Soundness of reset workflow nets

Michael Blondin[*]
michael.blondin@usherbrooke.ca
Université de Sherbrooke
Sherbrooke, Québec, Canada

Alain Finkel
alain.finkel@ens-paris-saclay.fr
Université Paris-Saclay, CNRS, ENS
Paris-Saclay, IUF
Gif-sur-Yvette, France

Piotr Hofman
piotr.hofman@uw.edu.pl
University of Warsaw
Warsaw, Poland

Filip Mazowiecki[†]
f.mazowiecki@mimuw.edu.pl
University of Warsaw
Warsaw, Poland

Philip Offtermatt
philip.offtermatt@informal.systems
Informal Systems
Lausanne, Switzerland

ABSTRACT

Workflow nets are a well-established variant of Petri nets for the modeling of process activities such as business processes. The standard correctness notion of workflow nets is soundness, which comes in several variants. Their decidability was shown decades ago, but their complexity was only identified recently. In this work, we are primarily interested in two popular variants: 1-soundness and generalised soundness.

Workflow nets have been extended with resets to model workflows that can, e.g., cancel actions. It has been known for a while that, for this extension, all variants of soundness, except possibly generalised soundness, are undecidable.

We complete the picture by showing that generalised soundness is also undecidable for reset workflow nets. We then blur this undecidability landscape by identifying a property, coined "1-in-between soundness", which lies between 1-soundness and generalised soundness. It reveals an unusual non-monotonic complexity behaviour: a decidable soundness property is in between two undecidable ones. This can be valuable in the algorithmic analysis of reset workflow nets, as our procedure yields an output of the form "1-sound" or "not generalised sound" which is always correct.

CCS CONCEPTS

• **Theory of computation → Automata over infinite objects**; **Computational complexity and cryptography**.

KEYWORDS

Workflow nets, Petri nets, resets, soundness, generalised soundness, decidability

ACM Reference Format:
Michael Blondin, Alain Finkel, Piotr Hofman, Filip Mazowiecki, and Philip Offtermatt. 2024. Soundness of reset workflow nets. In *39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '24), July 8–11, 2024, Tallinn, Estonia.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3661814.3662086

1 INTRODUCTION

Workflow nets are a well-established formalism for the modeling of process activities such as business processes [30]. For example, they can be used as the formal representation of workflow procedures in business process management systems (see e.g. [31, Section 4] and [30, Section 3] for details on modeling procedures). Workflow nets enable the algorithmic formal analysis of their behaviour. This is relevant, e.g., for organizations that seek to manage complex business processes. For example, according to a survey [33], over 20% instances from the SAP reference model have been detected to be flawed (due to deadlocks, livelocks, etc.)

A workflow net $\mathcal{W}$ is essentially a Petri net — a prominent model of concurrency — that satisfies extra properties. In particular, $\mathcal{W}$ has two designated places i and f respectively called *initial* and *final*. Initially, $\mathcal{W}$ starts with k tokens in place i that can evolve according to the transitions of $\mathcal{W}$, which can consume and create tokens. Informally, a token that reaches f indicates that some activity has been completed.

1.1 Soundness

The standard correctness notion of workflow nets is soundness. Various definitions have been considered in the literature. Most prominently, *k-soundness* requires that, starting from k tokens in the initial place, no matter what transitions are taken, it is always possible to complete properly, i.e. to end up with k tokens in the final place, and no token elsewhere. *Generalised soundness* requires a net to be k-sound for all $k > 0$, while *structural soundness* requires k-soundness for some $k > 0$. *Classical soundness* requires 1-soundness and each transition to be fireable in at least one execution.

The decidability of soundness was established some two decades ago [5, 30, 35, 36] (see [33] for a survey). The underlying algorithms relied on Petri net reachability, which was recently shown to be non-primitive recursive [7, 20, 21]. Until recently, no better bound was known. At LICS'22 [3], the computational complexity of all variants of soundness was established: generalised soundness is PSPACE-complete, while the other variants are EXPSPACE-complete.

[*] Supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (NSERC), and by the Fonds de recherche du Québec – Nature et technologies (FRQNT).
[†] Supported by Polish National Science Centre SONATA BIS-12 grant number 2022/46/E/ST6/00230.

LICS '24, July 8–11, 2024, Tallinn, Estonia

ACM ISBN 979-8-4007-0660-8/24/07
https://doi.org/10.1145/3661814.3662086

One shortcoming of workflow nets is that they lack useful features like cancellation. As mentioned in [29], "[m]any practical languages have a cancelation feature, e.g., Staffware has a withdraw construct, YAWL has a cancelation region, BPMN has cancel, compensate, and error events, etc." Thus, workflow nets have been extended with *resets* that instantly remove all tokens from a region of the net upon taking a transition (e.g. see [32]). Around fifteen years ago, it was shown that, for reset workflow nets,

> *[... of] the many soundness notions described in [the] literature only generalised soundness may be decidable (this is still an open problem). All other notions are shown to be undecidable.* [29]

So, while all studied variants such as 1-soundness, k-soundness, structural soundness and classical soundness are undecidable for reset workflow nets, the decidability of generalised soundness has remained open.

The recent results of [3] shows that in the absence of resets, generalised soundness is computationally easier than other soundness variants. This was an indication that generalised soundness could remain decidable in the presence of resets.

1.2 Our contribution

In this work, we first show that such a conjecture is false: generalised soundness for reset workflow nets is undecidable.

The undecidability of the various types of soundness in reset workflow nets indicates the necessity for a new approach. Consequently, we embarked on the journey of exploring potential methods for "approximating" soundness. We propose a precise definition of an acceptable approximation.

More precisely, we say that a property $\mathcal{P}$ of reset workflow nets is a 1-*in-between soundness* property if it meets the following criteria: all generalised sound reset workflow nets satisfy $\mathcal{P}$, and every reset workflow net satisfying $\mathcal{P}$ is 1-sound. Remarkably, we have discovered such a property, denoted as $\mathcal{P}_1$, which is decidable. Thus, we have

$$\text{Generalised sound nets} \subseteq \mathcal{P}_1 \text{ nets} \subseteq \text{1-sound nets},$$

where generalised soundness and 1-soundness are undecidable, but $\mathcal{P}_1$ is decidable.

This reveals a non-monotonic complexity behaviour. By this, we mean three properties A, B and C such that

$$A \text{ (hard)} \subseteq B \text{ (easy)} \subseteq C \text{ (hard)}.$$

This phenomenon can be regarded as unnatural and is rarely actively exploited. We present two other such examples.

First, consider the setting where we are given two graphs, each with a designated initial node and with directed edges labelled from a common finite alphabet (so, finite automata whose states are all accepting). These inclusions hold:

$$\text{Isomorphism} \subseteq \text{Bisimilarity} \subseteq \text{Trace(-language) equivalence},$$

The first property is known to be checkable in quasi-polynomial time (a survey on current advances can be found in [14]), the second is checkable in polynomial time [25], and the third one is well known to be PSPACE-complete.

Now, consider the setting where we are given two one-counter nets (OCN), where an OCN is an automaton with a single counter over the naturals that can be incremented and decremented (but not zero-tested). These inclusions hold:

$$\begin{aligned}&\text{Weak bisimulation [24]} \subseteq\\ &\text{Weak simulation in both directions [16]} \subseteq\\ &\text{Language equivalence [28]},\end{aligned}$$

The first and the last properties are undecidable, while the middle one is PSPACE-complete.

Our property $\mathcal{P}_1$ can be valuable in the analysis of reset workflow nets. For example, we provide an algorithm that, on any reset workflow net which is not 1-sound, classifies it as not generalised sound, and on any generalised sound reset workflow net, guarantees at least 1-soundness. For reset workflow nets that are 1-sound but not generalised sound, the algorithm provides the correct description of either not being generalised sound or being 1-sound. It is worth noting that all these answers accurately characterize the given reset workflow net.

However, the computational complexity of verifying $\mathcal{P}_1$ is non-primitive recursive in the worst case, which hinders immediate applications. Nevertheless, we cannot rule out the existence of similar predicates with better computational complexity, or of implementations performing faster on real-world instances. This area requires further investigation.

The definition of $\mathcal{P}_1$ is exceedingly technical, making it challenging to convey basic intuitions concisely. We actually define a decidable family of properties $\{\mathcal{P}_k\}_{k>0}$. In our final result, we prove a connection with another notion of soundness found in the literature, namely *up-to-k soundness* [34]. Specifically, we demonstrate that for every reset workflow net $\mathcal{W}$, there is a computable but substantially large value K such that, for every $k > K$, $\mathcal{W}$ is up-to-k sound if and only if it satisfies the property $\mathcal{P}_k$.

1.3 Further related work

A significant portion of the work that relates to reset workflow nets consists of results on reset Petri nets, in particular, on

- The theory of well-structured transition systems and results for the general class of reset Petri nets [8, 12];
- Results on restricted subclasses, such as those with a limited number of places that can be reset [11], acyclic reset Petri nets and workflow nets [4], or Petri nets with hierarchical reset arcs [1];
- More practically oriented results on relaxations of the reachability relation like the integer relaxation [15].

An important branch of soundness-related research includes work on reduction rules that preserve (un)soundness while reducing the size and structure of the workflow net. Rules specific to reset workflow nets are explored in [37, 38].

Another line of research lies in the field of process discovery, e.g. the discovery of cancellation regions within process mining techniques [17].

1.4 Paper organization

The paper is organized as follows. In Section 2, we introduce preliminary definitions such as Petri nets, workflow nets and notions of soundness. In Section 3, we establish the undecidability of generalised soundness for reset workflow nets. Section 4 shows the existence of the k-in-between soundness property $\mathcal{P}_k$, and relates it with another notion of soundness. In particular, Section 4.1 introduces the intermediate notion of "nonredundancy", and Section 4.2 introduces the intermediate notion of "skeleton" workflow nets.

2 PRELIMINARIES

Let $\mathbb{N} \coloneqq \{0, 1, \ldots\}$ and $\mathbb{N}_{>0} \coloneqq \mathbb{N} \setminus \{0\}$. For every $a, b \in \mathbb{N}$ such that $a \leq b$, we define $[a..b] \coloneqq \{a, a+1, \ldots, b\}$. Given a set X, we denote its cardinality by $|X|$.

Let $\boldsymbol{m}, \boldsymbol{m'} : P \to \mathbb{N}$ where P is a finite set. We see $\boldsymbol{m}$ and $\boldsymbol{m'}$ as both unordered vectors and multisets. For example, we have $\boldsymbol{m} = \boldsymbol{0}$ if $\boldsymbol{m}(p) = 0$ for every $p \in P$, and we have $\boldsymbol{m'} = \{p\colon 1, q\colon 3\}$ if $\boldsymbol{m'}(p) = 1$, $\boldsymbol{m'}(q) = 3$ and $\boldsymbol{m'}(r) = 0$ for every $r \in P \setminus \{p, q\}$. We write $\boldsymbol{m} \leq \boldsymbol{m'}$ iff $\boldsymbol{m}(p) \leq \boldsymbol{m'}(p)$ for every $p \in P$. We write $\boldsymbol{m} < \boldsymbol{m'}$ if $\boldsymbol{m} \leq \boldsymbol{m'}$ and $\boldsymbol{m} \neq \boldsymbol{m'}$. We define $\boldsymbol{m}+\boldsymbol{m'}$ as the mapping satisfying $(\boldsymbol{m}+\boldsymbol{m'})(p) = \boldsymbol{m}(p)+\boldsymbol{m'}(p)$ for every $p \in P$. The mapping $\boldsymbol{m}-\boldsymbol{m'}$ is defined similarly, provided that $\boldsymbol{m} \geq \boldsymbol{m'}$. For every $Q \subseteq P$, we define $\boldsymbol{m}(Q) \coloneqq \sum_{q \in Q} \boldsymbol{m}(q)$.

2.1 Ackermannian complexity

In the upcoming sections, we prove theorems about the existence of some numbers bounded by functions on the size of the input. To do this rigorously, we should refer to these individual bounds and precisely track dependencies between them. However, some of the bounds we use are based on the Ackermann function (which is non-primitive recursive), and tracking them precisely is tedious. Moreover, keeping all the constants in mind would be troublesome for the reader. That is why we choose to simply state that some number is of "Ackermannian size" or"Ackermannianly bounded". In our reasoning, we use the following rules:

- The maximum, sum and product of Ackermannianly bounded constants yields Ackermannianly bounded constants;
- The application of a function within the Ackermannian class of functions to a bound of Ackermannian size yields an Ackermannianly bounded constant.

Since we will make a constant number of such manipulations, the Ackermannian bounds will be preserved.

For an introduction to high computational complexity classes (from the fast-growing hierarchy), we refer the reader to [23, 27].

2.2 Petri nets

A *reset Petri net* $\mathcal{N}$ is a tuple (P, T, F, R) where

- P is a finite set of elements called *places*,
- T is a finite set, disjoint from P, of elements called *transitions*,
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of elements called *arcs*,
- $R \subseteq (P \times T)$ is a set of elements called *reset arcs.*

A *(standard) Petri net* is a reset Petri net with no reset arc ($R = \emptyset$).

Example 2.1. Figure 1 depicts a reset Petri net $\mathcal{N} = (P, T, F, R)$ where $P = \{p_1, p_2, p_3, p_4\}$ (circles), $T = \{t_1, t_2, t_3\}$ (boxes), each arc $(u, v) \in F$ is depicted by a directed edge, and the only reset arc is depicted by a dotted directed edge. □

Given a transition $t \in T$, we define ${}^\bullet t \coloneqq \{p \in P : (p, t) \in F\}$ and $t^\bullet \coloneqq \{p \in P : (t, p) \in F\}$. Both of these sets will often be interpreted implicitly as mappings from P to $\{0, 1\}$. For example, we can write either "$p \in t^\bullet$" or "$t^\bullet(p) = 1$". In Figure 1, we have, e.g., ${}^\bullet t_1 = \{p_1\}$, $t_1{}^\bullet = \{p_2, p_3\}$ and ${}^\bullet t_3 = \{p_2\}$. Given a place $p \in P$, we define ${}^\bullet p \coloneqq \{t \in T : (t, p) \in F\}$ and $p^\bullet \coloneqq \{t \in T : (p, t) \in F\}$.

