymbol{n} \rightarrow^{t_{sc}} \{\mathsf{i}\colon 1\} + \boldsymbol{n}$ which violates boundedness. Thus, $\boldsymbol{n} = \boldsymbol{0}$, and hence $\boldsymbol{m} \rightarrow^{*} \{\mathsf{f}\colon 1\}$ as desired. □

From the previous proposition, we prove the following.

Lemma 3.3. *A workflow net $\mathcal{N} = (P, T, F)$ is 1-sound iff $\mathcal{N}_{sc}$ is bounded and cyclic from $\{\mathsf{i}\colon 1\}$, and some transition $t \in T$ satisfies ${}^\bullet t = \{\mathsf{i}\colon 1\}$.*

Proof. $\Rightarrow$) Let $\mathcal{N}$ be 1-sound. Since $\{\mathsf{i}\colon 1\} \rightarrow^{*} \{\mathsf{f}\colon 1\}$ and $\mathsf{i} \neq \mathsf{f}$, some $t \in T$ satisfies ${}^\bullet t = \{\mathsf{i}\colon 1\}$. By Theorem 3.2, from $\{\mathsf{i}\colon 1\}$, $\mathcal{N}_{sc}$ is bounded and t_{sc} is live. It remains to show that $\mathcal{N}_{sc}$ is cyclic. Let $\{\mathsf{i}\colon 1\} \rightarrow^{*} \boldsymbol{m}$. By liveness of t_{sc}, there is a marking $\boldsymbol{m}'$ such that $\boldsymbol{m} \rightarrow^{*} \boldsymbol{m}'$ and $\boldsymbol{m}'$ enables t_{sc}. Note that ${}^\bullet t_{sc} = \{\mathsf{f}\colon 1\}$. If $\boldsymbol{m}' > \{\mathsf{f}\colon 1\}$, that is, $\boldsymbol{m}' = \{\mathsf{f}\colon 1\} + \boldsymbol{n}$ with $\boldsymbol{n} > \boldsymbol{0}$, then we obtain $\{\mathsf{i}\colon 1\} \rightarrow^{*} \{\mathsf{f}\colon 1\} + \boldsymbol{n} \rightarrow^{t_{sc}} \{\mathsf{i}\colon 1\} + \boldsymbol{n}$, and hence boundedness is violated. Thus, by boundedness and liveness of t_{sc}, $\boldsymbol{m} \rightarrow^{*} \boldsymbol{m}' = \{\mathsf{f}\colon 1\} \rightarrow^{t_{sc}} \{\mathsf{i}\colon 1\}$, which proves cyclicity.

$\Leftarrow$) Assume $\mathcal{N}_{sc}$ is bounded and cyclic from $\{\mathsf{i}\colon 1\}$, and that some $t \in T$ is as described. By Theorem 3.2, it suffices to show that t_{sc} is live from $\{\mathsf{i}\colon 1\}$. Let $\boldsymbol{m} \in \mathbb{N}^P$ be such that $\{\mathsf{i}\colon 1\} \rightarrow^{*} \boldsymbol{m}$ in $\mathcal{N}_{sc}$. We either have $\boldsymbol{m} = \{\mathsf{i}\colon 1\}$ or $\boldsymbol{m}[\mathsf{i}] = 0$, as otherwise $\mathcal{N}_{sc}$ is unbounded. If $\boldsymbol{m} = \{\mathsf{i}\colon 1\}$, we can fire t and obtain a marking where i is empty. Thus, assume w.l.o.g. that $\boldsymbol{m}[\mathsf{i}] = 0$. By cyclicity, we have $\boldsymbol{m} \rightarrow^{\pi} \{\mathsf{i}\colon 1\}$ for some π. Since t_{sc} is the only transition that produces tokens in place i, transition t_{sc} must appear in π. Hence, t_{sc} is live. □

Since classical soundness amounts to quasi-liveness and 1-soundness, we obtain the following corollary.

Corollary 3.4. *A workflow net $\mathcal{N}$ is classically sound iff $\mathcal{N}_{sc}$ is quasi-live, bounded and cyclic from $\{\mathsf{i}\colon 1\}$.*

Theorem 3.5. *Both 1-soundness and classical soundness are in EXPSPACE.*

Proof. Checking whether a transition t satisfies ${}^\bullet t = \{\mathsf{i}\colon 1\}$ can be carried in polynomial time. The other properties of Theorem 3.3 for 1-soundness, namely boundedness and cyclicity, belong to EXPSPACE [3, 19].

For quasi-liveness, we proceed as follows. The *coverability problem* asks whether given a Petri net and two markings $\boldsymbol{m}, \boldsymbol{m}'$, there exists a marking $\boldsymbol{m}'' \geq \boldsymbol{m}'$ such that $\boldsymbol{m} \rightarrow^{*} \boldsymbol{m}''$. This problem belongs to EXPSPACE [19]. Recall that quasi-liveness asks whether for each transition $t \in T \cup \{t_{sc}\}$, it is the case that $\{\mathsf{i}\colon 1\} \rightarrow^{*} \boldsymbol{m}$ for some some marking $\boldsymbol{m}$ that enables t, i.e. such that $\boldsymbol{m} \geq {}^\bullet t$. The latter is a coverability question. Hence, quasi-liveness amounts to $|T| + 1$ coverability queries, which can be checked in EXPSPACE. □

We further show that the previous result can be extended to k-soundness through the following lemma.

Lemma 3.6. *Given a workflow net $\mathcal{N}$ and $k > 0$, one can compute, in polynomial time, a workflow net $\mathcal{N}'$ with $\|\mathcal{N}'\| = \|\mathcal{N}\| + \log(k)$ such that, for all $c > 0$, $\mathcal{N}$ is ck-sound iff $\mathcal{N}'$ is c-sound.*

Proof. Let $\mathcal{N} = (P, T, F)$. We define $\mathcal{N}' := (P', T', F')$ that rescales everything by k. Formally, we add two new places that are the new initial and final places $P' := P \cup \{\mathsf{i}', \mathsf{f}'\}$. We denote by i and f the previous initial and final places. We add two new transitions t_i and t_f defined by:

$$\begin{aligned} {}^\bullet t_\mathsf{i}[\mathsf{i}'] = 1 \text{ and } {}^\bullet t_\mathsf{i}[p] &= 0 \text{ for } p \neq \mathsf{i}', \\ t_\mathsf{i}^\bullet[\mathsf{i}] = k \text{ and } t_\mathsf{i}^\bullet[p] &= 0 \text{ for } p \neq \mathsf{i}, \\ {}^\bullet t_\mathsf{f}[\mathsf{f}] = k \text{ and } {}^\bullet t_\mathsf{f}[p] &= 0 \text{ for } p \neq \mathsf{f}, \\ t_\mathsf{f}^\bullet[\mathsf{f}'] = 1 \text{ and } t_\mathsf{f}^\bullet[p] &= 0 \text{ for } p \neq \mathsf{f}'. \end{aligned}$$

It is straightforward that $\mathcal{N}'$ satisfies the lemma. □

Corollary 3.7. *The k-soundness problem is in EXPSPACE.*

Proof. It suffices to invoke Theorem 3.6 with $c = 1$, and test 1-soundness of the resulting workflow net via Theorem 3.5. □

3.2 EXPSPACE-hardness

Let us now establish EXPSPACE-hardness of classical soundness. We will need the forthcoming lemma that essentially states that so-called reversible Petri nets can count up to (or down from) a doubly exponential number. Formally, we say that a Petri net $\mathcal{N} = (P, T, F)$ is *reversible* if each transition of $\mathcal{N}$ has an inverse, *i.e.* for every $t \in T$, there exists $t^{-1} \in T$

such that ${}^\bullet(t^{-1}) = t^\bullet$ and $(t^{-1})^\bullet = {}^\bullet t$. Note that for reversible Petri nets, it is the case that $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$ if and only if $\boldsymbol{m}' \rightarrow^* \boldsymbol{m}$. To emphasise this, we will sometimes write $\boldsymbol{m} \leftrightarrow^* \boldsymbol{m}'$.

Lemma 3.8 ([17, Lemma 3]). *Let $\mathcal{N}$ be a reversible Petri net and let $\boldsymbol{m}$ and $\boldsymbol{m}'$ be two markings. Let $n \coloneqq \text{size}(\mathcal{N}, \boldsymbol{m}, \boldsymbol{m}')$. There exists $c_n \in 2^{2^{O(n)}}$ such that if $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$ then $\boldsymbol{m} \rightarrow^\rho \boldsymbol{m}'$ for a c_n-bounded run ρ.*

Lemma 3.9 ([17, reformulation of Lemma 6 and Lemma 8]). *Let $n \in \mathbb{N}$ and $c_n \in 2^{2^{O(n)}}$. There exists a reversible Petri net $\mathcal{N}_n = (P_n, T_n, F_n)$ with four distinguished places $s, c, f, b \in P_n$. Let $\boldsymbol{m}_n \coloneqq \{s\colon 1, c\colon 1\}$ and $\boldsymbol{m}'_n \coloneqq \{f\colon 1, c\colon 1, b\colon c_n\}$. The following holds for all $\boldsymbol{m}$:*

1. *$\boldsymbol{m}_n \leftrightarrow^* \boldsymbol{m}'_n$;*
2. *$\boldsymbol{m}_n \leftrightarrow^* \boldsymbol{m}$ and $\boldsymbol{m}[f] > 0$ implies $\boldsymbol{m} = \boldsymbol{m}'_n$;*
3. *$\boldsymbol{m} \leftrightarrow^* \boldsymbol{m}'_n$ and $\boldsymbol{m}[s] > 0$ implies $\boldsymbol{m} = \boldsymbol{m}_n$;*
4. *if $\boldsymbol{m} < \boldsymbol{m}'_n$ and $\boldsymbol{m}[f] = 0$ then no transition can be fired from $\boldsymbol{m}$;*
5. *for all $p \in P_n$ there exists $\boldsymbol{m}_n \leftrightarrow^* \boldsymbol{m}$ s.t. $\boldsymbol{m}[p] > 0$.*

Furthermore, $\mathcal{N}_n$ is: of polynomial size in n; constructible in polynomial time in n; and quasi-live both from $\boldsymbol{m}_n$ and $\boldsymbol{m}'_n$.

Theorem 3.10. *The classical soundness and 1-soundness problems are EXPSPACE-hard.*

Proof. We give a reduction from the reachability problem for reversible Petri nets. This problem is known to be EXPSPACE-complete [4, 17]. Let $\mathcal{N} = (P, T, F)$ be a reversible Petri net, and let $\boldsymbol{m}, \boldsymbol{m}'$ be two markings for which we would like to know whether $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$ in $\mathcal{N}$.

Let $n \coloneqq \text{size}(\mathcal{N}, \boldsymbol{m}, \boldsymbol{m}')$, let c_n be the value given by Theorem 3.8 for n, and let $\mathcal{N}_n = (P_n, T_n, F_n)$ be the Petri net given by Theorem 3.9 for c_n.

We construct a workflow net $\mathcal{N}' = (P', T', F')$ such that $\mathcal{N}'$ is classically sound if and only if $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$ in $\mathcal{N}$. To avoid any confusion, we will denote markings in $\mathcal{N}'$ by $\boldsymbol{n}, \boldsymbol{n}'$, etc.

The construction will ensure that

$$\boldsymbol{m} \rightarrow^* \boldsymbol{m}' \text{ in } \mathcal{N} \text{ iff } \mathcal{N}' \text{ is classically sound.} \quad (1)$$

Moreover, 1-soundness of $\mathcal{N}'$ will imply $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$, which will prove that both classical soundness and 1-soundness are EXPSPACE-hard.