A *marking* is a mapping $\boldsymbol{m}\colon P \to \mathbb{N}$ that indicates the number of *tokens* in each place. Given a marking $\boldsymbol{m}$ and a transition t, we let $Reset_t(\boldsymbol{m})$ denote the marking obtained from $\boldsymbol{m}$ by emptying all places that are reset by t. Formally, $Reset_t(\boldsymbol{m}) \coloneqq 0$ if $(p, t) \in R$, and $\boldsymbol{m}(p)$ otherwise.

We say that a transition $t \in T$ is *enabled* in marking $\boldsymbol{m}$ if $\boldsymbol{m} \geq {}^\bullet t$, i.e. if each place of ${}^\bullet t$ contains at least one token. If t is enabled in $\boldsymbol{m}$, then it can be *fired.* In words, upon firing t, a token is consumed from each place of ${}^\bullet t$; then, all places of $\{p : (p, t) \in R\}$ are emptied; and, finally, a token is produced in each place of $t^\bullet$. More formally, firing t leads to the marking $\boldsymbol{m'}$ defined as follows, for every $p \in P$:

$$\boldsymbol{m'}(p) = \begin{cases} \boldsymbol{m}(p) - {}^\bullet t(p) + t^\bullet(p) & \text{if } (p, t) \notin R, \\ t^\bullet(p) & \text{otherwise.} \end{cases}$$

Equivalently, and more succinctly, $\boldsymbol{m'} = Reset_t(\boldsymbol{m} - {}^\bullet t) + t^\bullet$.

We write $\boldsymbol{m} \to^{t} \boldsymbol{m'}$ whenever t is enabled in $\boldsymbol{m}$ and firing t from $\boldsymbol{m}$ leads to $\boldsymbol{m'}$. We write $\boldsymbol{m} \to \boldsymbol{m'}$ if $\boldsymbol{m} \to^{t} \boldsymbol{m'}$ holds for some $t \in T$. We write $\to^*$ to denote the reflexive-transitive closure of $\to$. Given a subset X of markings, we write $\boldsymbol{m} \to^* X$ to denote that there exists $\boldsymbol{m'} \in X$ such that $\boldsymbol{m} \to^* \boldsymbol{m'}$.

Given a sequence of transitions $\pi = t_1 t_2 \cdots t_n$ and a transition s, we define $|\pi| \coloneqq n$ and $|\pi|_s = |\{i \in [1..n] : t_i = s\}|$.

Example 2.2. Reconsider the reset Petri net of Figure 1. For the sake of brevity, let us write a marking $\{p_1\colon a, p_2\colon b, p_3\colon c, p_4\colon d\}$ as (a, b, c, d). From marking $(1, 0, 0, 0)$, we can only fire t_1. Firing t_1 leads to $(0, 1, 1, 0)$. From the latter, we can fire either t_2 or t_3. Firing t_2 leads to $(1, 0, 1, 0)$. From there, we can only fire t_1, which leads to $(0, 1, 2, 0)$. From the latter, we can fire either t_2 or t_3. Firing t_3 leads to $(0, 0, 0, 1)$, from which no transition is enabled. The described sequence can be written as

$$\begin{aligned}(1,0,0,0) \to^{t_1} (0,1,1,0) \to^{t_2} (1,0,1,0) & \\ \to^{t_1} (0,1,2,0) \to^{t_3} (0,0,0,1).&\end{aligned}$$

Thus, we have $(1, 0, 0, 0) \to^{t_1 t_2 t_1 t_3} (0, 0, 0, 1)$, or more succinctly $(1, 0, 0, 0) \to^* (0, 0, 0, 1)$. □

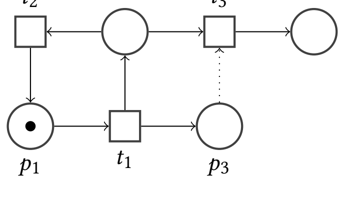


Figure 1: Example of a reset Petri net where circles are places, boxes are transitions, solid edges are arcs, and dotted edges are reset arcs. The small filled circle depicts a token.

2.3 Upward and downward closed sets

The set of markings $\mathbb{N}^P$ is partially ordered by $\leq$. The latter is a well-quasi-order, which means that it neither contains infinite antichains nor infinite (strictly) decreasing sequences.

A set of markings $X \subseteq \mathbb{N}^P$ is *upward closed* if for every $\boldsymbol{m} \in X$ and every other marking $\boldsymbol{m'} \in \mathbb{N}^P$, it is the case that $\boldsymbol{m} \leq \boldsymbol{m'}$ implies $\boldsymbol{m'} \in X$. The *upward closure* of $Y \subseteq \mathbb{N}^P$ is defined as the smallest upward closed set, denoted $\uparrow Y$, such that $Y \subseteq \uparrow Y$.

Similarly, a subset $X \subseteq \mathbb{N}^P$ is *downward closed* if for every $\boldsymbol{m} \in X$ and every other marking $\boldsymbol{m'} \in \mathbb{N}^P$, we have $\boldsymbol{m'} \leq \boldsymbol{m} \implies \boldsymbol{m'} \in X$. The *downward closure* of $Y \subseteq \mathbb{N}^P$, denoted $\downarrow Y$, is the smallest downward closed set such that $Y \subseteq \downarrow Y$.

We extend upward and downward closures to individual markings, i.e. $\uparrow \boldsymbol{m} := \uparrow \{\boldsymbol{m}\}$ and $\downarrow \boldsymbol{m} := \downarrow \{\boldsymbol{m}\}$.

It is worth noting that every upward closed set is characterized by the set of its minimal elements. Since such a set is an antichain, it must be finite, resulting in a finite representation for each upward closed set. Similarly, every downward closed set can be finitely represented by a finite representation of its complement, which is an upward closed set.

Given a reset Petri net $\mathcal{N}$, we say that from a marking $\boldsymbol{m}$, it is possible to *cover* a marking $\boldsymbol{m'}$ if there exists a marking $\boldsymbol{m''} \geq \boldsymbol{m'}$ such that $\boldsymbol{m} \rightarrow^* \boldsymbol{m''}$ holds in $\mathcal{N}$. Let $C_{\boldsymbol{m'}}$ be the set of all markings $\boldsymbol{m}$ from which it is possible to cover a marking $\boldsymbol{m'}$. Observe that $C_{\boldsymbol{m'}}$ is upward closed. An important result is that, for any given marking $\boldsymbol{m'}$, it is possible to compute the minimal elements of the set $C_{\boldsymbol{m'}}$. This computation can be achieved using the so-called "backward coverability algorithm" [13].

Additionally, it is crucial to note the following lemma.

Lemma 2.3 ([9]). *If it is possible to cover $\boldsymbol{m'}$ from $\boldsymbol{m}$, then it can be done with a run whose length is Ackermannianly bounded in the size of the Petri net and $\boldsymbol{m'}$.*

Throughout the paper we will often use Lemma 2.3 without referring to it. We will even use a stronger property that the set of minimal elements of $C_{\boldsymbol{m'}}$ can be computed in Ackermannian time. This follows from [19], where it is proved that the backward coverability algorithm works in Ackermannian time.

2.4 Workflow nets and soundness

A *reset workflow net* is a tuple (P, T, F, R, i, f) where

- $\mathcal{N} = (P, T, F, R)$ is a reset Petri net,
- $i \in P$ is a place called *initial* that satisfies ${}^\bullet i = \emptyset$,
- $f \in P$ is a place called *final* that satisfies $f^\bullet = \emptyset$, and $(f, t) \notin R$ for all $t \in T$ (no transition resets f),
- each element of $P \cup T$ is on some path from i to f in the underlying graph of $\mathcal{N}$ without considering reset arcs, i.e. in the graph $G := (V, E)$ with vertices $V := P \cup T$ and directed edges $E := \{(u, v) \in V \times V : (u, v) \in F\}$.

A *(standard) workflow net* is a reset workflow net with no reset arc, i.e. with $R = \emptyset$. For example, Figure 2 depicts a reset workflow net.

Given $k \in \mathbb{N}_{>0}$, we say that a reset workflow net $\mathcal{W}$ is *k-sound* if for every marking $\boldsymbol{m}$ such that $\{i\colon k\} \rightarrow^* \boldsymbol{m}$, it is the case that $\boldsymbol{m} \rightarrow^* \{f\colon k\}$. In other words, $\mathcal{W}$ is k-sound if, from k tokens in the initial place, no matter what is fired, it is always possible to end up with k tokens in the final place (and no token elsewhere).

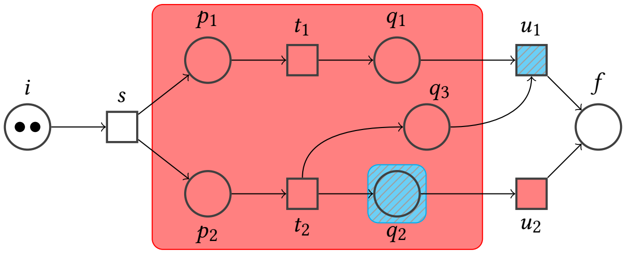


Figure 2: Example of a reset workflow net. Reset arcs are depicted implicitly by colored patterns, rather than explicitly by dotted directed edges. In words, transition u_1 resets place q_2, and transition u_2 resets all places from $\{p_1, p_2, q_1, q_2, q_3\}$. Formally, $R = \{(q_2, u_1)\} \cup \{(r, u_2) : r \in \{p_1, p_2, q_1, q_2, q_3\}\}$. The two filled circles within place i represent two tokens.

We say that a reset workflow net $\mathcal{W}$ is *generalised sound* if it is k-sound for all $k \in \mathbb{N}_{>0}$.

A marking $\boldsymbol{m}$ is a *witness of k-unsoundness* if $\{i\colon k\} \rightarrow^* \boldsymbol{m}$ and $\boldsymbol{m} \not\rightarrow^* \{f\colon k\}$. A marking $\boldsymbol{m}$ is a *witness of unsoundness* if it is a witness of k-unsoundness for some $k \in \mathbb{N}_{>0}$.

Example 2.4. Reconsider the reset Petri net of Figure 1. Taking $i := p_1$ and $f := p_4$ does not yield a reset workflow net for the two following reasons:

- we have: ${}^\bullet p_1 = \{t_2\} \neq \emptyset$, and
- there is no path from p_3 to p_4 (along solid edges).

Figure 2 depicts a reset workflow net $\mathcal{W}$. The set of markings reachable from $\{i\colon 1\}$ in $\mathcal{W}$ is depicted in Figure 3. It is readily seen that any reachable marking $\boldsymbol{m}$ can reach $\{f\colon 1\}$. Thus, $\mathcal{W}$ is 1-sound. However, $\mathcal{W}$ is not 2-sound and hence not generalised sound. Indeed, we have

$$\{i\colon 2\} \rightarrow^{ss} \{p_1\colon 2, p_2\colon 2\} \rightarrow^{t_2}$$
$$\{p_1\colon 2, p_2\colon 1, q_2\colon 1, q_3\colon 1\} \rightarrow^{u_2} \{f\colon 1\}.$$

As $\{f\colon 1\}$ cannot reach any other marking, it cannot reach $\{f\colon 2\}$ as required. So, $\{f\colon 1\}$ is a witness of 2-unsoundness, and hence of unsoundness. □

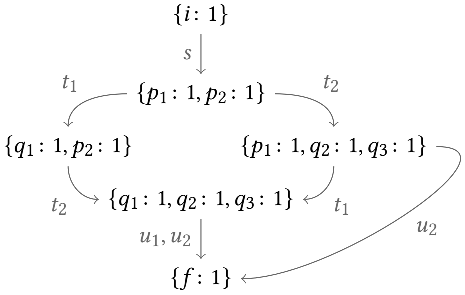

$\{i\colon 1\}$
s
t_1
$\{p_1\colon 1, p_2\colon 1\}$
t_2
$\{q_1\colon 1, p_2\colon 1\}$
$\{p_1\colon 1, q_2\colon 1, q_3\colon 1\}$
t_2
$\{q_1\colon 1, q_2\colon 1, q_3\colon 1\}$
t_1
u_1, u_2
u_2
$\{f\colon 1\}$

Figure 3: The set of markings reachable from $\{i\colon 1\}$ in the reset workflow net of Figure 2. Each edge $\boldsymbol{m} \rightarrow^t \boldsymbol{m'}$ indicates that firing transition t in marking $\boldsymbol{m}$ leads to marking $\boldsymbol{m'}$.

2.5 Subnets

Let $\mathcal{W} = (P, T, F, R, i, f)$ be a reset workflow net, let $Q \subseteq P$ and let $S \subseteq T$. The reset Petri net *obtained from $\mathcal{W}$ by removing places Q and transitions S* is the reset Petri net (P, T, F, R) from which we remove Q, S and any remaining isolated node (i.e. with no incoming and outgoing arc.) More formally, it is the reset Petri net $\mathcal{N} \coloneqq (P', T', F', R')$ where

$$\begin{aligned}
P' &\coloneqq \{p \in P \setminus Q : ({}^\bullet p \cup p^\bullet) \nsubseteq S\}\\
T' &\coloneqq \{t \in T \setminus S : ({}^\bullet t \cup t^\bullet) \nsubseteq Q\},\\
F' &\coloneqq \{(p,t) \in F : p \in P', t \in T'\} \cup \{(t,p) \in F : p \in P', t \in T'\},\\
R' &\coloneqq \{(p,t) \in R : p \in P', t \in T'\}.
\end{aligned}$$

By definition, $\mathcal{N}$ has no isolated node, i.e. with no incoming or outgoing arc. This holds even if we only remove places ($S = \emptyset$) or only remove transitions ($Q = \emptyset$). Indeed, since $\mathcal{W}$ is a reset workflow net, each place $p \in P$ and each transition $t \in T$ has at least one incoming arc or one outgoing arc in $\mathcal{W}$.

3 GENERALISED SOUNDNESS IS UNDECIDABLE

In this section, we prove the following result.

Theorem 3.1. *The generalised soundness problem for reset workflow nets is undecidable.*

We will give a reduction from the reachability problem in Minksy machines. A *(two-counter) Minsky machine* is a finite automaton with two counters that can be incremented, decremented and zero-tested. More formally, it is a pair (Q, Δ) where Q is a finite set of elements called *control states*, and where $\Delta \subseteq Q \times \{\mathtt{x}_i\mathtt{++}, \mathtt{x}_i\mathtt{--}, \mathtt{x}_i = \mathtt{0?} : i \in \{1,2\}\} \times Q$ is a set of elements called *transitions*. A *configuration* of such a machine is a triple $(p, a, b) \in Q \times \mathbb{N} \times \mathbb{N}$ which we will write more concisely as $p(\boldsymbol{v})$ where $\boldsymbol{v} = (a, b)$. For the following definition, let $\neg i$ denote $3 - i$, i.e. the index of the other counter. A transition $(p, \mathtt{oper}, q)$ allows to update a current configuration in control state p into a configuration in control state q with the expected semantic of $\mathtt{oper}$:

$$\begin{aligned}
q(\boldsymbol{v}) &\rightarrow^{(q, \mathtt{x}_i\mathtt{++}, q')} q'(\boldsymbol{v}') && \text{if } \boldsymbol{v}'(i) = \boldsymbol{v}(i) + 1 && \text{and } (*),\\
q(\boldsymbol{v}) &\rightarrow^{(q, \mathtt{x}_i\mathtt{--}, q')} q'(\boldsymbol{v}') && \text{if } \boldsymbol{v}'(i) = \boldsymbol{v}(i) - 1 \geq 0 && \text{and } (*),\\
q(\boldsymbol{v}) &\rightarrow^{(q, \mathtt{x}_i = \mathtt{0?}, q')} q'(\boldsymbol{v}') && \text{if } \boldsymbol{v}'(i) = \boldsymbol{v}(i) = 0 && \text{and } (*),
\end{aligned}$$

where "$(*)$" stands for "$\boldsymbol{v}'(\neg i) = \boldsymbol{v}(\neg i)$".

We write $q(\boldsymbol{v}) \rightarrow q'(\boldsymbol{v}')$ if $q(\boldsymbol{v}) \rightarrow^t q'(\boldsymbol{v}')$ for some $t \in \Delta$. We write $\rightarrow^*$ to denote the reflexive-transitive closure of $\rightarrow$. The *(control-state) reachability problem* asks, given $p, q \in Q$, whether $p(\mathbf{0}) \rightarrow^* q(\mathbf{0})$. It is well known that this problem is undecidable.

Let $q(\boldsymbol{v}) \rightarrow_k q'(\boldsymbol{v}')$ denote the fact that $q(\boldsymbol{v}) \rightarrow q'(\boldsymbol{v}')$ and $0 \leq \boldsymbol{v}(i), \boldsymbol{v}'(i) \leq k$ for both $i \in \{1, 2\}$. Let $\rightarrow_k^*$ denote the reflexive-transitive closure of $\rightarrow_k$. In words, $\rightarrow_k^*$ is the reachability relation where counters remain within $[0..k]$. Clearly, $p(\boldsymbol{u}) \rightarrow^* q(\boldsymbol{v})$ holds iff there exists $k \in \mathbb{N}$ such that $p(\boldsymbol{u}) \rightarrow_k^* q(\boldsymbol{v})$ holds.

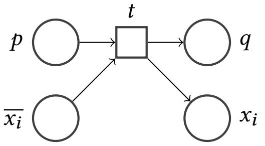
Proposition 3.2. *Given a Minsky machine $\mathcal{M} = (Q, \Delta)$ and control states $q_{src}, q_{tgt} \in Q$, one can construct, in polynomial time, a reset Petri net $\mathcal{N}$ with places $P \coloneqq Q \cup \{x_1, x_2, \overline{x_1}, \overline{x_2}\}$ such that the following holds for every $p, q \in Q$ and $k \in \mathbb{N}$:*

- *$p(\mathbf{0}) \rightarrow_k^* q(\mathbf{0})$ holds in $\mathcal{M}$ iff $\{p\colon 1, \overline{x_1}\colon k, \overline{x_2}\colon k\} \rightarrow^* \{q\colon 1, \overline{x_1}\colon k, \overline{x_2}\colon k\}$ holds in $\mathcal{N}$;*
- *$\{p\colon 1, \overline{x_1}\colon k, \overline{x_2}\colon k\} \rightarrow^* \boldsymbol{m}$ in $\mathcal{N}$ implies $\boldsymbol{m}(Q) = 1$ and $\boldsymbol{m}(x_i) + \boldsymbol{m}(\overline{x_i}) \leq k$ for all $i \in \{1, 2\}$;*
- *Each node of $\mathcal{N}$ is on some path from q_{src} to q_{tgt}.*

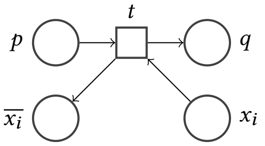
Proof. We use classical notions: budget places and weak simulation of zero-tests. More precisely, each counter x_i of $\mathcal{M}$ is represented by two places in $\mathcal{N}$: x_i and $\overline{x_i}$. Initially, x_i is empty and $\overline{x_i}$ contains k tokens. Whenever x_i is incremented, $\overline{x_i}$ is decremented, and vice versa. This forces x_i to remain within $[0..k]$. Each zero-test of $\mathcal{M}$ is simulated by a reset of x_i. If a reset occurs whenver x_i is empty, then nothing happens. However, if $\mathcal{N}$ "cheats" and resets x_i whenever it is non empty, then $\{x_i, \overline{x_i}\}$ now contains less than k tokens and it will never be possible to increase that number back.

More formally, let us define $\mathcal{N} = (P, T, F, R)$. We set $P \coloneqq Q \cup \{x_1, x_2, \overline{x_1}, \overline{x_2}\}$ and $T \coloneqq \Delta$. For each transition $t = (p, \mathtt{oper}, q) \in \Delta$, we add the following arcs to $\mathcal{N}$.

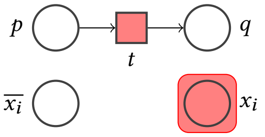
- *Case* $\mathtt{oper} = \mathtt{x}_i\mathtt{++}$. We move the token from p to q, increment x_i and decrement its dual:



- *Case* $\mathtt{oper} = \mathtt{x}_i\mathtt{--}$. We move the token from p to q, decrement x_i and increment its dual:



- *Case* $\mathtt{oper} = \mathtt{x}_i = \mathtt{0?}$. We move the token from p to q, reset x_i and leave $\overline{x_i}$ unchanged:



It is readily seen that starting from marking $\{q_{src}\colon 1, \overline{x_1}\colon k, \overline{x_2}\colon k\}$ there is always exactly one token in Q. Moreover, the two first types of transitions leave the number of tokens in $\{x_i, \overline{x_i}\}$ unchanged, while the third type of transitions may decreases the number of tokens in $\{x_i, \overline{x_i}\}$. So, the second item of the proposition holds.

Let us explain why the first item holds.

$\Rightarrow$) Assume $p(\mathbf{0}) \rightarrow_k^\pi q(\mathbf{0})$ holds in $\mathcal{M}$. We claim that

$$\{p\colon 1, \overline{x_1}\colon k, \overline{x_2}\colon k\} \rightarrow^\pi \{q\colon 1, \overline{x_1}\colon k, \overline{x_2}\colon k\} \text{ holds in } \mathcal{N}.$$

Indeed, (i) resets only occur on empty places, which maintains the invariant that $\{x_i, \overline{x_i}\}$ contains exactly k tokens; (ii) no increment or decrement is ever blocked in $\mathcal{N}$ since we know that counters never exceed k in $\mathcal{M}$.

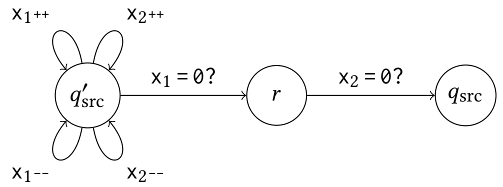
$\Leftarrow$) Assume that

$$\{p\colon 1, \overline{x_1}\colon k, \overline{x_2}\colon k\} \rightarrow^\pi \{q\colon 1, \overline{x_1}\colon k, \overline{x_2}\colon k\} \text{ holds in } \mathcal{N}.$$

As each $\overline{x_i}$ starts and ends with k tokens, this means that each reset that occured in π did not consume any token. Thus, zero-tests were simulated faithfully. Consequently, $p(\mathbf{0}) \to_k^\pi q(\mathbf{0})$ holds in $\mathcal{M}$.

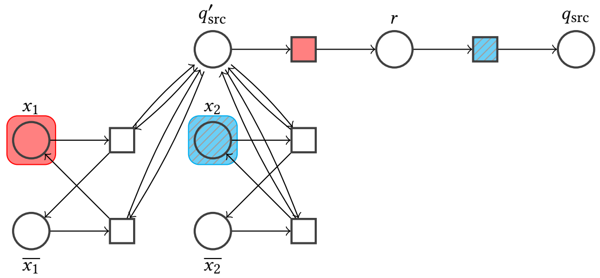
It remains to prove the last item of the proposition, namely that each node of $\mathcal{N}$ is on some path from q_{src} to q_{tgt}. We preprocess $\mathcal{M}$ as follows:

(1) each control state unreachable from q_{src} in the underlying graph is removed from $\mathcal{M}$;
(2) each control state that cannot reach q_{tgt} in the underlying graph is removed from $\mathcal{M}$;
(3) We add two new control states q'_{src} and r to $\mathcal{M}$, and these transitions:



The resulting machine $\mathcal{M}'$ is clearly equivalent, i.e. $q_{\text{src}}(\mathbf{0}) \to_k^* q_{\text{tgt}}(\mathbf{0})$ holds in $\mathcal{M}$ iff $q'_{\text{src}}(\mathbf{0}) \to_k^* q_{\text{tgt}}(\mathbf{0})$ holds in $\mathcal{M}'$.

By the definition of the Petri net $\mathcal{N}'$ obtained from $\mathcal{M}'$, each place of Q is on some path from q'_{src} to q_{tgt} due to (1) and (2), and each place of $\{x_1, x_2, \overline{x_1}, \overline{x_2}\}$ as well since they can all reach q_{src} in this fragment of $\mathcal{N}'$ due to (3):



□

The following proposition establishes Theorem 3.1.

Proposition 3.3. *Given a Minsky machine $\mathcal{M} = (Q, \Delta)$ and control states $q_{src}, q_{tgt} \in Q$, one can construct, in polynomial time, a reset workflow net $\mathcal{W}$ such that $\mathcal{W}$ is generalised sound iff $q_{src}(\mathbf{0}) \not\to^* q_{tgt}(\mathbf{0})$ holds in $\mathcal{M}$.*

Proof. We first sketch the reset workflow net $\mathcal{W}$, describe the construction, and show that it is indeed a reset workflow net.

Let $\mathcal{N}$ be the reset Petri net given by Proposition 3.2 from $\mathcal{M}$. The reset workflow net $\mathcal{W}$ consists of $\mathcal{N}$ together with the extra places $\{i, r, f\}$ and transitions $\{t_1, t_2, t_3\}$. Figure 4 depicts $\mathcal{W}$ where the solid red part corresponds to $\mathcal{N}$. Places i and f are respectively the initial and final places of $\mathcal{W}$. When either of t_1, t_2 or t_3 is fired, all places from the corresponding colored area is reset. By Proposition 3.2, each node of $\mathcal{W}$ is on some path from i to f.

Let us now show that there exists $k \in \mathbb{N}$ such that $q_{\text{src}}(\mathbf{0}) \to_k^* q_{\text{tgt}}(\mathbf{0})$ in $\mathcal{M}$ iff $\mathcal{W}$ is not generalised sound.

$\Rightarrow$) Let $k \in \mathbb{N}$ be such that $q_{\text{src}}(\mathbf{0}) \to_k^* q_{\text{tgt}}(\mathbf{0})$ in $\mathcal{M}$. By Proposition 3.2, we have

$$\{q_{\text{src}}\colon 1, \overline{x_1}\colon k, \overline{x_2}\colon k\} \to^\pi \{q_{\text{tgt}}\colon 1, \overline{x_1}\colon k, \overline{x_2}\colon k\} \text{ in } \mathcal{N}.$$

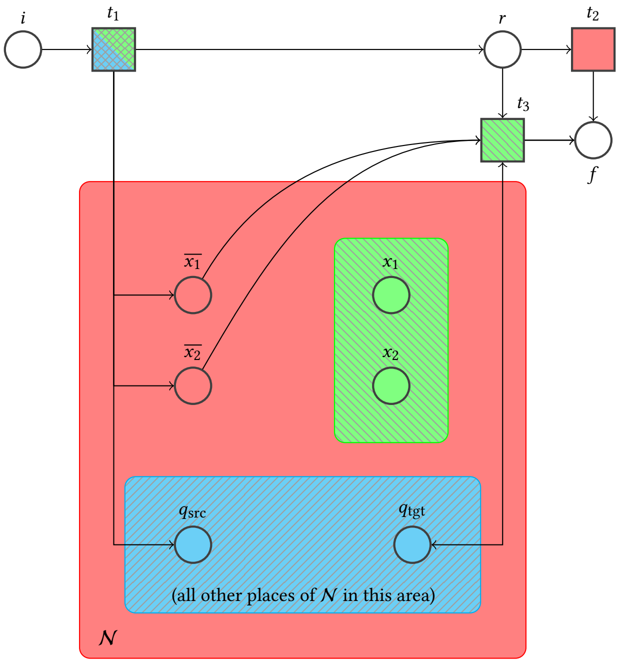


Figure 4: The reset workflow net $\mathcal{W}$. The bidirectional arc between q_{tgt} and t_3 represents two arcs: one in each direction. Each transition t_i resets the places within the part of the corresponding pattern and color. More precisely, t_1 resets all places of $\mathcal{N}$ except for $\{\overline{x_1}, \overline{x_2}\}$; t_2 resets all places of $\mathcal{N}$; and t_3 resets $\{x_1, x_2\}$.