Informally, we wish for $\mathcal{N}'$ to convert $\{\mathsf{i}\colon 1\}$ into $\boldsymbol{m}$, simulate $\mathcal{N}$, and convert $\boldsymbol{m}'$ into $\{\mathsf{f}\colon 1\}$. Here, the reversibility of $\mathcal{N}$ is crucial to ensure soundness: "erroneous runs" should still be able to reach $\{\mathsf{f}\colon 1\}$. This approach is however idealised since $\mathcal{N}'$ has no way to test whether $\mathcal{N}$ has reached $\boldsymbol{m}'$. By Theorem 3.8, we know that $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$ is witnessed by a c_n-bounded run. Hence, c_n tokens per place suffice. Thus, we add a dual place $\overline{p}$ for each place p of $\mathcal{N}$ such that, in marking $\boldsymbol{n}$, $\overline{p}$ contains $c_n - \boldsymbol{n}[p]$ tokens. This allows to implement a form of equality test for $\boldsymbol{m}'$. Yet, this is again oversimplified as it must be implemented with great care. Indeed, if $\mathcal{N}$ has reached $\boldsymbol{m}'' > \boldsymbol{m}'$, then the gadget for equality test will consume some tokens, but *not all* c_n tokens from the dual places (recall that producing and consuming c_n cannot be achieved atomically, but rather via $\mathcal{N}_n$). Thus, a mechanism is needed to restore $\boldsymbol{m}''$ *and* the budget, as $\boldsymbol{m}'' \rightarrow^* \boldsymbol{m}'$ could hold.

Formally, the set of places P' consists of: P; its disjoint copy $\overline{P} \coloneqq \{\overline{p} \mid p \in P\}$; seven extra places

$$\{\mathsf{i}, \mathsf{f}, p_{\text{start}}, p_{\text{inProgress}}, p_{\text{cover}}, p_{\text{simple}}, p_{\text{canFire}}\};$$

two disjoint copies of P_n (from Theorem 3.9), with one copy of b removed. One of the copies will be marked with $\heartsuit$ to avoid any confusion, thus we write *e.g.* $p^\heartsuit \in P_n^\heartsuit$. The two places b and $b^\heartsuit$ are merged into a single place denoted b.

Before presenting the transitions, we would like to emphasise that, intuitively, place $\overline{p} \in \overline{P}$ will contain a "budget" of tokens that is an upper bound on how many more tokens can be present in p. Most of the time, for every marking $\boldsymbol{n}$ and place $p \in P$, we will keep $\boldsymbol{n}[p] + \boldsymbol{n}[\overline{p}] = c_n$ as an invariant.

In Figure 4, we present the most relevant parts of $\mathcal{N}'$. Formally, the set of transitions is divided into four subsets $T' = T_1 \cup T_2 \cup T_3 \cup T_4$. Transitions will be defined by giving ${}^\bullet t'[p]$ and $t'^\bullet[p]$. The values are zero on unmentioned places.

First, for every transition $t \in T$, we define $t' \in T_1$ by:

- ${}^\bullet t'[p] \coloneqq {}^\bullet t[p]$ and $t'^\bullet[p] \coloneqq t^\bullet[p]$ for all $p \in P$;
- ${}^\bullet t'[\overline{p}] \coloneqq t^\bullet[p]$ and $t'^\bullet[\overline{p}] \coloneqq {}^\bullet t[p]$ for all $p \in P$;
- ${}^\bullet t'[p_{\text{canFire}}] = t'^\bullet[p_{\text{canFire}}] \coloneqq 1$.

It is easy to see that since $\mathcal{N}$ is a reversible Petri net, for every transition in T_1, its reverse is also in T_1. We will say that T_1 is reversible. Notice that, for all $t' \in T_1$ and $p \in P$, the sum of tokens in p and $\overline{p}$ does not change under t'.

Second, for every $t \in T_n$, we add $t' \in T_2$ such that:

- ${}^\bullet t'[p] \coloneqq {}^\bullet t[p]$ and $t'^\bullet[p] \coloneqq t^\bullet[p]$ for all $p \in P_n$;
- ${}^\bullet t'[\overline{p}] \coloneqq {}^\bullet t[b]$ and $t'^\bullet[\overline{p}] \coloneqq t^\bullet[b]$ for all $\overline{p} \in \overline{P}$.

Intuitively, places in $\overline{P}$ behave as b to initialise the budget of c_n tokens. Similarly, for every $t^\heartsuit \in T_n^\heartsuit$, we add $t' \in T_3$ such that:

- ${}^\bullet t'[p^\heartsuit] \coloneqq {}^\bullet t[p^\heartsuit]$ and $t'^\bullet[p^\heartsuit] \coloneqq t^\bullet[p^\heartsuit]$ for all $p^\heartsuit \in P_n^\heartsuit$;
- ${}^\bullet t'[\overline{p}] \coloneqq {}^\bullet t[b]$ and $t'^\bullet[\overline{p}] \coloneqq t^\bullet[b]$ for all $\overline{p} \in \overline{P}$.

Note that since $\mathcal{N}_n$ is reversible, both T_2 and T_3 are reversible.

The set T_4 consists of the ten remaining transitions

$$\{t_{\text{hard}}, t_{\text{start}}, t_{\boldsymbol{m}}, t_{\boldsymbol{m}'}, t_{\boldsymbol{m}'}^{-1}, t_{\text{isEmpty}}, t_{\text{reach}}, t_{\text{reach}}^{-1}, t_{\text{simple}}, t_{\text{simple2}}\}.$$

Intuitively, the first two are needed to initialise places in $\overline{P}$ with c_n tokens; the next three transitions respectively add $\boldsymbol{m}$, $-\boldsymbol{m}'$ and $\boldsymbol{m}'$ to P; the next three transitions transfer tokens towards the final places; and the last two transitions are needed for quasi-liveness. Formally,

- ${}^\bullet t_{\text{hard}}[\mathsf{i}] = t_{\text{hard}}^\bullet[s] = t_{\text{hard}}^\bullet[c] \coloneqq 1$;
- ${}^\bullet t_{\text{start}}[f] = {}^\bullet t_{\text{start}}[c] = t_{\text{start}}^\bullet[p_{\text{start}}] \coloneqq 1$;
- $t_{\boldsymbol{m}}^\bullet[p] = {}^\bullet t_{\boldsymbol{m}}[\overline{p}] \coloneqq \boldsymbol{m}[p]$ for all $p \in P$; and ${}^\bullet t_{\boldsymbol{m}}[p_{\text{start}}] = t_{\boldsymbol{m}}^\bullet[p_{\text{inProgress}}] = t_{\boldsymbol{m}}^\bullet[p_{\text{canFire}}] \coloneqq 1$;
- ${}^\bullet t_{\boldsymbol{m}'}[p] = t_{\boldsymbol{m}'}^\bullet[\overline{p}] \coloneqq \boldsymbol{m}'[p]$ for all $p \in P$; ${}^\bullet t_{\boldsymbol{m}'}[p_{\text{inProgress}}] = {}^\bullet t_{\boldsymbol{m}}[p_{\text{canFire}}] = t_{\boldsymbol{m}'}^\bullet[p_{\text{cover}}] \coloneqq 1$; and $t_{\boldsymbol{m}'}^{-1}$ is its reverse transition;

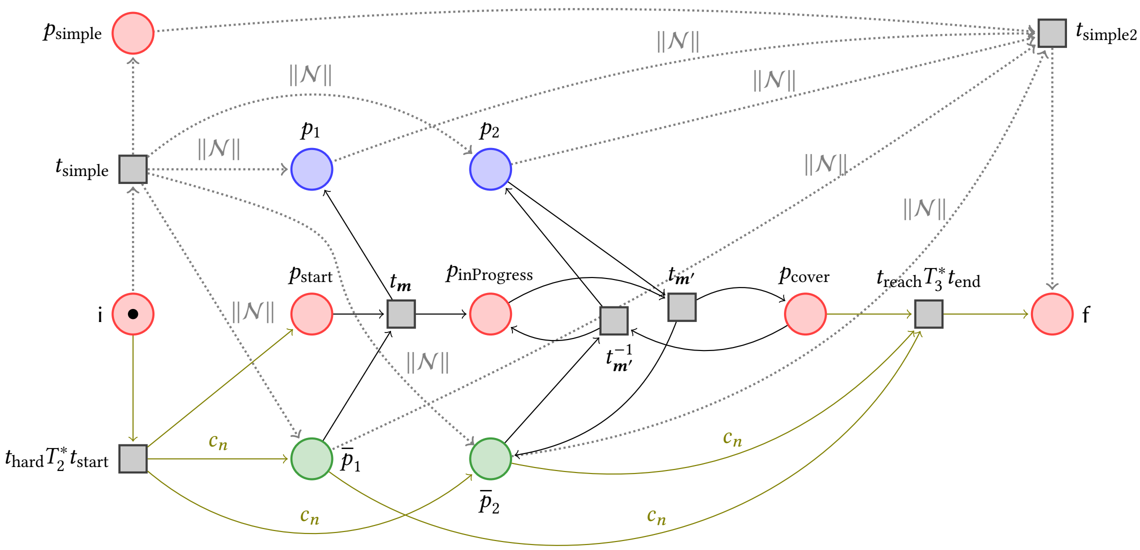

Figure 4. A workflow net $\mathcal{N}'$ which is classically sound iff $\boldsymbol{m} \rightarrow^* \boldsymbol{m'}$ in the reversible Petri net $\mathcal{N} = (P, T, F)$. In the example, $P = \{p_1, p_2\}$, $\boldsymbol{m} = (1, 0)$ and $\boldsymbol{m'} = (0, 1)$. The original places are blue, their copies are green, and other new places are red. We omit the transitions in T_1 that originated from T (recall that these transitions are modified to consume and produce tokens also in green places), and we omit the place p_{canFire} (used only to allow transitions in T_1 to fire). We only sketch transitions in T_2 and T_3 (and some other transitions), by writing the intuitive meaning of the gadgets that add/remove c_n tokens (arcs of these "transitions" are marked with a different color). The transition t_{hard} initiates the bottom part of $\mathcal{N}'$ (by filling the green places with c_n tokens) that checks $\boldsymbol{m} \rightarrow^* \boldsymbol{m'}$. The transition t_{simple} initiates the top part of $\mathcal{N}'$. We denote arcs in the top part with dotted gray color. This part is rather trivial and its only purpose is to ensure quasi-liveness of transitions in T_1 (by filling blue and green places with $\|\mathcal{N}\|$ tokens).

- ${}^\bullet t_{\text{reach}}[p_{\text{cover}}] = t^\bullet_{\text{reach}}[f^\heartsuit] = t^\bullet_{\text{reach}}[c^\heartsuit] \coloneqq 1$; and t^{-1}_{reach} is its reverse transition;
- ${}^\bullet t_{\text{end}}[s^\heartsuit] = {}^\bullet t_{\text{end}}[c^\heartsuit] = t^\bullet_{\text{end}}[\mathsf{f}] \coloneqq 1$;
- ${}^\bullet t_{\text{simple}}[\mathsf{i}] = t^\bullet_{\text{simple}}[p_{\text{simple}}] = t^\bullet_{\text{simple}}[p_{\text{canFire}}] \coloneqq 1$; and $t^\bullet_{\text{simple}}[p] = t^\bullet_{\text{simple}}[\overline{p}] \coloneqq \|\mathcal{N}\|$ for all $p \in P$;
- ${}^\bullet t_{\text{simple2}}[p] = {}^\bullet t_{\text{simple2}}[\overline{p}] \coloneqq \|\mathcal{N}\|$ for all $p \in P$; and ${}^\bullet t_{\text{simple2}}[p_{\text{simple}}] = {}^\bullet t_{\text{simple2}}[p_{\text{canFire}}] = t^\bullet_{\text{simple2}}[\mathsf{f}] \coloneqq 1$.

Recall that $P \subseteq P'$ and that $\boldsymbol{m}$ is a marking on P. To ease the notation, we will assume that $\boldsymbol{m}$ is a marking on P' (with 0 tokens in places from $P' \setminus P$).

We are ready to prove Equation 1. Notice that for every reachable configuration $\{\mathsf{i}\colon 1\} \rightarrow^\rho \boldsymbol{n}$ the value $\boldsymbol{n}[p_{\text{canFire}}]$ is always equal to $\boldsymbol{n}[p_{\text{simple}}]$ or $\boldsymbol{n}[p_{\text{inProgress}}]$ (depending on whether the first transitions of ρ is t_{simple} or t_{hard}). For readability, we omit the value of p_{canFire} in the markings of $\mathcal{N}'$.

$\Leftarrow$) Suppose that $\mathcal{N}'$ is 1-sound (we will not rely on $\mathcal{N}'$ being quasi-live). By Theorem 3.9 (1), we know that

$$\{\mathsf{i}\colon 1\} \rightarrow^{t_{\text{hard}}} \{s\colon 1, c\colon 1\} \rightarrow^* \{f\colon 1, c\colon 1, b\colon c_n\} + \sum_{\overline{p} \in \overline{P}} \{\overline{p}\colon c_n\}.$$

Let us denote the last marking by $\boldsymbol{n}$. Notice that

$$\boldsymbol{n} \rightarrow^{t_{\text{start}} t_{\boldsymbol{m}}} \{p_{\text{inProgress}}\colon 1, b\colon c_n\} + \boldsymbol{m} + \sum_{\overline{p} \in \overline{P}} \{\overline{p}\colon c_n - \boldsymbol{m}[p]\}.$$

We denote the latter marking by $\boldsymbol{n'}$. Since $\mathcal{N}'$ is 1-sound, $\boldsymbol{n'} \rightarrow^\rho \{\mathsf{f}\colon 1\}$ for some run ρ. This is possible if t_{reach} was fired at least once in ρ. Let $\boldsymbol{n}_1 \rightarrow^{t_{\text{reach}}} \boldsymbol{n}_2$ be the last time t_{reach} was fired in ρ. We claim that $\boldsymbol{n}_2 = \{f^\heartsuit\colon 1, c^\heartsuit\colon 1, b\colon c_n\} + \sum_{\overline{p}} \{\overline{p}\colon c_n\}$. Indeed, it has to be that

$$\boldsymbol{n}_2 \rightarrow^{\rho'} \{s^\heartsuit\colon 1, c^\heartsuit\colon 1\} \rightarrow^{t_{\text{end}}} \{\mathsf{f}\colon 1\},$$

where ρ' uses transition only from T_3. By Theorem 3.9 (4), this is possible only if $\boldsymbol{n}_2$ is as claimed. Let ρ'' be the prefix of the run ρ from $\boldsymbol{n'}$ such that it ends in $\boldsymbol{n}_1$. Finally, ρ'', when restricted to P, witnesses reachability for $\boldsymbol{m} \rightarrow^* \boldsymbol{m'}$.

$\Rightarrow$) Suppose that $\boldsymbol{m} \rightarrow^* \boldsymbol{m'}$. The proof of 1-soundness is very technical and can be found in the appendix. In a nutshell, recall that T_1, T_2 and T_3 are reversible, and for $t_{\boldsymbol{m'}}, t_{\text{reach}} \in T_4$ we include their reverse transitions. This allows us to revert any configuration to a configuration from which it is easy to define a run to $\{\mathsf{f}\colon 1\}$.

To conclude this implication, we need to prove that $\mathcal{N}'$ is quasi-live. Indeed, from the proof of 1-soundness it is easy to see that $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$ implies that all transitions are fireable, with the possible exception of transitions from T_1. However,

$$\{\mathsf{i}\colon 1\} \rightarrow^{t_{\text{simple}}} \{p_{\text{simple}}\colon 1\} + \sum_{p \in P} \{p\colon \|\mathcal{N}\|, \overline{p}\colon \|\mathcal{N}\|\}.$$

From the latter configuration, any transition of T_1 is fireable.

Finally, observe that $\mathcal{N}'$ is a workflow net. Indeed, by taking t_{simple} we put tokens in P and $\overline{P}$. Each place from copies in $\mathcal{N}_n$ is on a path from i to f by Theorem 3.9 (5). The remaining places are clearly on such a path by definition (see Figure 4). □

4 Bounds on vector reachability

In this section, we present technical results that will be helpful to establish complexity bounds in the forthcoming sections. It is well-known that Petri nets are complex due to their nonnegativity constraints. Namely, markings are over $\mathbb{N}$ (not $\mathbb{Z}$), which blocks transitions from being fired whenever the amount of tokens would drop below zero. By lifting this restriction, *i.e.* allowing markings over $\mathbb{Z}$, transitions cannot be blocked and we obtain a provably simpler model (*e.g.* see [12]). We recall known results that provide bounds on reachability problems for vectors over $\mathbb{Z}$. Based on these results, we will derive useful bounds for the next sections.

4.1 Integer linear programs

Given positive natural numbers $n, m > 0$, let $\mathbf{A} \in \mathbb{Z}^{m \times n}$ be an integer matrix, $\boldsymbol{b} \in \mathbb{Z}^m$ an integer vector and $\boldsymbol{x} = (x_1, \ldots, x_n)^\mathsf{T}$ a vector of variables. We say that $G \coloneqq \mathbf{A} \cdot \boldsymbol{x} \geq \boldsymbol{b}$ is an $(m \times n)$-*ILP*, that is, an integer linear program (ILP) with m inequalities and n variables. The set of solutions of G is

$$\llbracket G \rrbracket \coloneqq \{\boldsymbol{\mu} \in \mathbb{Z}^n \mid \mathbf{A} \cdot \boldsymbol{\mu} \geq \boldsymbol{b}\},$$

and the set of natural solutions is $\llbracket G \rrbracket_{\geq 0} \coloneqq \llbracket G \rrbracket \cap \mathbb{N}^n$. We will only be interested in the natural solutions $\llbracket G \rrbracket_{\geq 0}$ but sometimes we will need to refer to $\llbracket G \rrbracket$. We shall assume that these sets are equal, by implicitly adding a new inequality for each variable specifying that it is greater or equal to 0.

Often it is convenient to write an equality constraint, *e.g.* $x - y = 0$. This can be simulated by two inequalities, so we will allow to define G both with equalities and inequalities.

We introduce some notation about *semi-linear* sets from [6] to obtain bounds on the sizes of solutions to ILPs. A set of vectors is called *linear* if it is of the form $L(\boldsymbol{b}, P) = \{\boldsymbol{b} + \lambda_1 \boldsymbol{p}_1 + \ldots + \lambda_k \boldsymbol{p}_k \mid \lambda_1, \ldots, \lambda_k \in \mathbb{N}\}$, where $\boldsymbol{b} \in \mathbb{Z}^n$ is a vector and $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k\} \subseteq \mathbb{Z}^n$ is a finite set of vectors. A set is called *hybrid linear* if it is of the form $L(B, P) = \bigcup_{\boldsymbol{b} \in B} L(\boldsymbol{b}, P)$ for a finite set of vectors $B = \{\boldsymbol{b}_1 \ldots, \boldsymbol{b}_\ell\} \subseteq \mathbb{Z}^n$.

The *size* of a finite set of vectors B and of an $(m \times n)$-ILP G are defined respectively as $\|B\| \coloneqq \max_{\boldsymbol{b} \in B} \|\boldsymbol{b}\|$ and $\|G\| \coloneqq \|\mathbf{A}\| + \|\boldsymbol{b}\| + m + n$.

Lemma 4.1 ([30], presentation adapted from [6, Prop. 3])**.** *Let G be an $(m \times n)$-ILP. It is the case that $\llbracket G \rrbracket = \bigcup_{i \in I} L(B_i, P_i)$, where $max_{i \in I} \|B_i\| \leq \|G\|^{O(n \log n)}$.*

For the forthcoming lemmas, recall that $\boldsymbol{c} = (c, \ldots, c)$.

Lemma 4.2. *Let G be an $(m \times n)$-ILP. There exists a number $c \leq \|G\|^{O(n \log n)}$ such that for all $\boldsymbol{\mu} \in \llbracket G \rrbracket_{\geq 0}$, there is some $\boldsymbol{\mu}' \in \llbracket G \rrbracket_{\geq 0}$ such that $\boldsymbol{\mu}' \leq \boldsymbol{\mu}$ and $\boldsymbol{\mu}' \leq \boldsymbol{c}$.*

Proof. Recall that we can assume $\llbracket G \rrbracket = \llbracket G \rrbracket_{\geq 0}$. By Theorem 4.1, $\llbracket G \rrbracket = \bigcup_{i \in I} L(B_i, P_i)$. We set $c \coloneqq \max_{i \in I} \|B_i\|$. Let $\boldsymbol{\mu} \in \llbracket G \rrbracket_{\geq 0}$. There exist $i \in I$ and $\boldsymbol{b} \in B_i$ such that $\boldsymbol{\mu} \in L(\boldsymbol{b}, P_i)$. Note that $\boldsymbol{p} \geq \boldsymbol{0}$ for all $\boldsymbol{p} \in P_i$. Hence, we have $\boldsymbol{b} \in \llbracket G \rrbracket_{\geq 0}$, $\boldsymbol{b} \leq \boldsymbol{\mu}$ and $\boldsymbol{b} \leq \boldsymbol{c}$. Thus, we can set $\boldsymbol{\mu}' \coloneqq \boldsymbol{b}$. □

Lemma 4.3. *Let $G = \mathbf{A} \cdot \boldsymbol{x} \geq \boldsymbol{b}$ be an $(m \times n)$-ILP, where $\boldsymbol{b} \geq \boldsymbol{0}$. There exists $c \leq \|G\|^{O((m+n)\log(m+n))}$ such that the following holds. For every $\boldsymbol{\mu} \in \llbracket G \rrbracket_{\geq 0}$, there exists $\boldsymbol{\nu} \in \llbracket G \rrbracket_{\geq 0}$ such that $\boldsymbol{\nu} \leq \boldsymbol{\mu}$, $\boldsymbol{\nu} \leq \boldsymbol{c}$, and $\mathbf{A} \cdot \boldsymbol{\nu} \leq \mathbf{A} \cdot \boldsymbol{\mu}$.*

Proof. Let $x_1, \ldots, x_n$ be the variables of G. We define a $(3m \times (m+n))$-ILP G' by slightly modifying G. For every inequality in the original ILP G, we add one fresh variable. We denote them $y_1, \ldots, y_m$. Now, recall that the inequalities in G are of the form: $\sum_{i=1}^n \mathbf{A}[j, i] \cdot x_i \geq \boldsymbol{b}[j]$ for $j \in [1..m]$. The ILP G' is defined with the same inequalities, plus m new equalities (recall that this requires $2m$ inequalities): $\sum_{i=1}^n \mathbf{A}[j, i] \cdot x_i - y_j = 0$ for $j \in [1..m]$.