Thus, the following holds in $\mathcal{W}$:

$$\begin{aligned}\{i\colon k\} &\to^{t_1^k} \{q_{\text{src}}\colon 1, \overline{x_1}\colon k, \overline{x_2}\colon k, r\colon k\}\\ &\to^{\pi} \{q_{\text{tgt}}\colon 1, \overline{x_1}\colon k, \overline{x_2}\colon k, r\colon k\}\\ &\to^{t_3^k} \{q_{\text{tgt}}\colon 1, f\colon k\}.\end{aligned}$$

The latter marking witnesses k-unsoundness of $\mathcal{W}$. Indeed, by Proposition 3.2, the subset Q of places of $\mathcal{N}$, that corresponds to the control states of $\mathcal{M}$, cannot be emptied.

$\Leftarrow$) Let $\{i\colon k\} \to^\pi \boldsymbol{m}$ witness k-unsoundness of $\mathcal{W}$, where $k \in \mathbb{N}_{>0}$. First, we show that $\boldsymbol{m}(f) = k$. Note that in any marking $\boldsymbol{m}'$ reachable from $\{i\colon k\}$, we have $\boldsymbol{m}'(i) + \boldsymbol{m}'(r) + \boldsymbol{m}'(f) = k$. Thus, $\boldsymbol{m}(f) \leq k$. For the sake of contradiction, suppose that $\boldsymbol{m}(f) < k$. Let $k_i \coloneqq \boldsymbol{m}(i)$ and $k_r \coloneqq \boldsymbol{m}(r)$. By the previous equality, we have $k_i + k_r > 0$. The following holds in $\mathcal{W}$:

$$\boldsymbol{m} \to^{t_1^{k_i}\, t_2^{k_i+k_r}} \{f\colon k\}.$$

Indeed, as $k_i + k_r > 0$, transition t_2 is fired at least once and hence $\mathcal{N}$ is emptied (the solid red area of Figure 4). This contradicts the assumption that $\boldsymbol{m}$ witnesses k-unsoundness.

We have established that $\boldsymbol{m}(f) = k$, which implies that $\boldsymbol{m}(i) = 0$ and $\boldsymbol{m}(r) = 0$. Without loss of generality we can assume that k is minimal i.e. $\mathcal{W}$ is ℓ-sound for every $\ell < k$.

Claim 1. *Transition t_2 does not appear in π.*

For the sake of contradiction, suppose the claim does not hold. We split run π into $\pi = \pi_1\pi_2$ where π_2 is a maximal suffix that does not contain transition t_2. We have $\{i\colon k\} \xrightarrow{\pi_1} \boldsymbol{m}_1 \xrightarrow{\pi_2} \boldsymbol{m}$ for some marking $\boldsymbol{m}_1$.

Let us compare the number of occurrences of t_1 and t_3 in π_2. As t_2 is the last transition of π_1, we have $\boldsymbol{m}_1(p) = 0$ for every place p of $\mathcal{N}$. By Proposition 3.2, the number of tokens in $\{x_i, \overline{x_i}\}$ cannot increase by firing transitions of $\mathcal{N}$. Thus, we must have $|\pi_2|_{t_1} \geq |\pi_2|_{t_3}$. Since $\boldsymbol{m}(r) = 0$, we must have $|\pi_2|_{t_1} \leq |\pi_2|_{t_3}$. Consequently, $|\pi_2|_{t_1} = |\pi_2|_{t_3}$. Moreover, $|\pi_2|_{t_1} > 0$ as otherwise $\boldsymbol{m} = \{f\colon k\}$, which is obviously not a witness of k-unsoundness.

From this and $\boldsymbol{m}(r) = 0$, we conclude that $\boldsymbol{m}_1(r) = 0$. This means that all places, except possibly i and f, are empty in $\boldsymbol{m}_1$. Since $\boldsymbol{m}(f) = k$ and $|\pi_2|_{t_1} = |\pi_2|_{t_3} > 0$, the marking $\boldsymbol{m}_1$ is of the form $\boldsymbol{m}_1 = \{i\colon k - \ell, f\colon \ell\}$ where $0 < \ell < k$.

As no transition consumes from f, we have $\{i\colon k - \ell\} \xrightarrow{\pi_2} \boldsymbol{m}'$ where $\boldsymbol{m}' \coloneqq \boldsymbol{m} - \{f\colon \ell\}$. Since $\boldsymbol{m}'(f) = \boldsymbol{m}(f) - \ell = k - \ell$, we conclude that $\mathcal{W}$ is not $(k - \ell)$-sound as $\boldsymbol{m}'$ is also a witness of unsoundness. This contradicts the minimality of k. So, Claim 1 holds as desired.

Claim 2. *In run π,*

(1) *Every occurrence of t_1 does not remove any token from places $\{x_1, x_2\}$;*
(2) *Every occurrence of t_3 consumes (exactly) two tokens from places $\{x_1, x_2, \overline{x_1}, \overline{x_2}\}$.*

From Claim 1, we have $|\pi|_{t_1} = |\pi|_{t_3}$. Moreover, each occurrence of transition t_3 removes at least two tokens from $\{x_1, x_2, \overline{x_1}, \overline{x_2}\}$, and each occurrence of transition t_1 adds at most two tokens to $\{x_1, x_2, \overline{x_1}, \overline{x_2}\}$. Hence, Claim 2 holds as desired.

Now, let us split π into $\pi = \pi_3\pi_4\pi_5$ where $\pi_3\pi_4$ is the longest prefix of π without t_3, and π_4 is a maximal suffix of $\pi_3\pi_4$ that does not contain transition t_1. Let $\boldsymbol{m}_3$ and $\boldsymbol{m}_4$ be the markings such that $\{i\colon k\} \xrightarrow{\pi_3} \boldsymbol{m}_3 \xrightarrow{\pi_4} \boldsymbol{m}_4$. We have $\boldsymbol{m}_4(x_1) = \boldsymbol{m}_4(x_2) = 0$, as otherwise the first occurrence of t_3 removes at least three tokens from places $\{x_1, x_2, \overline{x_1}, \overline{x_2}\}$, which contradicts Claim 2. A similar argument shows that $\boldsymbol{m}_3(x_1) = \boldsymbol{m}_3(x_2) = 0$.

Since the last transition of π_3 is t_1, since π_3 contains no occurrence of $\{t_2, t_3\}$, since π_4 contains no occurrence of $\{t_1, t_2, t_3\}$, and since the first transition of π_5 is t_3, there exists $a > 0$ such that

- $\boldsymbol{m}_3 = \{i\colon k - a, r\colon a, q_{\mathrm{src}}\colon 1, \overline{x_1}\colon a, \overline{x_2}\colon a\}$, and
- $\boldsymbol{m}_4 = \{i\colon k - a, r\colon a, q_{\mathrm{tgt}}\colon 1, \overline{x_1}\colon a, \overline{x_2}\colon a\}$.

This implies that $\boldsymbol{m}_3 \xrightarrow{\pi_4} \boldsymbol{m}_4$ induces a run $q_{\mathrm{src}}(\mathbf{0}) \rightarrow_a^* q_{\mathrm{tgt}}(\mathbf{0})$ of Minsky machine $\mathcal{M}$. □

4 IN BETWEEN SOUNDNESS

We start this section by discussing properties that reset workflow nets can satisfy. So far, we mostly discussed generalised soundness and 1-soundness, or more generally, k-soundness for any $k \in \mathbb{N}_{>0}$. We say that a reset workflow net is *up-to-k-sound* [34, Definition 24] if it is j-sound for all $j \in [1..k]$. Observe that up-to-1-soundness is equivalent to 1-soundness. By definition, a reset workflow net which is generalised soundness is also up-to-k sound (for any $k \in \mathbb{N}_{>0}$).

We say that a property $\mathcal{P}$, of reset workflow nets, is *k-in-between sound* if: all generalised sound workflow nets satisfy $\mathcal{P}$; and every workflow net that satisfies $\mathcal{P}$ is up-to-k-sound. Figure 5 depicts all of the aforementioned classes.

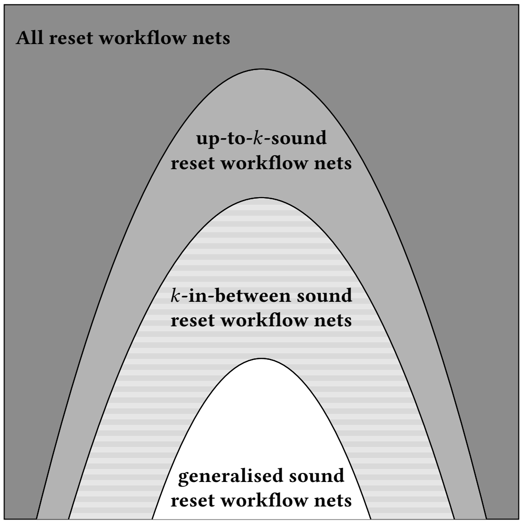

All reset workflow nets
up-to-k-sound reset workflow nets
k-in-between sound reset workflow nets
generalised sound reset workflow nets

Figure 5: Classes of reset workflow nets: generalised sound, up-to-k-sound, k-in-between sound and all reset workflow nets. Properties with lighter colors also satisfy darker colored properties. For example, the class of generalised sound reset workflow nets is the most restrictive and it is contained in all other classes. Note that "k-in-between sound" is a class of properties, not a single one. So, in the figure, one should think of each horizontal line as one of these properties.

Our first main result is as follows.

Theorem 4.1. *For every $k \in \mathbb{N}_{>0}$, there exists a decidable k-in-between sound property $\mathcal{P}_k$ of reset workflow nets. More precisely: given as input $k \in \mathbb{N}_{>0}$ and a reset workflow net, there is an algorithm deciding $\mathcal{P}_k$ running in Ackermannian time.*

Remark 1. *In particular, given a reset workflow net $\mathcal{W}$ and $k \in \mathbb{N}_{>0}$, there is an algorithm that correctly outputs: either that $\mathcal{W}$ is up-to-k-sound; or that $\mathcal{W}$ is not generalised sound. More precisely, if $\mathcal{W}$ is not up-to-k-sound, then it outputs that $\mathcal{W}$ is not generalised sound; if $\mathcal{W}$ is generalised sound, then it outputs that it is up-to-k-sound; otherwise, it can output either of the two properties (both hold).*

We deliberately postpone the formal definition of property $\mathcal{P}_k$ as it is technical. Instead, we give intuition on where it comes from. First observe that soundness is a conjunction of three simpler informal properties: *(i)* It is impossible to strictly cover the final marking; *(ii)* It is impossible to reach markings with tokens only in the final place, but with insufficiently many of them; and *(iii)* It is impossible to reach a marking that has tokens in places other than f and from which it is impossible to produce more tokens in f.

Now, on the one hand, to decide 1-soundness for *(i)* we can check if it is impossible to strictly cover $\{f\colon 1\}$ as coverability is decidable. Property *(ii)* holds as the workflow net cannot be emptied, indeed, firing any transition always produces tokens. So, the reason for undecidability of 1-soundness is the hardness of property *(iii)*. On the other hand, for generalised soundness, it can be proven that property *(ii)* is decidable, and, assuming that *(i)* holds *(iii)* is decidable, so the essential reason of the undecidability of the generalised soundness is *(i)*. The idea behind $\mathcal{P}_k$ is to exploit the decidable parts, which intuitively cover all three parts of soundness. A bit more precisely, we combine the test for *(ii)* for generalised soundness, the test for *(i)* for k-soundness, and *(iii)* for generalised soundness assuming that *(i)* holds. This mixture of decidable properties gives rise to a relation that is k-in-between.

Surprisingly, the property $\mathcal{P}_k$ defined in this way for sufficiently large k coincides with to up-to-k soundness. More formally:

Theorem 4.2. *Given a nonredundant reset workflow net $\mathcal{W}$, there is a computable number $k' \in \mathbb{N}_{>0}$ (Ackermannian in the size of $\mathcal{W}$) such that, for all $k \geq k'$, $\mathcal{W}$ satisfies $\mathcal{P}_k$ iff $\mathcal{W}$ is up-to-k sound.*

The rest of the section is organized as follows. First, in Section 4.1, we introduce the notion of "nonredundancy", which allows us to identify unimportant places and transitions. Next, in Section 4.2, we define the "skeleton" of a reset workflow net, which is a crucial object in deciding property (ii) for generalised soundness. In Section 4.3, we show that the skeleton of a nonredundant reset workflow net is a workflow net (without resets). In Section 4.4, we show that if a nonredundant reset workflow net is generalised sound, then its skeleton is also generalised sound. This last property implies property (ii) for generalised soundness. Finally, Section 4.5 makes use of these results to prove Theorems 4.1 and 4.2; Here, in particular, we show how property (iii) can be tested assuming that (i) holds, although it is not given explicitly to keep the argument shorter.