Notice that, in solutions for G', the variables y_j are uniquely determined by the valuation of $x_1, \ldots, x_n$. For convenience, we will write $\boldsymbol{\mu}[x_i], \boldsymbol{\mu}'[y_j]$ when referring to the components of solutions. For every $\boldsymbol{\mu} \in \llbracket G \rrbracket_{\geq 0}$, there is a unique $\boldsymbol{\mu}' \in \llbracket G' \rrbracket$ such that $\boldsymbol{\mu}'[x_i] = \boldsymbol{\mu}[x_i]$ for all $i \in [1..n]$. Thus, since $\boldsymbol{b} \geq \boldsymbol{0}$, we have $\llbracket G' \rrbracket_{\geq 0} = \{\boldsymbol{\mu}' \mid \boldsymbol{\mu} \in \llbracket G \rrbracket\}$. We define c as the constant from Theorem 4.2 for G'. Now, let $\boldsymbol{\mu} \in \llbracket G \rrbracket_{\geq 0}$ and let $\boldsymbol{\mu}' \in \llbracket G' \rrbracket_{\geq 0}$ be its corresponding solution. By Theorem 4.2, there exists $\boldsymbol{\nu}' \in \llbracket G' \rrbracket_{\geq 0}$ such that $\boldsymbol{\nu}' \leq \boldsymbol{\mu}'$ and $\boldsymbol{\nu}' \leq \boldsymbol{c}$. We define $\boldsymbol{\nu} \in \llbracket G \rrbracket_{\geq 0}$ as the solution corresponding to $\boldsymbol{\nu}'$. It is clear that $\boldsymbol{\nu} \leq \boldsymbol{\mu}$ and $\boldsymbol{\nu} \leq \boldsymbol{c}$. For the remaining part, fix $j \in [1..m]$. Recall that $\boldsymbol{\nu}'[y_j] = \sum_{i=1}^n \mathbf{A}[j, i] \cdot \boldsymbol{\nu}'[x_i]$ and $\boldsymbol{\mu}'[y_j] = \sum_{i=1}^n \mathbf{A}[j, i] \cdot \boldsymbol{\mu}'[x_i]$. Thus,

$$\sum_{i=1}^n \mathbf{A}[j, i] \cdot \boldsymbol{\nu}[x_i] \leq \sum_{i=1}^n \mathbf{A}[j, i] \cdot \boldsymbol{\mu}[x_i],$$

which concludes the proof. □

4.2 Steinitz Lemma

Let us recall the Steinitz Lemma [20] based on the presentation of [11].

Lemma 4.4. *Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$ be such that $\sum_{i=1}^n \boldsymbol{x}_i = \boldsymbol{0}$ and $\|\boldsymbol{x}_i\| \leq 1$ for all i. There exists a permutation π on $[1..n]$ such that*

$$\left\| \sum_{j=1}^{i} \boldsymbol{x}_{\pi(j)} \right\| \leq d \qquad \textit{for all } i \in [1..n].$$

The following formulation of the lemma, which is depicted graphically in Figure 5, will be more convenient for us.

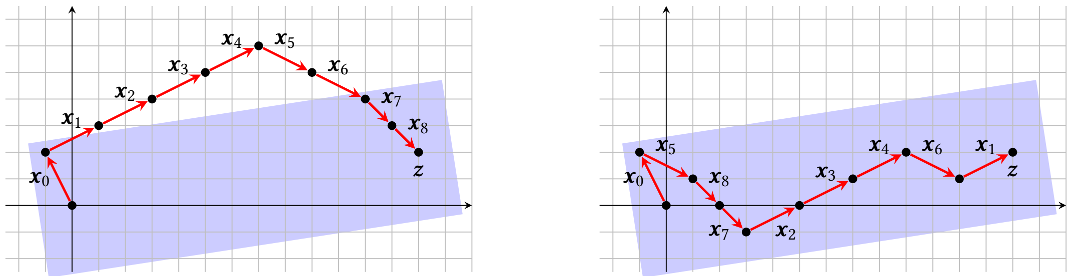

Figure 5. An example of Theorem 4.5 in dimension $d = 2$. The vectors $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_n$ form a path from $\mathbf{0}$ to $\boldsymbol{z}$. The colored background highlights points that are within some bounded distance from the line $\mathbf{0}$ to $\boldsymbol{z}$ (the bound depends on d and $\boldsymbol{x}_i$, but not on $\boldsymbol{z}$). In the right picture, the vectors are reordered so that they all fit within the bound. The additional constraints are that: the first vector $\boldsymbol{x}_0$ remains first ($\pi(0) = 0$); and, *in some way*, the points are getting closer to $\boldsymbol{z}$ ($0 \le c_0 \le c_1 \le \ldots \le c_n$).

Lemma 4.5. *Let* $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{Z}^d$, $b := \max_{j=0}^n \|\boldsymbol{x}_j\|$, *and* $\boldsymbol{z} := \sum_{j=0}^n \boldsymbol{x}_j$. *There exists a permutation* π *of* $[0..n]$ *such that:* $\pi(0) = 0$*; and there exist* $0 \le c_0 \le c_1 \le \ldots \le c_n \le 1$, *where*

$$\left\| \sum_{j=0}^{i} \boldsymbol{x}_{\pi(j)} - c_i \cdot \boldsymbol{z} \right\| \le b(d+2) \qquad \text{for all } i \in [0..n].$$

5 Generalised soundness

A Petri net $\mathcal{N}$ is $\mathbb{Z}$-*bounded* from a marking $\boldsymbol{m}$ if there exists $b \in \mathbb{N}$ such that $\boldsymbol{m} \rightarrow^*_{\mathbb{Z}} \boldsymbol{m'} \ge \mathbf{0}$ implies $\boldsymbol{m'} \le \boldsymbol{b}$ (*i.e.* we replace $\rightarrow^*$ with $\rightarrow^*_{\mathbb{Z}}$ in the definition of boundedness). Otherwise, we say that $\mathcal{N}$ is $\mathbb{Z}$-*unbounded*. Observe that being $\mathbb{Z}$-bounded does not mean that the set of reachable markings is bounded by below, but only from above.

Let $k \ge 0$. We say that $\mathcal{N}$ is *strongly k-sound* if for every $\boldsymbol{m} \in \mathbb{N}^P$ such that $\{\mathsf{i}\colon k\} \rightarrow^*_{\mathbb{Z}} \boldsymbol{m}$, it holds that $\boldsymbol{m} \rightarrow^* \{\mathsf{f}\colon k\}$. Note that every strongly k-sound net is also k-sound.

The aim of the next three subsections is to prove the following theorem.

Theorem 5.1. *Generalised soundness is in PSPACE.*

The proof has two parts. First, we prove that if there is a k for which the net is not k-sound, then there is also such a k bounded exponentially. Second, we prove that k-soundness for exponentially bounded k can be verified in PSPACE.

5.1 Nonredundant workflow nets

Fix a workflow net $\mathcal{N} = (P, T, F)$. We say that a place $p \in P$ is *nonredundant* if there exists $k \in \mathbb{N}$ such that $\{\mathsf{i}\colon k\} \rightarrow^* \boldsymbol{m}$ and $\boldsymbol{m}[p] > 0$. By removing a redundant place p from $\mathcal{N}$, we mean removing p from P and all transitions $t \in T$ such that ${}^\bullet t[p] > 0$. With the remaining transitions restricted to the domain $P \setminus \{p\}$, we obtain a new workflow net $\mathcal{N}' := (P \setminus \{p\}, T')$. It is clear that $\mathcal{N}$ is k-sound if and only if $\mathcal{N}'$ is k-sound for all $k \in \mathbb{N}$. Thus, in particular, this procedure preserves generalised soundness.

It will be convenient to assume that all places in the studied workflow nets are nonredundant. At first, it might seem that this requires coverability checks for every place. However, since the number of initial tokens is arbitrary, finding redundant places amounts to a simple polynomial-time saturation procedure. More details can be found in [29, Thm. 8, Def. 10, Sect. 3.2] (and in the appendix). We will call workflow nets without redundant places *nonredundant workflow nets*[3]. To summarise we conclude the following.

Proposition 5.2. *Given a workflow net* $\mathcal{N}$*, one can identify and remove all redundant places from it in polynomial time. The resulting workflow net* $\mathcal{N}'$ *is nonredundant. Moreover,* $\mathcal{N}$ *is* k*-sound if and only if* $\mathcal{N}'$ *is* k*-sound for all* $k \in \mathbb{N}$.

In the following lemma, intuitively, we show that the initial budget is small for nonredundant workflow nets.

Lemma 5.3. *Let* $\mathcal{N} = (P, T, F)$ *be a nonredundant workflow net and let* $p \in P$ *be a place. There exists* $k < (\|T\| + 2)^{|T|}$ *such that* $\{\mathsf{i}\colon k\} \rightarrow^* \boldsymbol{m}$ *and* $\boldsymbol{m}[p] > 0$.

Proof. A transition t *increases* a place p' if $\Delta(t)[p'] > 0$. We say that a run ρ *increases* p' if there exists $t \in \mathrm{supp}(\rho)$ that increases p'. For the proof of the lemma, we assume that $p \neq \mathsf{i}$, as otherwise it suffices to define $k = 1$.

We prove that for all run $\{\mathsf{i}\colon k'\} \rightarrow^\rho \boldsymbol{m'}$, there is a run π such that: $\mathrm{supp}(\pi) = \mathrm{supp}(\rho)$, and $\{\mathsf{i}\colon k\} \rightarrow^\pi \boldsymbol{m}$ for some $k < (\|T\| + 2)^n$ and $\boldsymbol{m}$, where $\boldsymbol{m}[p'] \ge 1$ for all places p' increased by ρ. Note that, since $\mathcal{N}$ is a nonredundant workflow net, if we exhibit such a run then we are done as there exists ρ that increases p.

Let $\{\mathsf{i}\colon k'\} \rightarrow^\rho \boldsymbol{m'}$. The proof is by induction on n, where $\mathrm{supp}(\rho) = \{t_1, \ldots, t_n\}$. Assume $n = 1$. The only transition used by ρ is t_1, which increases p. Recall that $\|T\|$ is the maximal number occurring on any arc of $\mathcal{N}$. Since workflow nets start with tokens only in place i, we must have $\{\mathsf{i}\colon \|T\|\} \ge {}^\bullet\pi$. It suffices to define $\pi := t_1$ and $k := \|T\| < (\|T\| + 2)$.

For the induction step, assume $n > 1$ and that the lemma holds for $n - 1$. Let ρ_{n-1} be the longest prefix of ρ such that $\mathrm{supp}(\rho_{n-1}) = \{t_1, \ldots, t_{n-1}\}$. The induction hypothesis for

[3]The results in [29] deal with batch workflow nets, which are in particular nonredundant workflow nets.

ρ_{n-1} yields $k_{n-1} < (\|T\|+2)^{n-1}$, and π_{n-1} with $\text{supp}(\pi_{n-1}) = \{t_1, \ldots, t_{n-1}\}$. Let $\{\text{i}\colon k_{n-1}\} \rightarrow^{\pi_{n-1}} \boldsymbol{m}_{n-1}$. Note that $\text{supp}({}^\bullet t_n) \subseteq \text{supp}(\pi_{n-1}^\bullet) \cup \{i\}$ since ρ is a run, where t_n is fired. By repeating $\|T\| + 1$ times the run π_{n-1}, we get

$$\{\text{i}\colon (k_{n-1}+1)\cdot(\|T\|+1)\} \rightarrow^* \{\text{i}\colon \|T\|+1\} + (\|T\|+1)\cdot \boldsymbol{m}_{n-1}.$$

To ease the notation, let $\boldsymbol{n} \coloneqq \{\text{i}\colon \|T\| + 1\} + (\|T\| + 1) \cdot \boldsymbol{m}_n$. By definition of $\boldsymbol{m}_{n-1}$, it holds that $\boldsymbol{n}[p'] \geq \|T\| + 1$ for all $p' \in \pi^\bullet$. Furthermore, we can fire t_n from $\boldsymbol{n}$. Let $\boldsymbol{n} \rightarrow^{t_n} \boldsymbol{m}$. To conclude, consider a place p' increased by ρ. If it is increased by one of the transitions $t_1, \ldots, t_{n-1}$, then after firing t_n at least one token was left in p'. Otherwise, p' is increased by t_n. In both cases, we have $\boldsymbol{m}[p] \geq 1$. It remains to observe that $k = (k_{n-1} + 1) \cdot (\|T\| + 1) < (\|T\| + 2)^n$. ◻

5.2 Relating soundness and strong soundness

We recall a result by van Hee et al. that establishes a connection between the reachability relations $\rightarrow_\mathbb{Z}^*$ and $\rightarrow^*$.