4.1 Nonredundancy

We provide a technical definition of redundancy, following similar definitions for workflow nets (without resets) [36]. Intuitively, it allows to ignore useless places and transitions from the net without changing its set of reachable markings.

Formally, given a reset workflow net $\mathcal{W}$, we say that

(1) a place p is *nonredundant* if there exist $k \in \mathbb{N}$ and a marking $\boldsymbol{m}$ such that $\{i\colon k\} \rightarrow^* \boldsymbol{m}$ and $\boldsymbol{m}(p) \geq 1$;
(2) a transition t is *nonredundant* if there exist $k \in \mathbb{N}$ and a marking $\boldsymbol{m}$ such that $\{i\colon k\} \rightarrow^* \boldsymbol{m}$ and t is enabled in $\boldsymbol{m}$.

If Item 1 does not hold for a place p, then we say that p is *redundant*, and likewise for a transition t.

Proposition 4.3. *Given a reset workflow net $\mathcal{W}$, one can compute its set of redundant places and transitions. The procedure works in Ackermannian time. Further, for every nonredundant transition t and place p, there are numbers $k_t, k_p \in \mathbb{N}$ bounded Ackermannianly and runs of length at most Ackermannian such that $\{i\colon k_t\} \rightarrow^* \uparrow({}^\bullet t)$ and $\{i\colon k_p\} \rightarrow^* \uparrow\{p\colon 1\}$.*

Proof. Let us show how to compute the set of nonredundant places and transitions. Recall that $C_{\boldsymbol{m}'}$ is the upward closed set of all markings $\boldsymbol{m}$ from which it is possible to cover the marking $\boldsymbol{m}'$; one may compute the set of minimal elements of $C_{\boldsymbol{m}'}$ with the backward coverability algorithm. As mentioned in the preliminaries, every upward closed subset of $\mathbb{N}^P$ is equal to a finite union of elements $\uparrow\boldsymbol{x}_i$ with $\boldsymbol{x}_i \in \mathbb{N}^P$.

From the definition of nonredundancy, a place p is nonredundant iff there exists some k_p such that $\{i\colon k_p\}$ is in $C_{\{p\colon 1\}}$; this is decidable, using the backwards coverability algorithm [19], and one may compute such a k_p. Similarly, a transition t is nonredundant iff there exists k_t such that $\{i\colon k_t\}$ is in $C_{{}^\bullet t}$, and this is also decidable and k_t is still computable. Now, let K be the sum of all k_p and k_t for $p \in P$ and $t \in T$. Observe that, for all markings $\boldsymbol{m} \in \bigcup_{p \in P, t \in T}\{\{p\colon 1\} + {}^\bullet t\}$, there is a run, from the initial marking $\{i\colon K\}$, that covers $\boldsymbol{m}$. The Ackermannian bounds follow from the bounds on the coverability problem [9]. □

We make the following observation on nonredundant transitions.

Claim 3. *Let $\mathcal{W}$ be a reset workflow net which is generalised sound. Nonredundant transitions cannot reset either i or f. Moreover, this claim still holds if we relax the requirement of $\mathcal{W}$ being generalised sound to a weaker one, namely to "$\mathcal{W}$ is up-to-K sound", where number K is at most Ackermannian.*

Proof: For the final place f, the claim follows by definition. Let us consider the case of the initial place i. Towards a contradiction, suppose that $\mathcal{W}$ has a nonredundant transition t such that t resets place i. By Proposition 4.3, there exists a number $k \in \mathbb{N}_{>0}$ which is at most Ackermannian such that

$$\{i\colon k\} \rightarrow^{\pi} \boldsymbol{m}' \rightarrow^{t} \boldsymbol{m},$$

for some run π that does not use transition t, and some markings $\boldsymbol{m}'$ and $\boldsymbol{m}$. As k is Ackermannian, we can safely assume that $k < K$. Thus, the following holds for $K > k$:

$$\{i\colon K\} \rightarrow^{\pi} \boldsymbol{m}' + \{i\colon x\} \rightarrow^{t} \boldsymbol{m},$$

for $x \geq K - k > 0$. Note that t resets place i, and that k is defined in such a way that at least one token is lost. By generalised soundness, or up-to-K soundness of $\mathcal{W}$, we have $\boldsymbol{m} \rightarrow^* \{f\colon k\}$ and $\boldsymbol{m} \rightarrow^* \{f\colon K\}$. This contradicts generalised soundness of $\mathcal{W}$ and up-to-K soundness. ■

4.2 Skeletons of reset workflow nets

In this subsection, we consider a relaxation of generalised soundness. Given a reset workflow net $\mathcal{W}$, we will define a workflow net $\mathcal{W}^s$ obtained by removing redundancy and resetable places. Intuitively, generalised soundness of $\mathcal{W}$ should imply generalised soundness of $\mathcal{W}^s$. As we shall see, this requires some work. We start by introducing the notation.

Let $\mathcal{W} = (P, T, F, R, i, f)$ be a reset workflow net. We say that place $p \in P$ is *resetable* if there exists a nonredundant transition $t \in T$ such that $(p, t) \in R$. The *skeleton* of a reset workflow net $\mathcal{W}$ is the Petri net obtained by removing redundant places, redundant transitions and resetable places, and next removing isolated transitions (as defined in Section 2.5 of the preliminaries).

We denote this Petri net by $\mathcal{W}^s = (P^s, T^s, F^s)$. If there are well-defined initial and final places i^s, f^s such that $(P^s, T^s, F^s, i^s, f^s)$ is a workflow net, then, slightly abusing the notation, we write that $\mathcal{W}^s = (P^s, T^s, F^s, i^s, f^s)$ is a workflow net. We will devote the forthcoming Sections 4.3 and 4.4 to proving the following.

Proposition 4.4. *Let $\mathcal{W}$ be a reset workflow net which is generalised sound. It is the case that*

(1) *the skeleton $\mathcal{W}^s$ is a workflow net;*

(2) *$\mathcal{W}^s$ is also generalised sound.*

Moreover, this claim still holds if we relax the requirement of $\mathcal{W}$ being generalised sound to a weaker one, namely to "$\mathcal{W}$ is up-to-K sound", where number K is at most Ackermannian.

Note that generalised soundness for workflow nets (without resets) is decidable [36], and belongs to PSPACE [3]. So, Proposition 4.4 implies that generalised soundness for the skeleton workflow net is a decidable relaxation of generalised soundness for $\mathcal{W}$.

Before we prove Proposition 4.4, we need a definition that allows us to associate markings in $\mathcal{W}$ with markings in $\mathcal{W}^s$.

The function $\mathit{Reset}\colon \mathbb{N}^P \to \mathbb{N}^P$ is defined by

$$\mathit{Reset}(\boldsymbol{x})(p) = \begin{cases} \boldsymbol{x}(p) & \text{if } p \in P^s, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\mathit{Reset}(\boldsymbol{x})$ is equal to applying $\mathit{Reset}_t(\cdot)$ to $\boldsymbol{x}$ for each nonredundant transition t (in any order). Lemma 4.6 will provide some intuition on why $\mathit{Reset}(\boldsymbol{x})$ corresponds to a marking in $\mathcal{W}^s$.

As we often consider runs in the reset workflow $\mathcal{W}$, as well as runs in its skeleton $\mathcal{W}^s$, we introduce the notation $\rightarrow_s$ to denote runs specifically in $\mathcal{W}^s$.

Lemma 4.5. *Let $\mathcal{W}$ be a reset workflow net which is generalised sound. There exist $z \in \mathbb{N}$ and a run $\{i\colon z\} \rightarrow^{\zeta} \{f\colon z\}$, where ζ contains each nonredundant transition of $\mathcal{W}$. Moreover, z and $|\zeta|$ are at most Ackermannian.*

Further, the claim still holds if we relax the assumption of $\mathcal{W}$ being generalised sound to a weaker one, namely to "$\mathcal{W}$ is up-to-K sound", where K is at most Ackermannian (the same as the bound on z).

Proof. For every nonredundant transition t of $\mathcal{W}$, there is a run firing t, i.e. $\{i\colon z_t\} \rightarrow^{\rho_t t} \boldsymbol{m}_t$ for some $z_t \in \mathbb{N}$ and some marking $\boldsymbol{m}_t$. We define $z := \sum_{t\in T} z_t$. Let ρ be the concatenation (in any order) of runs $\rho_t t$. Since $\mathcal{W}$ is up-to-K sound, by Claim 3, the initial place i cannot be reset by any nonredundant transition of $\mathcal{W}$. Thus, $\{i\colon z\} \rightarrow^{\rho} \boldsymbol{m}'$ for some marking $\boldsymbol{m}'$. Since $\mathcal{W}$ is generalised sound (or up-to-z-sound, as $z \leq K$), there is a run δ such that $\boldsymbol{m}' \rightarrow^{\delta} \{f\colon z\}$. We conclude by taking $\zeta := \rho\delta$. The Ackermannian bounds on ρ follow from Proposition 4.3. The Ackermannian bounds on δ follow from a bound on the maximal length of the coverability run in the reset Petri net (Lemma 2.3 with target marking $\{f\colon z\}$). Thus, $|\zeta| = |\rho\delta|$ is at most Ackermannian. □

Remark 2. *From the proof of Lemma 4.5, we can see that if the initial place of $\mathcal{W}$ cannot be reset, then there exist $z \in \mathbb{N}$ and $\{i\colon z\} \rightarrow^{\zeta} \boldsymbol{m}$, for some marking $\boldsymbol{m}$, that contains each nonredundant transition.*

We say that a marking $\boldsymbol{m} \in \mathbb{N}^P$ of $\mathcal{W}$ is *reachable* if there exists $k \in \mathbb{N}_{>0}$ such that $\{i\colon k\} \rightarrow^{*} \boldsymbol{m}$.

Lemma 4.6. *Let ζ be as in Lemma 4.5, and let $\boldsymbol{m}$ be a reachable marking of $\mathcal{W}$. It is the case that $\{i\colon z\} + \boldsymbol{m} \rightarrow^{\zeta} \{f\colon z\} + \mathit{Reset}(\boldsymbol{m})$.*

Proof. We have $\{i\colon z\} + \boldsymbol{m} \rightarrow^{\zeta} \boldsymbol{n}$ for some marking $\boldsymbol{n}$. We need to prove that $\boldsymbol{n} = \{f\colon z\} + \mathit{Reset}(\boldsymbol{m})$. It is easy to see that $\boldsymbol{n}(p) = (\{f\colon z\} + \mathit{Reset}(\boldsymbol{m}))(p)$ for all $p \in P^s$ as these places are not resetable, and the effect on them is the sum of effects on all transitions in ζ. Thus, we need to prove that $\boldsymbol{n}(p) = 0$ for all $p \notin P^s$.

Let $p \notin P^s$. If p is redundant, then $\boldsymbol{n}(p) = \boldsymbol{m}(p) = 0$ since p is not marked in any reachable marking. Further, f cannot be reset by Claim 3, and $p \neq f$ as f is nonredundant by generalised soundness (or up-to-K soundness) of $\mathcal{W}$.

Thus, we may assume that p is a nonredundant place, which is reset by a nonredundant transition t. By definition, we can decompose ζ as $\zeta = \rho_1 t \rho_2$. Let $\{i\colon z\} \rightarrow^{\rho_1 t} \boldsymbol{n}_1$ and $\{i\colon z\} + \boldsymbol{m} \rightarrow^{\rho_1 t} \boldsymbol{n}_2$. We know that $\boldsymbol{n}_1(p) = \boldsymbol{n}_2(p) = 0$. Consider two runs, with the same sequence of transitions: ρ_2, but different starting points: $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$. By induction, one can prove that the markings in both runs will always have the same value in p. Indeed, they are the same initially, and the same sequence of transitions are applied afterwards. Thus, in the end $\boldsymbol{n}(p) = (\{f\colon z\} + \mathit{Reset}(\boldsymbol{m}))(p) = 0$ as required. □

The next two subsections are devoted to proving the two items of Proposition 4.4.

4.3 Skeletons are workflow nets

In this subsection, we prove the first item of Proposition 4.4. We fix a reset workflow net $\mathcal{W} = (P, T, F, R, i, f)$ as in the statement of Proposition 4.4 and its skeleton (P^s, T^s, F^s). Overall, our aim is to prove that $\mathcal{W}^s = (P^s, T^s, F^s, i, f)$ is a well-defined workflow net.

It follows directly from Claim 3 that no nonredundant transition of $\mathcal{W}$ resets i or f. Thus, the following holds.

Lemma 4.7. *It is the case that $i, f \in P^s$.*

The following claim is useful and trivial by definition.

Claim 4. *Consider a run $\boldsymbol{m} \rightarrow^{\rho} \boldsymbol{m}'$ of $\mathcal{W}$. Let $\boldsymbol{m}_s, \boldsymbol{m}'_s \in \mathbb{N}^{P^s}$ be the markings obtained by projecting $\boldsymbol{m}$ and $\boldsymbol{m}'$ onto P^s; and let ρ_s be the run in $\mathcal{W}^s$ obtained from ρ (i.e. transitions are restricted to P^s, and possibly isolated transitions are removed). It is the case that $\boldsymbol{m}_s \rightarrow_s^{\rho_s} \boldsymbol{m}'_s$.*

We do not know yet whether $\mathcal{W}^s$ is a workflow net, but the definition of nonredundancy makes sense for $\mathcal{W}^s$ even if it is not a workflow net. Below, we note that nonredundancy of places in $\mathcal{W}$ easily implies nonredundancy of places in $\mathcal{W}^s$.

Claim 5. *For every place $p \in P^s$, there exist $k \in \mathbb{N}$, with $k \leq K$, and a run $\{i\colon k\} \rightarrow_s^{*} \boldsymbol{m}_s$ such that $\boldsymbol{m}_s(p) > 0$ and $\boldsymbol{m}_s \rightarrow_s^{*} \{f\colon k\}$.*

Proof: By definition of $\mathcal{W}^s$, place p is nonredundant in $\mathcal{W}$, and hence we have $\{i\colon k\} \rightarrow^{*} \boldsymbol{m}$ where $\boldsymbol{m}(p) > 0$. Since $\mathcal{W}$ is generalised sound or up-to-k sound for $k \leq K$, there is a run $\boldsymbol{m} \rightarrow^{*} \{f\colon k\}$. These two runs and Claim 4 conclude the proof. ■

To show Proposition 4.4 (1), it remains to prove that all places from P^s and all transitions from T^s are on a path from i to f in $\mathcal{W}^s$. The following lemma reduces the problem to checking this property for places only.

Lemma 4.8. *Let $t \in T^s$. It is the case that ${}^\bullet t \neq \emptyset$ and $t^\bullet \neq \emptyset$.*

We will need the following technical claim. Note that although its statement deals with places $P^s \subseteq P$ of the skeleton, the claim deals with runs from $\mathcal{W}$.

Claim 6. *Let $p \in P^s$. There is no $k \in \mathbb{N}$, $k \leq K$ such that $\mathcal{W}$ has two runs $\{i\colon k\} \rightarrow^{\pi} \boldsymbol{m}$ and $\{i\colon k\} \rightarrow^{\pi'} \boldsymbol{m}'$ with $\boldsymbol{m}' \geq \boldsymbol{m} + \{p\colon 1\}$.*

Proof: For the sake of contradiction, suppose two such runs exist. By generalised soundness or up-to-K soundness of $\mathcal{W}$, there is a run $\boldsymbol{m} \to^{\gamma} \{f\colon k\}$. Since $\boldsymbol{m'} \geq \boldsymbol{m} + \{p\colon 1\}$ and p is not resetable, we have $\{i\colon k\} \to^{\pi'\gamma} \boldsymbol{m''}$ for some $\boldsymbol{m''} \geq \{f\colon k, p\colon 1\}$. This contradicts general soundness or up-to-K soundness as $\boldsymbol{m''} \not\to^{*} \{f\colon k\}$. ■

Proof of Lemma 4.8. Towards a contradiction, suppose there exists $t \in T^s$ such that ${}^\bullet t = \emptyset$. By definition, there exists $p \in t^\bullet$ for some $p \in P^s$ (otherwise t would be an isolated transition in $\mathcal{W}^s$). Let $u \in T$ be the original nonredundant transition from which t is obtained. Let z, z_u, ζ, and ρ_u be as in the proof of Lemma 4.5. We consider these runs of $\mathcal{W}$:

(1) $\{i\colon z_u\} \to^{\rho_u} \boldsymbol{m}_u \to^{u} \boldsymbol{m}'_u$ (it exists by nonredundancy);
(2) $\{i\colon z_u + z\} \to^{\rho_u} \boldsymbol{m}_u + \{i\colon z\} \to^{\zeta} \{f\colon z\} + Reset(\boldsymbol{m}_u)$;
(3) $\{i\colon z_u + z\} \to^{\rho_u u} \boldsymbol{m}'_u + \{i\colon z\} \to^{\zeta} Reset(\boldsymbol{m}'_u) + \{f\colon z\}$.

The last two runs are obtained by Lemma 4.6. Observe that

$$Reset(\boldsymbol{m}'_u) \geq Reset(\boldsymbol{m}_u) + \{p\colon 1\}.$$

Indeed: since ${}^\bullet t = \emptyset$, we have $\boldsymbol{m}'_u(r) \geq \boldsymbol{m}_u(r)$ for all $r \in P^s$; and, since $p \in t^\bullet$, we also have $\boldsymbol{m}'_u(p) \geq \boldsymbol{m}_u(p) + 1$. By Claim 6, the runs (2) and (3) contradict the generalised soundness of $\mathcal{W}$ and up-to-K soundness assuming that $K > z_u + z$.

The proof of the case where there is a transition $t \in T^s$ such that $t^\bullet = \emptyset$ is essentially the same. □

To prove Proposition 4.4 (1), it remains to prove that all places in P^s are on a path from i to f. We need to recall some standard definitions from Petri net theory (e.g. see [26]).

A *siphon* is a subset of places $S \subseteq P$ such that for every transition t: if $t^\bullet \cap S \neq \emptyset$ then ${}^\bullet t \cap S \neq \emptyset$. Similarly, a *trap* is a subset of places S' such that for every transition t: if ${}^\bullet t \cap S' \neq \emptyset$ then $t^\bullet \cap S' \neq \emptyset$. By definition, it is readily seen that every unmarked siphon remains unmarked and every marked trap remains marked. More formally:

Lemma 4.9. *Let S be a siphon and let S' be a trap of a Petri net (without resets). For every $\boldsymbol{m} \to^{*} \boldsymbol{m'}$, the following holds:*

(1) *if $\boldsymbol{m}(S) = 0$, then $\boldsymbol{m'}(S) = 0$;*
(2) *if $\boldsymbol{m}(S') > 0$, then $\boldsymbol{m'}(S') > 0$.*

The next two lemmas conclude the proof of Proposition 4.4 (1).

Lemma 4.10. *For every $p \in P^s$, there is a path from i to p in $\mathcal{W}^s$.*

Proof. Let $p \in P^s$. Let $X \subseteq P^s$ be the set of places from which there is a path to p in $\mathcal{W}^s$. Observe that X is a siphon in $\mathcal{W}^s$. Indeed, if $t^\bullet \cap X \neq \emptyset$ then ${}^\bullet t \subseteq X$ and ${}^\bullet t \neq \emptyset$ because of Lemma 4.8.

Note that $p \in X$ and, by Claim 5, the place p can be marked in $\mathcal{W}^s$. Initially, only place i is marked. Thus, by Lemma 4.9, it must be the case that $i \in X$ as otherwise p could not be marked. □

Lemma 4.11. *For every $p \in P^s$, there is a path from p to f in $\mathcal{W}^s$.*

Proof. Let $p \in P^s$. Let $X \subseteq P^s$ be the set of places to which there is a path from p in $\mathcal{W}^s$. Observe that X is a trap in $\mathcal{W}^s$. Indeed, if ${}^\bullet t \cap X \neq \emptyset$, then $t^\bullet \subseteq X$, and $t^\bullet \neq \emptyset$ because of Lemma 4.8.

By Claim 5, there exist $k \in \mathbb{N}$ and a marking $\boldsymbol{m}_s \in \mathbb{N}^{P^s}$ such that $\{i\colon k\} \to_s^{*} \boldsymbol{m}_s$, where $\boldsymbol{m}_s(p) > 0$ and $\boldsymbol{m}_s \to_s^{*} \{f\colon k\}$. Note that $p \in X$. Since $\boldsymbol{m}_s \to_s^{*} \{f\colon k\}$ by Lemma 4.9, we get $f \in X$. □

4.4 Skeletons preserve generalised soundness

In this subsection, we will prove Proposition 4.4 (2), i.e. we show that if $\mathcal{W}$ is generalised sound (or up-to-K sound for some sufficiently large k), then its skeleton $\mathcal{W}^s$ is also generalised sound. To do so, we will prove a general lemma, which will also be useful in the next section. The lemma will not need the assumption of generalised soundness, but only a weaker property. We start by defining two properties for reset workflow nets.

We say that a reset workflow net $\mathcal{W}$ has a *full reset run* if there exist $z \in \mathbb{N}$ and a run ζ such that these three conditions all hold:

- $\{i\colon z\} \to^{\zeta} \{f\colon z\}$ and ζ contains each nonredundant transition of $\mathcal{W}$;
- if $\boldsymbol{m}$ is a reachable marking of $\mathcal{W}$, then $\{i\colon z\} + \boldsymbol{m} \to^{\zeta} \{f\colon z\} + Reset(\boldsymbol{m})$;
- for every decomposition $\zeta = \rho t \rho'$, where $\{i\colon z\} \to^{\rho} \boldsymbol{m}$, we have $Reset(\boldsymbol{m}) \to^{*} \{f\colon z\}$.

We call ζ a *full reset run.* Below, we note that having such a run is a weaker condition than generalised soundness.

Corollary 4.12. *If $\mathcal{W}$ is generalised sound, then it has a full reset run. Furthermore, the assumption of $\mathcal{W}$ being generalised sound can be relaxed to the weaker assumption that $\mathcal{W}$ is up-to-K-sound for an Ackermannianly bounded number K.*

Proof. The first two conditions follow directly from Lemmas 4.5 and 4.6. Let ζ be a run fulfilling these conditions, and consider a decomposition $\zeta = \rho t \rho'$ with $\{i\colon z\} \to^{\rho} \boldsymbol{m}$. We have $\{i\colon 2z\} \to^{\rho\zeta} Reset(\boldsymbol{m}) + \{f\colon z\}$. Since $\mathcal{W}$ is generalised sound or up-to-K sound for $K \geq 2z$, we get the third condition, i.e. $Reset(\boldsymbol{m}) \to^{*} \{f\colon z\}$. □

Lemma 4.13. *There is an algorithm that, given a reset workflow net $\mathcal{W}$, outputs: either that $\mathcal{W}$ has a full reset run; or that $\mathcal{W}$ is not generalised sound. The procedure works in Ackermannian time. Moreover, if there exists $\{i\colon z\} \to^{\zeta} \{f\colon z\}$ witnessing a full reset run, then there is also one where z is at most Ackermannian. If the algorithm outputs that $\mathcal{W}$ is not generalised sound then it computes $K \in \mathbb{N}_{>0}$, which is at most Ackermannian, for which $\mathcal{W}$ is not K-sound.*

Proof. If the initial place is resetable, then the algorithm outputs that $\mathcal{W}$ is not generalised sound by Claim 3. It also gives an Ackermannian bound for K such that $\mathcal{W}$ is not K-sound.

Otherwise, we invoke Remark 2 and obtain that from a marking $\{i\colon z\}$, there is a run $\{i\colon z\} \to^{\delta} \boldsymbol{m}$ that executes all nonredundant transitions of $\mathcal{W}$ and is of length at most Ackermannian. Because δ has a bounded length, we can find it in Ackermannian time. Next, we check if it is possible to cover $\{f\colon z\}$ from $\boldsymbol{m}$, using the backward coverability algorithm running in Ackermannian time. If not, then $\mathcal{W}$ is not up-to-z sound and not generalised sound. Otherwise, the algorithm produces a run $\boldsymbol{m} \to^{\delta'} \uparrow\{f\colon z\}$ of Ackermannian length. We have to check whether $\boldsymbol{m} \to^{\delta'} \{f\colon z\}$ holds. If not, then $\mathcal{W}$ is not up-to-z sound and not generalised sound. Otherwise, we pick $\zeta := \delta\delta'$.

The latter satisfies the first two conditions of a full reset run. We need to check whether it satisfies the third condition.

We do this exhaustively: for any decomposition of $\zeta = \rho\rho'$, we take a marking $\boldsymbol{m}$ such that $\{i\colon z\} \to^{\rho} \boldsymbol{m}$ and check a stronger property: whether for a run $Reset(\boldsymbol{m}) \to^{\xi} \uparrow\{f\colon z\}$ it is the case that $Reset(\boldsymbol{m}) \rightarrow^{\xi} \{f\colon z\}$. Note that this is not as trivial as for δ' since $Reset(\boldsymbol{m})$ does not have to be a reachable configuration.

We prove the property by contradiction. If it does not hold, then $\mathcal{W}$ is not generalised sound and not up-to-K sound. Indeed, due to Corollary 4.12, we know that if $\mathcal{W}$ is up-to-K sound, then it must have a full reset run $\{i\colon z'\} \rightarrow^{\zeta'} \{f\colon z'\}$. But, then, there would be a run $\{i\colon z+z'\} \rightarrow^{\rho} \boldsymbol{m} + \{i\colon z'\} \rightarrow^{\zeta'} Reset(\boldsymbol{m}) + \{f\colon z'\} \rightarrow^{\xi} \{f\colon z+z'\} + \boldsymbol{n}$ for some nonempty marking $\boldsymbol{n}$. This contradicts both generalised soundness and up-to-K soundness assuming $K > z+z'$.

To achieve this check, we use the backward coverability algorithm to find ξ, and then we execute ξ step by step. This process works in Ackermannian time. □

Below we discuss markings both in $\mathcal{W}$ and its skeleton $\mathcal{W}^s$. For convenience, if $\boldsymbol{m}_s$ is a marking over $\mathbb{N}^{P^s}$, then we also use it as a marking over $\mathbb{N}^P$, where $\boldsymbol{m}_s(p) = 0$ for $p \notin P^s$.

Lemma 4.14. *Let $\mathcal{W}$ be a reset workflow net with a full reset run $\{i\colon z\} \rightarrow^{\zeta} \{f\colon z\}$. Let $\{i\colon l\} \rightarrow_s^{\pi} \boldsymbol{m}_s$ be a run of the skeleton $\mathcal{W}^s$, where $l \in \mathbb{N}$. There exists $k' \in \mathbb{N}$ such that $\{i\colon l+k'\} \rightarrow^{*} \boldsymbol{m}_s + \{f\colon k'\}$ holds in $\mathcal{W}$. Moreover, $k' \le 2z|\pi|$.*

Before we prove Lemma 4.14, we show how it implies Proposition 4.4 (2).