Lemma 5.4 (adaptation of [29, Lemma 12])**.** *Let $\mathcal{N}$ be a nonredundant, generalised sound workflow net, and let $\boldsymbol{m}$ be a marking for which there exists $k \geq 0$ satisfying $\{\text{i}\colon k\} \rightarrow_\mathbb{Z}^* \boldsymbol{m}$. There exists $\ell \geq 0$ such that $\{\text{i}\colon k + \ell\} \rightarrow^* \boldsymbol{m} + \{\text{f}\colon \ell\}$.*

Note that Theorem 5.4 is an easy consequence of the definition of nonredundancy. Namely, it suffices to put "enough budget" in each place so that the run under $\rightarrow_\mathbb{Z}^*$ becomes a run under $\rightarrow^*$. We restate the result to give a bound on ℓ, and so that it does not need to assume $\mathcal{N}$ is generalised sound.

Lemma 5.5. *Let $\mathcal{N} = (P, T, F)$ be a nonredundant workflow net. Let k and $\boldsymbol{m} \in \mathbb{N}^P$ be such that $\{\text{i}\colon k\} \rightarrow_\mathbb{Z}^* \boldsymbol{m}$. There exist $\ell \leq (\|T\|+2)^{|T|} \cdot \max(\|T\|, k) \cdot |P|(|P|+2)$ and $\boldsymbol{m}' \in \mathbb{N}^P$ such that $\{\text{i}\colon \ell\} \rightarrow^* \boldsymbol{m}'$ and $\{\text{i}\colon \ell + k\} \rightarrow^* \boldsymbol{m} + \boldsymbol{m}'$.*

Proof. Let $\rho = t_1 t_2 \cdots t_n$ be such that $\{\text{i}\colon k\} \rightarrow_\mathbb{Z}^\rho \boldsymbol{m}$. Let us define $\boldsymbol{x}_0 \coloneqq \{\text{i}\colon k\}$ and $\boldsymbol{x}_j \coloneqq \Delta(t_j)$ for all $j \in [1..n]$. By Theorem 4.5, we can assume that the transitions t_j are ordered so that there exist $c_0, \ldots, c_n \geq 0$ where

$$\left\|\{\text{i}\colon k\} + \sum_{j=1}^{i} \Delta(t_j) - c_i \boldsymbol{m}\right\| \leq \max(\|T\|, k) \cdot (|P| + 2),$$

for all $i \in [0..n]$. Since $\boldsymbol{m} \geq 0$, we get for all $p \in P$:

$$\left(\{\text{i}\colon k\} + \sum_{j=1}^{i} \Delta(t_j)\right)[p] \geq -\max(\|T\|, k) \cdot (|P| + 2). \quad (2)$$

By Theorem 5.3, there exists $\ell_p \leq (\|T\|+2)^{|T|}$ such that for every place p there is a run $\{\text{i}\colon \ell_p\} \rightarrow^{\pi_p} \boldsymbol{m}_p$ with $\boldsymbol{m}_p[p] > 0$. Thus, to put $\max(\|T\|, k) \cdot (|P| + 2)$ tokens in all places, it suffices to repeat $\max(\|T\|, k) \cdot (|P| + 2)$ times the run π_p for every $p \in P$. This requires $\ell \leq (\|T\| + 2)^{|T|} \cdot \max(\|T\|, k) \cdot |P|(|P|+2)$ tokens. Let $\boldsymbol{m}'$ be the marking obtained afterwards. By (2), $\boldsymbol{m}'$ allows to fire ρ. Thus, we obtain $\{\text{i}\colon \ell\} \rightarrow^* \boldsymbol{m}'$ and $\{\text{i}\colon \ell + k\} \rightarrow^* \boldsymbol{m} + \boldsymbol{m}'$ as required. ◻

This lemma allows us to focus on $\rightarrow_\mathbb{Z}^*$ instead of $\rightarrow^*$.

Lemma 5.6. *Let $\mathcal{N} = (P, T, F)$ be a nonredundant workflow net. It is the case that $\mathcal{N}$ is generalised sound iff it is strongly k-sound for all $k \geq 0$. Moreover, if $\mathcal{N}$ is not strongly k-sound, then there exists $k' \leq k + (\|T\|+2)^{|T|} \cdot \max(\|T\|, k) \cdot |P|(|P|+2)$ such that $\mathcal{N}$ is not k'-sound.*

Proof. The "if" implication is trivial. Indeed, if $\mathcal{N}$ is not k-sound then it cannot be strongly k-sound.

To prove the "only if" implication, assume that $\mathcal{N}$ is not strongly k-sound. We show that there exists k' such that $\mathcal{N}$ is not k'-sound. We will also prove the promised bound on k'. Since $\mathcal{N}$ is not strongly k-sound, there must be some $\boldsymbol{m} \in \mathbb{N}^P$ and π such that $\{\text{i}\colon k\} \rightarrow_\mathbb{Z}^\pi \boldsymbol{m}$ and $\boldsymbol{m} \not\rightarrow^* \{\text{f}\colon k\}$. By Theorem 5.5, there exists $\ell \leq (\|T\| + 2)^{|T|} \cdot \max(\|T\|, k) \cdot |P|(|P| + 2)$ and $\boldsymbol{m}'$ such that $\{\text{i}\colon \ell\} \rightarrow^* \boldsymbol{m}'$ and $\{\text{i}\colon \ell + k\} \rightarrow^* \boldsymbol{m} + \boldsymbol{m}'$. If $\mathcal{N}$ is not ℓ-sound, then we are done. Otherwise, if $\mathcal{N}$ is ℓ-sound, then it must hold that $\boldsymbol{m}' \rightarrow^* \{\text{f}\colon \ell\}$. So, $\{\text{i}\colon \ell+k\} \rightarrow^* \boldsymbol{m}+\boldsymbol{m}' \rightarrow^* \boldsymbol{m}+\{\text{f}\colon \ell\}$. Recall that $\boldsymbol{m} \not\rightarrow^* \{\text{f}\colon k\}$. Thus, $\boldsymbol{m}+\{\text{f}\colon \ell\} \not\rightarrow^* \{\text{f}\colon \ell+k\}$. We are done since this means that $\mathcal{N}$ is not $(\ell + k)$-sound. ◻

5.3 Strong unsoundness occurs for small numbers

In this section, we will show that if there exists some k such that $\mathcal{N}$ is not strongly k-sound, then k is at most exponential in $|\mathcal{N}|$. We define an ILP which is closely related to the markings reachable from at least one initial number of tokens in $\mathcal{N}$. Essentially, the ILP will encode that there exists $k > 0$ and $\boldsymbol{m} \geq \boldsymbol{0}$ such that $\{\text{i}\colon k\} \rightarrow_\mathbb{Z}^* \boldsymbol{m}$. This can be done since only "firing counts" matter, *i.e.* $\boldsymbol{m} \rightarrow_\mathbb{Z}^\pi \boldsymbol{m}'$ implies $\boldsymbol{m} \rightarrow_\mathbb{Z}^{\pi'} \boldsymbol{m}'$ for any permutation π' of π.

Let $\mathcal{N} = (P, T, F)$ be a workflow net. We define $\text{ILP}_\mathcal{N} \coloneqq \mathbf{A} \cdot \boldsymbol{x} \geq \boldsymbol{0}$ as an ILP with $|P| + |T| + 1$ inequalities and $|T| + 1$ variables. The variables of ILP $_\mathcal{N}$ are $\boldsymbol{x} \coloneqq (\kappa, \tau_1, \ldots, \tau_{|T|})$. For ease of notation, we write $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{|T|})$. We assume an implicit bijection between T and $[1..|T|]$, *i.e.* for every $t \in T$ there is a unique i such that: $\boldsymbol{\tau}[t] = \tau_i$. The matrix $\mathbf{A}$ is defined by the following inequalities:

1. $\kappa + \sum_{t \in T} \boldsymbol{\tau}[t] \cdot \Delta(t)[\text{i}] \geq 0$,
2. $\kappa \geq 1$,
3. $\sum_{t \in T} \boldsymbol{\tau}[t] \cdot \Delta(t)[p] \geq 0$ for all $p \in P \setminus \{\text{i}\}$,
4. $\tau_i \geq 0$ for all $i \in [1..|T|]$.

The first two inequalities concern the initial "budget" k of tokens in i which is represented by κ. Intuitively, $\kappa \geq 1$ has to be at least as much as $\boldsymbol{\tau}$ consumes from the initial place. The last two inequalities guarantee that we obtain a marking over $\mathbb{N}^P$ and that the "firing count" is over $\mathbb{N}^T$.

Let $\boldsymbol{\mu}\colon \boldsymbol{x} \rightarrow \mathbb{N}$ be a solution to $\text{ILP}_\mathcal{N}$. We define

$$\text{marking}(\boldsymbol{\mu}) \coloneqq \{\text{i}\colon \mu(\kappa)\} + \sum_{t \in T}^{|T|} \mu(\tau_j) \cdot \Delta(t_j).$$

The following claim follows by definition of $\text{ILP}_\mathcal{N}$ and $\rightarrow_\mathbb{Z}^*$.

Claim 5.7. *Let $\boldsymbol{m} \in \mathbb{N}^P$ and $k > 0$. It holds that $\{\mathsf{i}\colon k\} \to_{\mathbb{Z}}^* \boldsymbol{m}$ iff there exists a solution $\boldsymbol{\mu}$ to $\mathrm{ILP}_{\mathcal{N}}$ such that $\mathrm{marking}(\boldsymbol{\mu}) = \boldsymbol{m}$ and $\boldsymbol{\mu}[\kappa] = k$.*

We conclude this part with the following bound.

Lemma 5.8. *Let $\mathcal{N}$ be a nonredundant workflow net. If $\mathcal{N}$ is strongly i-sound for all $1 \le i < k$, and not strongly k-sound, then $k \le c$, where c is the bound from Theorem 4.3 for $\mathrm{ILP}_{\mathcal{N}}$.*

Proof. For the sake of contradiction, assume that $k > c$ is as in the statement. Since $\mathcal{N}$ is not strongly k-sound, there exists a marking $\boldsymbol{m} \in \mathbb{N}^P$ such that $\{\mathsf{i}\colon k\} \to_{\mathbb{Z}}^* \boldsymbol{m}$ and $\boldsymbol{m} \not\to^* \{\mathsf{f}\colon k\}$. By Theorem 5.7, there exists a solution $\boldsymbol{\mu}$ to $\mathrm{ILP}_{\mathcal{N}}$ such that $\mathrm{marking}(\boldsymbol{\mu}) = \boldsymbol{m}$ and $\boldsymbol{\mu}[\kappa] = k$. By Theorem 4.3, there exists a solution $\boldsymbol{\mu}' \le \boldsymbol{\mu}$ to $\mathrm{ILP}_{\mathcal{N}}$ such that $\boldsymbol{\mu}'[\kappa] \le c < k = \boldsymbol{\mu}[\kappa]$ and $\mathbf{A}\boldsymbol{\mu}' \le \mathbf{A}\boldsymbol{\mu}$, where $\mathbf{A}$ is the underlying matrix of $\mathrm{ILP}_{\mathcal{N}}$. The latter inequality implies $\mathrm{marking}(\boldsymbol{\mu}') \le \mathrm{marking}(\boldsymbol{\mu})$.