Proof of Proposition 4.4 (2). Let $\mathcal{W}$ and $\mathcal{W}^s$ be as described in the proposition. Suppose that $\mathcal{W}^s$ is not generalised sound. Because of [3, Theorem 5.1], there exists an exponentially bounded number l such that $\{i\colon l\} \rightarrow_s^{*} \boldsymbol{m}_s$ in $\mathcal{W}^s$ and from $\boldsymbol{m}_s$ it is not possible to reach $\{f\colon l\}$. Observe that if such l and $\boldsymbol{m}_s$ exist then the shortest run π from $\{i\colon l\}$ to $\boldsymbol{m}_s$ is at most Ackermannian (due to bounds on the length of shortest runs in Petri nets). The bound on the length of the shortest run between two configurations in a Petri net is a consequence of the KLM [18] algorithm combined with the Ackermannian bound on its complexity [22]. By Lemma 4.14 and Corollary 4.12, there exists $k' \in \mathbb{N}$ bounded by $2z|\pi|$, i.e. Ackermannianly, such that $\{i\colon l+k'\} \rightarrow^{*} \boldsymbol{m}_s + \{f\colon k'\}$ in $\mathcal{W}$. Since $\mathcal{W}$ is generalised sound or up-to-K sound for $K > l+k'$, there is a run $\boldsymbol{m}_s \rightarrow^{*} \{f\colon l\}$ in $\mathcal{W}$. By Claim 4, this yields to the contradiction with the assumption that $\boldsymbol{m}_s \not\rightarrow_s^{*} \{f\colon l\}$ in $\mathcal{W}^s$. □

Proof of Lemma 4.14. We proceed by induction on the length of the run from $\{i\colon k\}$ to $\boldsymbol{m}_s$. If the length is 0, then the claim is trivial and $k' = 0$.

Suppose the induction claim holds for every marking $\boldsymbol{m}_s$ reachable via a run of length at most i. Let t be a transition in $\mathcal{W}^s$ and let $\pi' = \pi t$ be a run of length $i+1$ such that $\{i\colon k\} \rightarrow_s^{\pi} \boldsymbol{m}_s \rightarrow^{t} \boldsymbol{m}'_s$. By the induction hypothesis, there exists $k'' \in \mathbb{N}$ such that $\{i\colon k''+k\} \rightarrow^{*} \boldsymbol{m}_s + \{f\colon k''\}$. Thus, it is sufficient to prove that there exists $k_t \in \mathbb{N}$ such that $\{i\colon k_t\} + \boldsymbol{m}_s \rightarrow^{*} \boldsymbol{m}'_s + \{f\colon k_t\}$.

If we treat $\boldsymbol{m}_s$ and $\boldsymbol{m}'_s$ as markings over $\mathbb{N}^P$, then $Reset(\boldsymbol{m}_s) = \boldsymbol{m}_s$ and $Reset(\boldsymbol{m}'_s) = \boldsymbol{m}'_s$.

Let ζ be a full reset run in $\mathcal{W}$. Let t' be the nonredundant transition in $\mathcal{W}$ from which t, in the skeleton, is defined. Since ζ has all nonredundant transitions of $\mathcal{W}$, t' also occurs in it and we can decompose $\zeta = \rho t' \rho'$. We denote $\{i\colon z\} \rightarrow^{\rho} \boldsymbol{m}_t \rightarrow^{t'} \boldsymbol{m}'_t$. Consider the following run:

$$\begin{aligned}\{i\colon 2z\} + \boldsymbol{m}_s &\rightarrow^{\rho} \boldsymbol{m}_s + \boldsymbol{m}_t + \{i\colon z\}\\ &\rightarrow^{t'} \boldsymbol{m}'_s + Reset_{t'}(\boldsymbol{m}_t) + \boldsymbol{n} + \{i\colon z\},\end{aligned}$$

where $\boldsymbol{n}(p) = t'^{\bullet}(p)$ if $p \notin P^s$ and $\boldsymbol{n}(p) = 0$ otherwise. The last transition splits the effect of t' between $\boldsymbol{m}_s$ and $\boldsymbol{m}_t$ as follows. The places $p \in P^s$ are updated by changing $\boldsymbol{m}_s$ to $\boldsymbol{m}'_s$. The remaining places are updated by changing $\boldsymbol{m}_t$ to $Reset_{t'}(\boldsymbol{m}_t) + \boldsymbol{n}$. Note that $Reset(\boldsymbol{m}_t) = Reset(Reset_{t'}(\boldsymbol{m}_t) + \boldsymbol{n})$. Thus, we get

$$\{i\colon z\} + \boldsymbol{m}'_s + Reset_{t'}(\boldsymbol{m}_t) + \boldsymbol{n} \rightarrow^{\zeta} \boldsymbol{m}'_s + Reset(\boldsymbol{m}_t) + \{f\colon z\}.$$

Finally, recall that by definition of full reset runs (third condition) $Reset(\boldsymbol{m}_t) \rightarrow^{*} \{f\colon z\}$. Altogether, we get the following as required:

$$\{i\colon 2z\} + \boldsymbol{m}_s \rightarrow^{*} \boldsymbol{m}'_s + \{f\colon 2z\}.$$

The bound on k' follows directly from the proof. □

4.5 Proofs of Theorems 4.1 and 4.2

We may now prove Theorems 4.1 and 4.2 simultaneously. The constant k' in Theorem 4.2 will be defined at the end of the proof. There, we will observe that if k is large enough then the characterization of $\mathcal{P}_k$ in Theorem 4.2 holds.

Fix $k \in \mathbb{N}_{>0}$. We start with the definition of property $\mathcal{P}_k$.

Definition 4.15. Property $\mathcal{P}_k$ is defined as the conjunction of these properties:

1. Places i and f are not resetable;
2. There is a full reset run $\{i\colon z\} \rightarrow^{\zeta} \{f\colon z\}$ in $\mathcal{W}$;
3. $\mathcal{W}^s$ is a workflow net which is generalised sound;
4. It is not possible to strictly cover $\{f\colon k\}$ starting from $\{i\colon k\}$, where "strictly cover" means reaching some marking $\boldsymbol{m} > \{f\colon k\}$. We call this property *coverability-clean*;
5. The last property is more complex. Consider the following set of markings:
$$X \coloneqq \left\{\boldsymbol{m} \in \mathbb{N}^P \setminus \{\boldsymbol{0}\} : \boldsymbol{m} \not\rightarrow^{*} \uparrow\{f\colon 1\}\right\}.$$
In words, these are the nonzero markings that cannot mark place f. Let $F \coloneqq \{\{f\colon \ell\} : \ell \in \mathbb{N}\}$ and $X^f \coloneqq X + F$, i.e., markings of X with arbitrarily many tokens added to f. Let $Reset(X^f) \coloneqq \{Reset(\boldsymbol{m}) : \boldsymbol{m} \in X^f\}$. The last property requires that $\{i\colon j\} \not\rightarrow_s^{*} Reset(X^f) \setminus F$ holds for all $j \ge 1$.

First observe that properties (1–4) are implied by generalised soundness, as well as up-to-k soundness for sufficiently large k. It is clear for (4), while properties (1–3) follow respectively from Claim 3, Corollary 4.12 and Proposition 4.4.

To prove that $\mathcal{P}_k$ is k-in-between, it suffices to show that for any workflow satisfying properties (1–4), these two claims, capturing property (5), hold:

Claim 7. *If $\{i\colon j\} \rightarrow_s^{*} Reset(X^f) \setminus F$ holds for some $j \ge 1$, then $\mathcal{W}$ is not generalised sound.*

Claim 8. *If $\{i\colon j\} \not\rightarrow_s^{*} Reset(X^f) \setminus F$ holds for all $j \in [1..k]$, then $\mathcal{W}$ is up-to-k sound.*

Before proceeding, we show the claim below, which will be helpful for proving Claim 7 and Claim 8. Let us assume that properties (1–4) hold.

Claim 9. *$\mathcal{W}$ is not k-sound if and only if $\{i\colon k\} \rightarrow^* X^f$.*

Proof: ⇒) Suppose $\mathcal{W}$ is not k-sound and let $\{i\colon k\} \rightarrow^* \boldsymbol{m}$ be such that $\boldsymbol{m} \not\rightarrow^* \{f\colon k\}$. Let ℓ be the largest number such that from $\boldsymbol{m}$ we can cover $\{f\colon \ell\}$. Note that such a number exists and $\ell \leq k$, as otherwise we get a contradiction with Claim 4 and generalised soundness of $\mathcal{W}^s$. Let $\boldsymbol{m} \rightarrow^* \boldsymbol{m'} + \{f\colon \ell\}$, where $\boldsymbol{m'}(f) = 0$. If $\boldsymbol{m'} \neq \boldsymbol{0}$, then we are done as $\boldsymbol{m'} \in X$. Otherwise, note that $\ell < k$ as $\boldsymbol{m} \not\rightarrow^* \{f\colon k\}$, which yields a contradiction with Claim 4 and generalised soundness of $\mathcal{W}^s$.

⇐) This follows by definition of X and by the fact that, in reset workflow nets, the effect of firing transitions cannot be zero. ■

Let $X_{\boldsymbol{0}} \coloneqq X \cup \{\boldsymbol{0}\}$. We may now prove Claim 7:

Proof: Let $\{i\colon j\} \rightarrow_s^* \boldsymbol{m} + \{f\colon \ell\} \in \mathit{Reset}(X^f) \setminus F$, where $\boldsymbol{m}(f) = 0$. Note that $\boldsymbol{m} \neq \boldsymbol{0}$ as the set F was excluded. By Lemma 4.14, there exists k' such that $\{i\colon j{+}k'\} \rightarrow^* \boldsymbol{m}{+}\{f\colon \ell{+}k'\}$. As $X_{\boldsymbol{0}}$ is downward closed and $\boldsymbol{m} \neq \boldsymbol{0}$, we get $\boldsymbol{m} + \{f\colon \ell + k'\} \in X^f$. Thus, by Claim 9, $\mathcal{W}$ is not $(j + k')$-sound, and hence it is not generalised sound. ■

We may now prove Claim 8:

Proof: We show the contrapositive. For the sake of contradiction, suppose $\mathcal{W}$ is not up-to-k sound, i.e. not j-sound for some $j \in [1..k]$. We must exhibit a run $\{i\colon j\} \rightarrow_s^* \mathit{Reset}(X^f) \setminus F$.

By Claim 9, we have $\{i\colon j\} \rightarrow^* \boldsymbol{m} + \{f\colon \ell\} \in X^f$ in $\mathcal{W}$, where $\boldsymbol{m}(f) = 0$. By Claim 4, we have $\{i\colon j\} \rightarrow_s^* \mathit{Reset}(\boldsymbol{m}) + \{f\colon \ell\}$ in $\mathcal{W}^s$. If $\mathit{Reset}(\boldsymbol{m}) \neq \boldsymbol{0}$, then we get a contradiction since

$$\mathit{Reset}(\boldsymbol{m}) + \{f\colon \ell\} \in \mathit{Reset}(X^f) \setminus F.$$

Suppose $\mathit{Reset}(\boldsymbol{m}) = \boldsymbol{0}$. From this, we have $\{i\colon j\} \rightarrow_s \{f\colon l\}$ which is possible only if $\ell = j$, as otherwise we get a contradiction with $\mathcal{W}^s$ being generalised sound. This means that $\boldsymbol{m} > \{f\colon j\}$, which is a contradiction with $\mathcal{W}$ being coverability-clean. ■

To conclude the proof, it suffices to show that $\mathcal{P}_k$ is decidable, i.e. that properties (1–5) can be checked. First, computing $\mathcal{W}^s$ amounts to identifying the subset of nonredundant transitions of $\mathcal{W}$, which can be done in Ackermannian time by Proposition 4.3. Then:

1. Property (1) is trivial;
2. Property (2) is decidable because of Lemma 4.13;
3. Testing whether $\mathcal{W}^s$ is generalised sound can be done in PSPACE [3, Theorem 5.1];
4. Property (4) is a coverability check, which can be done using the backward coverability algorithm [19] (or by Lemma 2.3);
5. Property (5) requires more effort. We explain it below.

Let us show that we can determine whether $\{i\colon k\} \rightarrow^* X^f$. We start by computing a representation of X^f. A representation of $X_{\boldsymbol{0}}$ can be computed with the backwards coverability algorithm [19]. More precisely, we can compute the set X' of all markings from which there is a run covering $\{f : 1\}$. Then, $X_{\boldsymbol{0}}$ is the complement of X'. Moreover, since X' is upward closed, the set $X_{\boldsymbol{0}}$ is downward closed. Obviously, this yields a representation of both X and X^f.

To simplify the notation, we also identify $\mathit{Reset}(X^f)$ with the set of markings over P^s (i.e. by dropping the places outside of P^s). Recall $F = \{\{f\colon \ell\} : \ell \in \mathbb{N}\}$ from the definition of X^f.

It remains to show that we can decide whether there exists $j \geq 1$ such that:

$$\{i\colon j\} \rightarrow_s^* \mathit{Reset}(X^f) \setminus F \text{ holds in } \mathcal{W}^s. \qquad (\star)$$

Moreover, we must prove that if such a j exists, then it is Ackermannianly bounded. This will allow to prove Theorem 4.2.

We reduce query ($\star$) to a reachability query for Petri nets (without resets). To do so, we modify $\mathcal{W}^s$ into a new Petri net $\mathcal{N}^s$ (whose arcs may consume or produce several tokens at once). First, we add a transition t_i that can always add one token in place i, this allows to produce arbitrarily many tokens in i. Second, we add a place p_{all} that keeps the sum of tokens in all places from $P^s \setminus \{f\}$ (it will be needed as X forbids $\boldsymbol{0}$). This can be easily achieved by adjusting all transitions on p_{all} as follows:

$$^{\bullet}t(p_{\text{all}}) \coloneqq \sum_{p \in P^s \setminus \{f\}} {}^{\bullet}t(p) \qquad \text{and} \qquad t^{\bullet}(p_{\text{all}}) \coloneqq \sum_{p \in P^s \setminus \{f\}} t^{\bullet}(p).$$

We define X^f_{all} as the set

$$\big\{\boldsymbol{m'} \in \mathbb{N}^{P^s \cup \{p_{\text{all}}\}} : \exists \boldsymbol{m} \in X \text{ s.t. } \boldsymbol{m'}(p) = \boldsymbol{m}(p) \text{ for all } p \in P^s \\ \text{and } \boldsymbol{m'}(p_{\text{all}}) \geq \boldsymbol{m}(P^s \setminus \{f\})\big\}.$$

Note that if $\boldsymbol{m'} \in X^f_{\text{all}}$, then we have $\boldsymbol{m'}(p_{\text{all}}) > 0$. Note that only markings such that $\boldsymbol{m'}(p_{\text{all}}) = \boldsymbol{m}(P^s \setminus \{f\})$ make sense, but it will be convenient to allow markings to be larger in place p_{all}.

Observe that query ($\star$) is equivalent to testing whether $\boldsymbol{0} \rightarrow^* X^f_{\text{all}}$ in $\mathcal{N}^s$. Indeed, transition t_i allows to guess the initial value j, and place p_{all} guarantees that we at least one token among places other than f.

Now we analyse the set X^f_{all}. Let $\boldsymbol{m} \in X^f_{\text{all}}$. The following holds:

- if $\boldsymbol{m'} > \boldsymbol{m}$ and $\boldsymbol{m'}(p) = \boldsymbol{m}(p)$ for $p \in P^s \setminus \{f\}$ then $\boldsymbol{m'} \in X^f_{\text{all}}$;
- if $\boldsymbol{m'} < \boldsymbol{m}$ and $\boldsymbol{m'}(p) = \boldsymbol{m}(p)$ for $p \in \{f, p_{\text{all}}\}$ then $\boldsymbol{m'} \in X^f_{\text{all}}$.

Intuitively, X^f_{all} is downward closed on some places and upward closed on other places. Since $\mathcal{N}^s$ is a Petri net (without resets), it is folklore that reachability queries to such sets can be performed in Ackermannian time (see e.g. [6, Lemma 7]). Moreover, if there is such a run, then there is one of length at most Ackermannian. This concludes the proof of Theorem 4.1. It also provides an Ackermannian bound on the minimal j satisfying query ($\star$).

We briefly explain that it also proves Theorem 4.2. Indeed, let us comment on the threshold k' such that, for any $k \geq k'$, $\mathcal{P}_k$ is equivalent to up-to-k-soundness. Observe that properties (1), (2), (3) and (5) do not depend on k, so intuitively there is a k' such that if they are satisfied for k', then they are satisfied for all $k > k'$. So, properties (1), (2), (3) and (5) are implied by up-to-k soundness for $k > k'$. Moreover, property (4) is also implied by up-to-k soundness, which means that $\mathcal{P}_k$, for $k > k'$, is implied by up-to-k soundness.

What remains is to show that an Ackermannianly bounded k' suffices. Property (1) is implied by up-to-k' soundness for an Ackermannianly bounded k' according to Claim 3. Similarly, property (2) is implied by up-to-k' soundness for an Ackermannianly bounded k' according to Lemma 4.5. Property (3) is implied by up-to-k' soundness for an Ackermannianly bounded k' according to Proposition 4.4. Thus, it remains to bound the number k' needed for property (5). We know that, if there is a run that violates property (5), then there is one of length ℓ which is at most Ackermannian. Now, because of Lemma 4.14, we conclude that there is a run $\{i\colon \ell + \ell \cdot 2z\} \rightarrow^* X^f$, where z is Ackermannianly bounded as in Lemma 4.14. This, together with Claim 9, shows that $k' > \ell + \ell \cdot 2z$ suffices. Altogether, an Ackermannianly bounded k' suffices for the proof of Theorem 4.2.

Remark 3. *One may think that the proof of Theorem 4.1 is contradictory with the undecidability of generalised soundness, as it might seem that, using Claim 7 and Claim 8, we can decide generalised soundness. The reason why there is no contradiction is that, earlier, we assumed that $\mathcal{W}$ is coverability-clean. In some sense, checking the coverability-clean property, for all k, is the source of undecidability for generalised soundness.*

5 CONCLUSION

In this paper, we studied soundness in reset workflow nets: the standard correctness notion of a well-established formalism for the modeling of process activities such as business processes.

All existing variants of soundness, but generalised soundness, were known to be undecidable for reset workflow nets. In this work, we have shown that generalised soundness is also undecidable. This closes its status which had been open for over fifteen years.

Given the resulting undecidable landscape, we investigated a new approach. We introduced the notion of k-in-between soundness, which lies between k-soundness and generalised soundness. We revealed an unusual complexity behaviour: a decidable soundness property is in between two undecidable ones. We think this can be valuable in the algorithmic analysis of reset workflow nets, and that it may spark a new line of research both in theory and practice.

5.1 Other future work

The reachability problem for Minsky machines is already undecidable for two transitions that test counters for zero. Thus, our proof of the undecidability of generalised soundness only requires four transitions that reset some places. The question about decidability of generalised soundness for reset workflow nets with fewer than four transitions that reset some places, remains open. We conjecture decidability for reset workflow net with only one such transition.

Furthermore, it would be interesting to extend the definition of soundness to more powerful models like well-structured transition systems (WSTS): the properties of resilience [10] can be seen as a first step. We may also try to adapt the efficient reductions for Petri nets [2] to reset Petri nets and reset workflow nets.

REFERENCES

[1] S. Akshay, Supratik Chakraborty, Ankush Das, Vishal Jagannath, and Sai Sandeep. 2017. On Petri Nets with Hierarchical Special Arcs. In *Proc. 28th International Conference on Concurrency Theory (CONCUR) (LIPIcs, Vol. 85)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 40:1–40:17. https://doi.org/10.4230/LIPICS.CONCUR.2017.40

[2] Nicolas Amat, Silvano Dal-Zilio, and Didier Le Botlan. 2024. Project and Conquer: Fast Quantifier Elimination for Checking Petri Net Reachability. In *Proc. 25th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI) (Lecture Notes in Computer Science, Vol. 14499)*. Springer, 101–123. https://doi.org/10.1007/978-3-031-50524-9_5

[3] Michael Blondin, Filip Mazowiecki, and Philip Offtermatt. 2022. The complexity of soundness in workflow nets. In *Proc. 37th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. 20:1–20:13. https://doi.org/10.1145/3531130.3533341

[4] Dmitry Chistikov, Wojciech Czerwinski, Piotr Hofman, Filip Mazowiecki, and Henry Sinclair-Banks. 2023. Acyclic Petri and Workflow Nets with Resets. In *Proc. 43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS) (LIPIcs, Vol. 284)*. 16:1–16:18. https://doi.org/10.4230/LIPICS.FSTTCS.2023.16

[5] 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

[6] Wojciech Czerwinski and Piotr Hofman. 2022. Language Inclusion for Boundedly-Ambiguous Vector Addition Systems Is Decidable. In *Proc. 33rd International Conference on Concurrency Theory (CONCUR) (LIPIcs, Vol. 243)*. 16:1–16:22. https://doi.org/10.4230/LIPICS.CONCUR.2022.16

[7] Wojciech Czerwinski and Lukasz Orlikowski. 2021. Reachability in Vector Addition Systems is Ackermann-complete. In *Proc. 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*. 1229–1240. https://doi.org/10.1109/FOCS52979.2021.00120

[8] Catherine Dufourd, Alain Finkel, and Philippe Schnoebelen. 1998. Reset Nets Between Decidability and Undecidability. In *Proc. 25th International Colloquium on Automata, Languages and Programming (ICALP) (Lecture Notes in Computer Science, Vol. 1443)*. Springer, 103–115. https://doi.org/10.1007/BFB0055044

[9] Diego Figueira, Santiago Figueira, Sylvain Schmitz, and Philippe Schnoebelen. 2011. Ackermannian and Primitive-Recursive Bounds with Dickson's Lemma. In *Proc. 26th Annual IEEE Symposium on Logic in Computer Science (LICS)*. IEEE Computer Society, 269–278. https://doi.org/10.1109/LICS.2011.39

[10] Alain Finkel and Mathieu Hilaire. 2024. Resilience and Home-Space for WSTS. In *Proc. 25th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI) (Lecture Notes in Computer Science, Vol. 14499)*. Springer, 147–168. https://doi.org/10.1007/978-3-031-50524-9_7

[11] Alain Finkel, Jérôme Leroux, and Grégoire Sutre. 2018. Reachability for Two-Counter Machines with One Test and One Reset. In *Proc. 38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS) (LIPIcs, Vol. 122)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 31:1–31:14. https://doi.org/10.4230/LIPICS.FSTTCS.2018.31

[12] Alain Finkel, Pierre McKenzie, and Claudine Picaronny. 2004. A well-structured framework for analysing Petri net extensions. *Information and Computation* 195, 1 (2004), 1–29. https://doi.org/10.1016/j.ic.2004.01.005

[13] Alain Finkel and Philippe Schnoebelen. 2001. Well-structured transition systems everywhere! *Theoretical Computer Science* 256, 1-2 (2001), 63–92. https://doi.org/10.1016/S0304-3975(00)00102-X

[14] Martin Grohe and Pascal Schweitzer. 2020. The graph isomorphism problem. *Commun. ACM* 63, 11 (oct 2020), 128–134. https://doi.org/10.1145/3372123

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

[16] Piotr Hofman, Slawomir Lasota, Richard Mayr, and Patrick Totzke. 2016. Simulation Problems Over One-Counter Nets. *Logical Methods in Computer Science* 12, 1 (2016). https://doi.org/10.2168/LMCS-12(1:6)2016

[17] Anna A. Kalenkova and Irina A. Lomazova. 2014. Discovery of Cancellation Regions within Process Mining Techniques. *Fundamenta Informaticae* 133, 2-3 (2014), 197–209. https://doi.org/10.3233/FI-2014-1071

[18] S. Rao Kosaraju. 1982. Decidability of Reachability in Vector Addition Systems (Preliminary Version). In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber (Eds.). ACM, 267–281. https://doi.org/10.1145/800070.802201

[19] Ranko Lazić and Sylvain Schmitz. 2021. The ideal view on Rackoff's coverability technique. *Information and Computation* 277 (2021), 104582. https://doi.org/10.1016/J.IC.2020.104582

[20] Jérôme Leroux. 2021. The Reachability Problem for Petri Nets is Not Primitive Recursive. In *Proc. 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*. 1241–1252. https://doi.org/10.1109/FOCS52979.2021.00121

[21] 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)*. 1–13. https://doi.org/10.1109/LICS.2019.8785796

[22] Jérôme Leroux and Sylvain Schmitz. 2019. Reachability in Vector Addition Systems is Primitive-Recursive in Fixed Dimension. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*. IEEE, 1–13. https://doi.org/10.1109/LICS.2019.8785796

[23] Martin H Löb and Stanley S Wainer. 1970. Hierarchies of number-theoretic functions. I. *Archiv für mathematische Logik und Grundlagenforschung* 13, 1-2 (1970), 39–51.

[24] Richard Mayr. 2003. Undecidability of Weak Bisimulation Equivalence for 1-Counter Processes. In *Proc. 30th International Colloquium on Automata, Languages and Programming (ICALP)*, Vol. 2719. 570–583. https://doi.org/10.1007/3-540-45061-0_46

[25] Robert Paige and Robert Endre Tarjan. 1987. Three Partition Refinement Algorithms. *SIAM J. Comput.* 16, 6 (1987), 973–989. https://doi.org/10.1137/0216062

[26] Wolfgang Reisig. 2013. *Understanding Petri Nets - Modeling Techniques, Analysis Methods, Case Studies.* Springer. https://doi.org/10.1007/978-3-642-33278-4

[27] Sylvain Schmitz. 2016. Complexity Hierarchies beyond Elementary. *ACM Transactions on Computation Theory* 8, 1 (2016), 3:1–3:36. https://doi.org/10.1145/2858784

[28] Leslie G. Valiant. 1973. *Decision procedures for families of deterministic pushdown automata.* Ph. D. Dissertation. University of Warwick, Coventry, UK. http://wrap.warwick.ac.uk/34701/

[29] WMP van der Aalst, KM van Hee, AHM ter Hofstede, N Sidorova, HMW Verbeek, M Voorhoeve, and MT Wynn. 2008. Soundness of workflow nets with reset arcs is undecidable!. In *Concurrency methods: issues and applications (CHINA).* 57.

[30] 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

[31] 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

[32] 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. 2009. Soundness of Workflow Nets with Reset Arcs. *Transactions on Petri Nets and Other Models of Concurrency* 3 (2009), 50–70. https://doi.org/10.1007/978-3-642-04856-2_3

[33] 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

[34] Robert van der Toorn. 2004. *Component-Based Software Design with Petri nets: An Approach Based on Inheritance of Behavior.* Ph. D. Dissertation. Technische Universiteit Eindhoven. https://doi.org/10.6100/IR571325

[35] 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

[36] 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

[37] Moe Thandar Wynn, H. M. W. Verbeek, Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, and David Edmond. 2009. Reduction rules for YAWL workflows with cancellation regions and OR-joins. *Information & Software Technology* 51, 6 (2009), 1010–1020. https://doi.org/10.1016/J.INFSOF.2008.12.002

[38] Moe Thandar Wynn, H. M. W. Verbeek, Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, and David Edmond. 2009. Soundness-preserving reduction rules for reset workflow nets. *Information Sciences* 179, 6 (2009), 769–790. https://doi.org/10.1016/J.INS.2008.10.033