Consider the vector $\boldsymbol{\pi} := \boldsymbol{\mu} - \boldsymbol{\mu}'$. We prove that $\boldsymbol{\pi}$ is a solution to $\mathrm{ILP}_{\mathcal{N}}$. Since $\boldsymbol{\mu}' \le \boldsymbol{\mu}$ we know that $\boldsymbol{\pi}$ is nonnegative. The inequalities of $\mathbf{A}$ are satisfied since $\mathbf{A}\boldsymbol{\pi} \ge \mathbf{0} \equiv \mathbf{A}\boldsymbol{\mu} \ge \mathbf{A}\boldsymbol{\mu}'$ and $\boldsymbol{\mu}'[\kappa] \le c < \boldsymbol{\mu}[\kappa]$. Thus, $\boldsymbol{\pi}$ is a solution to $\mathrm{ILP}_{\mathcal{N}}$.

By Theorem 5.7, $\{\mathsf{i}\colon \boldsymbol{\mu}'[\kappa]\} \to_{\mathbb{Z}}^* \mathrm{marking}(\boldsymbol{\mu}')$ and $\{\mathsf{i}\colon \boldsymbol{\pi}[\kappa]\} \to_{\mathbb{Z}}^* \mathrm{marking}(\boldsymbol{\pi})$. Recall that $\boldsymbol{\mu}'[\kappa], \boldsymbol{\pi}[\kappa] < \boldsymbol{\mu}[\kappa] = k$. By assumption, $\mathcal{N}$ is strongly $\boldsymbol{\mu}'[\kappa]$-sound and strongly $\boldsymbol{\pi}[\kappa]$-sound. Therefore, $\mathrm{marking}(\boldsymbol{\mu}') \to^* \{\mathsf{f}\colon \boldsymbol{\mu}'[\kappa]\}$ and $\mathrm{marking}(\boldsymbol{\pi}) \to^* \{\mathsf{f}\colon \boldsymbol{\pi}[\kappa]\}$. Since the function $\mathrm{marking}(\cdot)$ is linear, we get

$$\boldsymbol{m} = \mathrm{marking}(\boldsymbol{\mu}) = \mathrm{marking}(\boldsymbol{\mu}') + \mathrm{marking}(\boldsymbol{\pi}).$$

This implies $\boldsymbol{m} \to^* \{\mathsf{f}\colon \boldsymbol{\mu}'[\kappa]\} + \{\mathsf{f}\colon \boldsymbol{\pi}[\kappa]\} = \{\mathsf{f}\colon k\}$, which is a contradiction. ◻

5.4 Reachability in $\mathbb{Z}$-bounded nets is in PSPACE

Note that $\{\mathsf{i}\colon 0\} = \{\mathsf{f}\colon 0\} = \mathbf{0}$. We will use these notations interchangeably depending on the emphasis.

Lemma 5.9. *Let $\mathcal{N} = (P, T, F)$ be a nonredundant workflow net and $k > 0$. If $\mathcal{N}$ is $\mathbb{Z}$-unbounded from $\{\mathsf{i}\colon k\}$, then $\mathcal{N}$ is not generalised sound.*

Proof. Since $\mathcal{N}$ is $\mathbb{Z}$-unbounded from $\{\mathsf{i}\colon k\}$, there exist $\boldsymbol{m}, \boldsymbol{m}'$ and π such that $\boldsymbol{m} < \boldsymbol{m}'$ and $\{\mathsf{i}\colon k\} \to_{\mathbb{Z}}^* \boldsymbol{m} \to_{\mathbb{Z}}^{\pi} \boldsymbol{m}'$. Thus, $\{\mathsf{i}\colon 0\} \to_{\mathbb{Z}}^{\pi} \boldsymbol{m}' - \boldsymbol{m} > \mathbf{0}$. For the sake of contradiction, assume that $\mathcal{N}$ is generalised sound. It is strongly k-sound in particular for $k = 0$ by Theorem 5.6, so we have $\boldsymbol{m}' - \boldsymbol{m} \to^* \{\mathsf{f}\colon 0\}$, which contradicts the fact that $t^\bullet \neq \mathbf{0}$ for all $t \in T$. ◻

Lemma 5.10. *Let $\mathcal{N} = (P, T, F)$ be a workflow net. Let $\boldsymbol{m} \in \mathbb{N}^P$ be a marking such that $\|\boldsymbol{m}\| > \max(\|T\|, k)^2 \cdot (|P|+2) \cdot |P|$. If $\{\mathsf{i}\colon k\} \to_{\mathbb{Z}}^* \boldsymbol{m}$ then $\mathcal{N}$ is $\mathbb{Z}$-unbounded.*

Proof. Let $\{\mathsf{i}\colon k\} \to_{\mathbb{Z}}^{\sigma} \boldsymbol{m}$ for some $\sigma = t_1 t_2 \cdots t_n$. We use the notation $\wr \cdot \wr$ for multisets, *e.g.* $\wr a, a, b \wr$ contains two occurrences of a and one of b. Without loss of generality, assume that no submultiset of $\wr t_1, t_2, \ldots, t_n \wr$ sums to $\mathbf{0}$. Otherwise, we can shorten σ by removing such a submultiset. Further observe that since $\|\boldsymbol{m}\| > \max(\|T\|, k)^2 \cdot (|P| + 2) \cdot |P|$, we know that $n > \max(\|T\|, k) \cdot (|P| + 2) \cdot |P|$.

By Theorem 4.5, we can assume that $t_1, t_2, \ldots, t_n$ are ordered so that there exist $0 \le c_0 \le c_1 \le \ldots \le c_n$, where

$$\left\| \{\mathsf{i}\colon k\} + \sum_{j=1}^{i} \Delta(t_j) - c_i \boldsymbol{m} \right\| \le \max(\|T\|, k) \cdot (|P| + 2),$$

for all $i \in [0..n]$. By the pigeonhole principle, there must exist $0 \le i_1 < i_2 \le n$ such that

$$\{\mathsf{i}\colon k\} + \sum_{j=1}^{i_1} \Delta(t_j) - c_{i_1} \boldsymbol{m} = \{\mathsf{i}\colon k\} + \sum_{j=1}^{i_2} \Delta(t_j) - c_{i_2} \boldsymbol{m}.$$

This is equivalent to

$$\sum_{j=i_1+1}^{i_2} \Delta(t_j) = (c_{i_2} - c_{i_1}) \boldsymbol{m}.$$

We have $(c_{i_2} - c_{i_1}) \boldsymbol{m} \ge \mathbf{0}$ and, since no subset of $\wr t_1, t_2, \ldots, t_n \wr$ sums to $\mathbf{0}$, we have a strict inequality. Let $\boldsymbol{z} := \sum_{j=i_1+1}^{i_2} \Delta(t_j)$. We proved that $\{\mathsf{i}\colon 0\} \to_{\mathbb{Z}}^* \boldsymbol{z} > \mathbf{0}$, so $\mathcal{N}$ is $\mathbb{Z}$-unbounded. ◻

We are ready to prove the PSPACE membership of generalised soundness.

Proof of Theorem 5.1. Consider a workflow net $\mathcal{N} = (P, T, F)$. By Theorem 5.2, we can assume that $\mathcal{N}$ is a nonredundant workflow net. By Theorem 5.6 and Theorem 5.8, to prove generalised soundness it suffices to prove that it is k-sound for all $k \le \|\mathcal{N}\|^{\mathrm{poly}(|\mathcal{N}|)}$.

By Theorem 5.9 and Theorem 5.10, if $\{\mathsf{i}\colon k\} \to^* \boldsymbol{m}$ (and thus $\{\mathsf{i}\colon k\} \to_{\mathbb{Z}}^* \boldsymbol{m}$) and $\|\boldsymbol{m}\| \ge C_k$ for some $C_k = (\|\mathcal{N}\| + k)^{\mathrm{poly}(|\mathcal{N}|)}$, then the net is unsound. Since we need to consider only $k \le \|\mathcal{N}\|^{\mathrm{poly}(|\mathcal{N}|)}$, all constants C_k are bounded exponentially and can be written in polynomial space.

Thus, to verify k-soundness we proceed as follows. First, we check if a configuration $\boldsymbol{m}$ such that $\|\boldsymbol{m}\| \ge C_k$ can be reached. This can be easily performed in NPSPACE = PSPACE as such a run would be witnessed by a sequence of configurations, such that each configuration can be stored in polynomial space. If such a configuration can be reached, then the algorithm outputs no. Otherwise, for every $\boldsymbol{m} \in \mathbb{N}^P$ such that $\|\boldsymbol{m}\| < C_k$ one needs to verify whether $\{\mathsf{i}\colon k\} \to_{\mathbb{Z}}^* \boldsymbol{m}$ implies $\boldsymbol{m} \to^* \{\mathsf{f}\colon k\}$. This can be done in coNPSPACE = coPSPACE = PSPACE. ◻

5.5 PSPACE-hardness

A *conservative Petri net* is a Petri net $\mathcal{N} = (P, T, F)$ such that transitions preserve the number of tokens. That is, for all $\boldsymbol{m}, \boldsymbol{m}' \in \mathbb{N}^P$, it is the case that $\boldsymbol{m} \to \boldsymbol{m}'$ implies $\sum_{p \in P} \boldsymbol{m}[p] = \sum_{p \in P} \boldsymbol{m}'[p]$. The *reachability problem* for conservative Petri nets asks whether $\boldsymbol{m} \to^* \boldsymbol{m}'$, given $\mathcal{N}$, a source marking $\boldsymbol{m}$ and a target marking $\boldsymbol{m}'$.

Theorem 5.11. *Generalised soundness is PSPACE-hard.*

Proof. We give a reduction from reachability in conservative Petri nets, which is known to be PSPACE-complete [18].

Let $\mathcal{N} = (P, T, F)$ be a conservative Petri net, and let $\boldsymbol{m}, \boldsymbol{m}'$ be the source and target markings. We define the constant $c := \sum_{p \in P} \boldsymbol{m}[p] = \sum_{p \in P} \boldsymbol{m}'[p]$.

We construct a workflow net $\mathcal{N}' = (P', T', F')$ such that $\mathcal{N}'$ is generalised sound if and only if $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$ in $\mathcal{N}$. To do so, we extend $\mathcal{N}$ with three new places $P' := P \cup \{\mathsf{i}, \mathsf{f}, r\}$. Places i and f serve as dedicated initial and final places, respectively. Place r will be used to reset configurations. It could be merged with i, if not for the restriction that, in a workflow net, place i cannot have any incoming arc.

We define $T' \supseteq T$ by keeping the existing transitions and adding $3 + |P|$ new transitions. Namely:

1. transition t_i defined by ${}^\bullet t_\mathsf{i} := \{\mathsf{i}\colon 1\}$, and $t_\mathsf{i}^\bullet := \{r\colon c\}$,
2. transition $t_{\boldsymbol{m}}$ defined by ${}^\bullet t_{\boldsymbol{m}} := \{r\colon c\}$, and $t_{\boldsymbol{m}}^\bullet := \boldsymbol{m}$,
3. transition $t_{\boldsymbol{m}'}$ defined by ${}^\bullet t_{\boldsymbol{m}'} := \boldsymbol{m}'$, and $t_{\boldsymbol{m}'}^\bullet := \{\mathsf{f}\colon 1\}$,
4. transition t_p defined by ${}^\bullet t_p := \{p\colon 1\}$, and $t_p^\bullet := \{r\colon 1\}$.

The first two transitions move a token from i and create the marking $\boldsymbol{m}$. The third transition consumes $\boldsymbol{m}'$ and puts a token into f. Transitions from the fourth group allow to move tokens from any place in the original Petri net P to r. See Figure 6 for a graphical presentation.

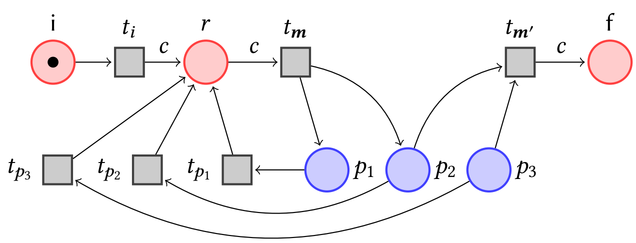

Figure 6. A workflow net $\mathcal{N}'$ which is generalised sound iff $m \rightarrow^* m'$ in the conservative Petri net $\mathcal{N} = (P, T, F)$. Here, $P = \{p_1, p_2, p_3\}$, $\boldsymbol{m} = \{p_1\colon 1, p_2\colon 1\}$, $\boldsymbol{m}' = \{p_2\colon 1, p_3\colon 1\}$ and $c = 2$. The original places are blue and the new places are red. We omit the original transitions (from T) in the picture.

It remains to show that $\mathcal{N}'$ is correct. Suppose $\mathcal{N}'$ is generalised sound. It must also be 1-sound and in particular $\{\mathsf{i}\colon 1\} \rightarrow^* \{\mathsf{f}\colon 1\}$. Since $\mathcal{N}$ is conservative, it is easy to see that $t_{\boldsymbol{m}}$ can be fired only if there are no tokens in P. Moreover, a token can be transferred to f only using $t_{\boldsymbol{m}'}$, which consumes $\boldsymbol{m}'$. Thus, we have $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$ in $\mathcal{N}$.

For the converse implication, suppose that $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$. Fix some k and suppose $\{\mathsf{i}\colon k\} \rightarrow^* \boldsymbol{v}$. Notice that the transitions are defined in such a way that for every reachable configuration $\boldsymbol{v}$, the invariant $ck = \boldsymbol{v}[\mathsf{i}] \cdot c + \sum_{p \in P \cup \{r\}} \boldsymbol{v}[p] + \boldsymbol{v}[\mathsf{f}] \cdot c$ holds. Thus, by repeatedly firing transitions t_i and t_p, all tokens but those in f can be moved to r, i.e.

$$\boldsymbol{v} \rightarrow^* \{r\colon (k - \boldsymbol{v}[\mathsf{f}]) \cdot c\} + \{\mathsf{f}\colon \boldsymbol{v}[\mathsf{f}]\}.$$

From there, to reach $\{\mathsf{f}\colon k\}$, it suffices to repeat $(k - \boldsymbol{v}[\mathsf{f}])$ times the following: fire $t_{\boldsymbol{m}}$; fire the run that witnesses $\boldsymbol{m} \rightarrow^* \boldsymbol{m}'$; and fire $t_{\boldsymbol{m}'}$. ◻

6 Structural soundness

In this section, we establish the EXPSPACE-completeness of structural soundness. Recall that the latter asks whether, given a workflow net, k-soundness holds for some $k \geq 1$.

6.1 EXPSPACE membership

Theorem 6.1. *Structural soundness is in EXPSPACE.*

Let $\mathcal{N} = (P, T, F)$ be a workflow net. We define an $(|T| + 2|P| + 1) \times (|T| + 1)$-ILP, called $\text{ILP}^s_{\mathcal{N}}$. The variables are the same as for $\text{ILP}_{\mathcal{N}}$ in subsection 5.2: $(\kappa, \tau_1, \ldots, \tau_n)$, with the intuition that κ denotes the number of initial tokens and τ_i the number of times the transitions are used. We will keep the notation $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_n)$ and the notation $\boldsymbol{\tau}[t]$ for $t \in T$. The inequalities are defined as follows:

1. $\{\mathsf{i}\colon \kappa\} + \sum_{t \in T} \boldsymbol{\tau}[t] \cdot \Delta(t) = \{\mathsf{f}\colon \kappa\}$ (expressed with $2|P|$ inequalities);
2. $\boldsymbol{\tau} \geq \mathbf{0}$ ($|T|$ inequalities);
3. and $\kappa > 0$.

The first set of inequalities expresses that the effect of the transitions yields the final marking. The second type ensures that each transition is fired a nonnegative number of times. Finally the last one ensures that the initial marking has at least one token. The following is immediate.

Claim 6.2. *There exists $k > 0$ such that $\{\mathsf{i}\colon k\} \rightarrow^*_{\mathbb{Z}} \{\mathsf{f}\colon k\}$ if and only if there exists a solution $\boldsymbol{\mu}$ to $\text{ILP}^s_{\mathcal{N}}$ such that $\boldsymbol{\mu}[\kappa] = k$.*

Lemma 6.3. *Let $\mathcal{N} = (P, T, F)$ be a nonredundant workflow net that is k-sound, and i-unsound for all $1 \leq i < k$. It is the case that $k \leq c + (\|T\| + 2)^{|T|} \cdot \max(\|T\|, c) \cdot |P|(|P| + 2)$, where c is the bound given by Theorem 4.3 for $\text{ILP}^s_{\mathcal{N}}$.*

Proof. Towards a contradiction, suppose that $k > c + (\|T\| + 2)^{|T|} \cdot \max(\|T\|, c) \cdot |P|(|P| + 2)$. Consider $\text{ILP}^s_{\mathcal{N}}$. Since $\mathcal{N}$ is k-sound, there is a run $\{\mathsf{i}\colon k\} \rightarrow^*_{\mathbb{Z}} \{\mathsf{f}\colon k\}$ and thus ILP_s has a solution $\boldsymbol{\mu}$. By Theorem 4.3, we can assume that $\boldsymbol{\mu} \leq \boldsymbol{c}$.

By Theorem 6.2, $\{\mathsf{i}\colon \boldsymbol{\mu}[\kappa]\} \rightarrow^*_{\mathbb{Z}} \{\mathsf{f}\colon \boldsymbol{\mu}[\kappa]\}$. By Theorem 5.5, there exist $\ell \leq (\|T\| + 2)^{|T|} \cdot \max(\|T\|, \boldsymbol{\mu}[\kappa]) \cdot |P|(|P| + 2)$ and $\boldsymbol{m} \in \mathbb{N}^P$ such that $\{\mathsf{i}\colon \ell\} \rightarrow^* \boldsymbol{m}$ and $\{\mathsf{i}\colon \ell + \boldsymbol{\mu}[\kappa]\} \rightarrow^* \boldsymbol{m} + \{\mathsf{f}\colon \boldsymbol{\mu}[\kappa]\}$. Note that $\ell + \boldsymbol{\mu}[\kappa] < k$. Let $g := k - (\ell + \boldsymbol{\mu}[\kappa]) > 0$. We have $\{\mathsf{i}\colon k\} = \{\mathsf{i}\colon \ell + \boldsymbol{\mu}[\kappa] + g\} \rightarrow^* \{\mathsf{i}\colon g\} + \boldsymbol{m} + \{\mathsf{f}\colon \boldsymbol{\mu}[\kappa]\}$. Since $\mathcal{N}$ is k-sound, we have

$$\{\mathsf{i}\colon g\} + \boldsymbol{m} + \{\mathsf{f}\colon \boldsymbol{\mu}[\kappa]\} \rightarrow^* \{\mathsf{f}\colon \ell + \boldsymbol{\mu}[\kappa] + g\}.$$

Thus, since ${}^\bullet t[\mathsf{f}] = 0$ for all $t \in T$, we have $\{\mathsf{i}\colon g\} + \boldsymbol{m} \rightarrow^* \{\mathsf{f}\colon \ell + g\}$. Altogether, obtain

$$\begin{aligned}\{\mathsf{i}\colon \ell + \boldsymbol{\mu}[\kappa] + g\} &\rightarrow^* \{\mathsf{i}\colon \boldsymbol{\mu}[\kappa] + g\} + \boldsymbol{m} \\ &\rightarrow^* \{\mathsf{i}\colon \boldsymbol{\mu}[\kappa]\} + \{\mathsf{f}\colon \ell + g\}.\end{aligned}$$

Therefore, since $\mathcal{N}$ is k-sound, it must be $\boldsymbol{\mu}[\kappa]$-sound (recall that tokens in f are never consumed). This contradicts the fact that $\mathcal{N}$ is i-unsound for all $1 \leq i < k$. ◻

We may now prove Theorem 6.1.

Proof of Theorem 6.1. By Theorem 5.2, we can assume that the input $\mathcal{N}$ is a nonredundant workflow net. By Theorem 6.3, it suffices to check if $\mathcal{N}$ is k-sound for some value k bounded by $\|\mathcal{N}\|^{\text{poly}|\mathcal{N}|}$. First, we guess k, which can be written with polynomially many bits. Then, we test k-soundness in EXPSPACE via Theorem 3.7. □

6.2 EXPSPACE-hardness

Theorem 6.4. *Structural soundness is EXPSPACE-hard.*

Proof. Let $\mathcal{N}$ be a workflow net. We construct a workflow net $\mathcal{N}'$ which is structurally sound iff $\mathcal{N}$ is 1-sound. We simply add a single new transition t to $\mathcal{N}$ with ${}^\bullet t := \{\mathsf{i}\colon 2\}$ and $t^\bullet := \{\mathsf{f}\colon 1\}$. We show that $\mathcal{N}'$ is k-unsound for every $k \geq 2$. Towards a contradiction, suppose it is k-sound for some $k \geq 2$.

Notice that k cannot be even because $\{\mathsf{i}\colon k\} \rightarrow^{t^{k/2}} \{\mathsf{f}\colon k/2\}$ and f has no outgoing arcs, and hence $\{\mathsf{f}\colon k/2\} \not\rightarrow^* \{\mathsf{f}\colon k\}$. Thus, it is the case that $k \geq 3$ is odd and $\{\mathsf{i}\colon k\} \rightarrow^{t^*} \{\mathsf{i}\colon 1\} + \{\mathsf{f}\colon \lfloor k/2 \rfloor\}$. Since $\mathcal{N}'$ is k-sound, $\{\mathsf{i}\colon 1\} \rightarrow^* \{\mathsf{f}\colon \lceil k/2 \rceil\}$. But that implies $\{\mathsf{i}\colon k\} \rightarrow^* \{\mathsf{f}\colon k \cdot \lceil k/2 \rceil\}$. Note that $k \cdot \lceil k/2 \rceil > k$ as $k \geq 3$, which yields a contradiction since f has no outgoing arcs to get rid of the extra tokens.

To conclude, we observe that if the initial configuration in $\mathcal{N}'$ is $\{\mathsf{i}\colon 1\}$, then it behaves like $\mathcal{N}$ would, since t will never be enabled, *i.e.* it is not quasi-live. Thus, $\mathcal{N}'$ is structurally sound if and only if $\mathcal{N}$ is 1-sound, and EXPSPACE-hardness follows from Theorem 3.10. □

7 Characterizing the set of sound numbers

Given a workflow net $\mathcal{N}$, we define the set $\text{Sound}(\mathcal{N}) := \{k \geq 1 \mid \mathcal{N} \text{ is } k\text{-sound}\}$. That is, Sound($\mathcal{N}$) contains all the numbers for which $\mathcal{N}$ is sound (except 0 which is trivial as any workflow net is 0-sound). This section is dedicated to providing and computing a representation of Sound($\mathcal{N}$).

First, let us state a simple fact about Sound($\mathcal{N}$).[4]

Lemma 7.1. *The set Sound($\mathcal{N}$) is closed under subtraction with positive results.*

Proof. Let $g, k \in \text{Sound}(\mathcal{N})$ be such that $g > k$. We show that $g - k \in \text{Sound}(\mathcal{N})$. Since $k \in \text{Sound}(\mathcal{N})$, we have $\{\mathsf{i}\colon g\} = \{\mathsf{i}\colon k + (g-k)\} \rightarrow^* \{\mathsf{f}\colon k\} + \{\mathsf{i}\colon g-k\}$. Since N is g-sound, it must also be $(g-k)$-sound. So, $g - k \in \text{Sound}(\mathcal{N})$. □

Corollary 7.2. *There exist $p > 0$ and $k \in \mathbb{N} \cup \{+\infty\}$ such that Sound($\mathcal{N}$) $= \{i \cdot p \mid 1 \leq i < k\}$.*

By the above, Sound($\mathcal{N}$) is characterized by p and k. We thus say that a net is *(k,p)-sound* if and only if $\text{Sound}(\mathcal{N}) = \{i \cdot p \mid 1 \leq i < k\}$. Note that $k = 0$ implies $\text{Sound}(\mathcal{N}) = \emptyset$. Further, $k = +\infty$ if and only if Sound($\mathcal{N}$) is infinite. Finally, a workflow net is generalised sound iff it is $(1, +\infty)$-sound; and it is structurally sound iff there exist $p, k \geq 1$ such that it is (k,p)-sound. We show that k and p can be computed. This will rely on insights from the prior sections about the smallest numbers for which a net is unsound or sound.

[4] A similar observation was made, but not explicitly stated, in [7, Lemma 2.2 and 2.3].

Theorem 7.3. *Given a workflow net $\mathcal{N}$, the numbers p and k that characterize Sound($\mathcal{N}$) are bounded by $\|\mathcal{N}\|^{\text{poly}\, O(|\mathcal{N}|)}$, and hence can be represented with polynomially many bits. Given $\mathcal{N}$, p' and k', the problem of deciding whether $\mathcal{N}$ is (k',p')-sound is in EXPSPACE. Moreover, the algorithm computes p and k such that $\mathcal{N}$ is (k,p)-sound.*

Proof. Consider a workflow net $\mathcal{N}$. By Theorem 5.2, we can assume that $\mathcal{N}$ is nonredundant. We will compute for which p and k the net $\mathcal{N}$ is (k,p)-sound. By Theorem 6.3, if $\text{Sound}(\mathcal{N}) \neq \emptyset$, then there exists $G \leq \|\mathcal{N}\|^{\text{poly}|\mathcal{N}|}$ such that $\mathcal{N}$ is ℓ-sound for some $\ell \leq G$. By Theorem 3.7, it is possible to check 1-soundness, 2-soundness, …, G-soundness in EXPSPACE. Thus, in EXPSPACE, we can identify the smallest p such that $\mathcal{N}$ is p-sound.

It remains to compute k. Using Theorem 3.6, we construct a net $\mathcal{N}'$ which is c-sound if and only if $\mathcal{N}$ is cp-sound for all $c > 0$. Thus, the smallest number c for which $\mathcal{N}'$ is not c-sound is the smallest c such that $\mathcal{N}$ is not cp-sound. By Theorem 5.8, if $\text{Sound}(\mathcal{N}') \neq \mathbb{N} \setminus \{0\}$ then there exists $G' \leq \|\mathcal{N}\|^{\text{poly}\, O(|\mathcal{N}|)}$ such that $\mathcal{N}'$ is c-unsound for some $c \leq G'$. Thus, it suffices to check 1-soundness, 2-soundness, …, G'-soundness to identify whether $k = +\infty$, or to compute the largest $k \in \mathbb{N}$ such that $\mathcal{N}$ is pk-sound. By Theorem 3.7, k can be computed in exponential space. □

8 Conclusion

In this work, we settled, after around two decades, the complexity of the main decision problems concerning workflow nets: k-soundness, classical soundness, structural soundness, and generalised soundness. The first three are EXPSPACE-complete, while the latter is PSPACE-complete and hence surprisingly simpler. We have further characterised the set of sound numbers of workflow nets: they have a specific shape that can be computed with exponential space.

As further work, we intend to study extensions of these problems in the context of Petri nets. For example, a natural extension of generalised soundness asks, given markings $\boldsymbol{m}$ and $\boldsymbol{m}'$, whether for every $k \in \mathbb{N}$, every marking reachable from $k \cdot \boldsymbol{m}$ can reach $k \cdot \boldsymbol{m}'$. Contrary to workflow nets, a Petri net that satisfies this property needs not to be bounded.

References

[1] Kamel Barkaoui and Laure Petrucci. 1998. Structural analysis of workflow nets with shared resources. In *Proc. Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM)*, Vol. 98/7. 82–95.

[2] Michael Blondin. 2020. The ABCs of Petri net reachability relaxations. *ACM SIGLOG News* 7, 3 (2020). https://doi.org/10.1145/3436980.3436984

[3] Zakaria Bouziane and Alain Finkel. 1997. Cyclic Petri net reachability sets are semi-linear effectively constructible. In *Second International Workshop on Verification of Infinite State Systems (INFINITY) (Electronic*

Notes in Theoretical Computer Science), Vol. 9. 15–24. https://doi.org/10.1016/S1571-0661(05)80423-2

[4] E. Cardoza, Richard J. Lipton, and Albert R. Meyer. 1976. Exponential Space Complete Problems for Petri Nets and Commutative Semigroups: Preliminary Report. In *Proc. 8th Annual ACM Symposium on Theory of Computing (STOC)*. 50–54. https://doi.org/10.1145/800113.803630

[5] Allan Cheng, Javier Esparza, and Jens Palsberg. 1993. Complexity Results for 1-safe Nets. In *Proc. 13th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, Vol. 761. 326–337. https://doi.org/10.1007/3-540-57529-4_66

[6] Dmitry Chistikov and Christoph Haase. 2016. The Taming of the Semi-Linear Set. In *Proc. 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, Vol. 55. 128:1–128:13. https://doi.org/10.4230/LIPIcs.ICALP.2016.128

[7] Ferucio Laurenţiu Ţiplea and Dan Cristian Marinescu. 2005. Structural soundness of workflow nets is decidable. *Inform. Process. Lett.* 96, 2 (2005), 54–58. https://doi.org/10.1016/j.ipl.2005.06.002

[8] Wojciech Czerwinski and Lukasz Orlikowski. 2021. Reachability in Vector Addition Systems is Ackermann-complete. In *Proc. 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. to appear.

[9] Wil Van der Aalst. 1999. Interorganizational Workflows: An Approach Based on Message Sequence Charts and Petri Nets. *Systems Analysis, Modelling, Simulation* 34, 3 (1999), 335–367.

[10] Jörg Desel and Javier Esparza. 1995. *Free Choice Petri Nets*. Cambridge University Press. https://doi.org/10.1017/CBO9780511526558

[11] Friedrich Eisenbrand and Robert Weismantel. 2019. Proximity results and faster algorithms for integer programming using the Steinitz lemma. *ACM Transactions on Algorithms (TALG)* 16, 5 (2019), 5:1–5:14. https://doi.org/10.1145/3340322

[12] Christoph Haase and Simon Halfon. 2014. Integer Vector Addition Systems with States. In *Proc. 8th International Workshop on Reachability Problems (RP) 2014*, Vol. 8762. 112–124. https://doi.org/10.1007/978-3-319-11439-2_9

[13] Michel Henri Théodore Hack. 1976. *Decidability questions for Petri Nets*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[14] Jérôme Leroux. 2021. The Reachability Problem for Petri Nets is Not Primitive Recursive. In *Proc. 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. to appear.

[15] Jérôme Leroux and Sylvain Schmitz. 2019. Reachability in Vector Addition Systems is Primitive-Recursive in Fixed Dimension. In *Proc. 34th Symposium on Logic in Computer Science (LICS)*.

[16] Ernst W. Mayr. 1981. An Algorithm for the General Petri Net Reachability Problem. In *Proc. 13th Symposium on Theory of Computing (STOC)*. 238–246. https://doi.org/10.1145/800076.802477

[17] Ernst W Mayr and Albert R Meyer. 1982. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in Mathematics* 46, 3 (1982), 305–329. https://doi.org/10.1016/0001-8708(82)90048-2

[18] Ernst W. Mayr and Jeremias Weihmann. 2014. A Framework for Classical Petri Net Problems: Conservative Petri Nets as an Application. In *Proc. 35th International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS)*. 314–333. https://doi.org/10.1007/978-3-319-07734-5_17

[19] Charles Rackoff. 1978. The Covering and Boundedness Problems for Vector Addition Systems. *Theoretical Computer Science* 6 (1978), 223–231. https://doi.org/10.1016/0304-3975(78)90036-1

[20] Ernst Steinitz. 1913. Bedingt konvergente Reihen und konvexe Systeme. (1913).

[21] Wil MP Van der Aalst. 1996. Structural characterizations of sound workflow nets. *Computing science reports* 96, 23 (1996), 18–22.

[22] Wil M. P. van der Aalst. 1997. Verification of Workflow Nets. In *Proc. 18th International Conference on Application and Theory of Petri Nets (ICATPN)*, Vol. 1248. 407–426. https://doi.org/10.1007/3-540-63139-9_48

[23] Wil M. P. van der Aalst. 1998. The Application of Petri Nets to Workflow Management. *Journal of Circuits, Systems, and Computers* 8, 1 (1998), 21–66. https://doi.org/10.1142/S0218126698000043

[24] Wil M. P. van der Aalst and Christian Stahl. 2011. *Modeling Business Processes - A Petri Net-Oriented Approach*. MIT Press.

[25] Wil M. P. van der Aalst, Kees M. van Hee, Arthur H. M. ter Hofstede, Natalia Sidorova, H. M. W. Verbeek, Marc Voorhoeve, and Moe Thandar Wynn. 2011. Soundness of workflow nets: classification, decidability, and analysis. *Formal Aspects of Computing* 23, 3 (2011), 333–363. https://doi.org/10.1007/s00165-010-0161-4

[26] Boudewijn F. van Dongen, Ana Karla A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and Wil M. P. van der Aalst. 2005. The ProM Framework: A New Era in Process Mining Tool Support. In *Proc. 26th International Conference on Applications and Theory of Petri Nets (ICATPN)*, Vol. 3536. 444–454. https://doi.org/10.1007/11494744_25

[27] Kees M. van Hee, Olivia Oanea, Natalia Sidorova, and Marc Voorhoeve. 2006. Verifying Generalized Soundness of Workflow Nets. In *Proc. 6th International Andrei Ershov Memorial Conference on Perspectives of Systems Informatics (PSI)*. 235–247. https://doi.org/10.1007/978-3-540-70881-0_21

[28] Kees M. van Hee, Natalia Sidorova, and Marc Voorhoeve. 2003. Soundness and Separability of Workflow Nets in the Stepwise Refinement Approach. In *Proc. 24th International Conference on Applications and Theory of Petri Nets 2003 (ICATPN)*, Vol. 2679. 337–356. https://doi.org/10.1007/3-540-44919-1_22

[29] Kees M. van Hee, Natalia Sidorova, and Marc Voorhoeve. 2004. Generalised Soundness of Workflow Nets Is Decidable. In *Proc. 25th International Conference on Applications and Theory of Petri Nets (ICATPN)*. 197–215. https://doi.org/10.1007/978-3-540-27793-4_12

[30] Joachim von zur Gathen and Malte Sieveking. 1978. A Bound on Solutions of Linear Integer Equalities and Inequalities. *Proc. Amer. Math. Soc.* 72, 1 (1978), 155–158.