

On Petri Nets with Hierarchical Special Arcs

S. Akshay¹, Supratik Chakraborty², Ankush Das³,
Vishal Jagannath⁴, and Sai Sandeep⁵

1 Department of Computer Science and Engineering, IIT Bombay, India

2 Department of Computer Science and Engineering, IIT Bombay, India

3 Computer Science Department, Carnegie Mellon University, Pittsburgh, PA,
USA

4 Department of Computer Science and Engineering, IIT Bombay, India

5 Department of Computer Science and Engineering, IIT Bombay, India

Abstract

We investigate the decidability of termination, reachability, coverability and deadlock-freeness of Petri nets endowed with a hierarchy of places, and with inhibitor arcs, reset arcs and transfer arcs that respect this hierarchy. We also investigate what happens when we have a mix of these special arcs, some of which respect the hierarchy, while others do not. We settle the decidability status of the above four problems for all combinations of hierarchy, inhibitor, reset and transfer arcs, except the termination problem for two combinations. For both these combinations, we show that deciding termination is as hard as deciding the positivity problem for linear recurrence sequences – a long-standing open problem.

1998 ACM Subject Classification F.1.1 Models of Computation, D.2.2 Design Tools and Techniques: Petri nets

Keywords and phrases Petri Nets, Hierarchy, Reachability, Coverability, Termination, Positivity

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2017.40

1 Introduction

Petri nets are an important and versatile mathematical modeling formalism for distributed and concurrent systems. Thanks to their intuitive visual representation, precise execution semantics, well-developed mathematical theory and availability of tools for reasoning about them, Petri nets are used in varied contexts, viz. computational, chemical, biological, workflow-related etc. Several extensions to Petri nets have been proposed in the literature to augment their modeling power. From a theoretical perspective, these provide rich and interesting models of computation that warrant investigation of their expressive powers, and decidability and/or complexity of various decision problems. From a practitioner's perspective, they enable new classes of systems to be modeled and reasoned about.

In this paper, we focus on an important class of extensions proposed earlier for Petri nets, pertaining to the addition of three types of *special* arcs, namely *inhibitor*, *reset* and *transfer* arcs from places to transitions. We investigate how different combinations of these extensions affect the decidability of four key decision problems: reachability, coverability, termination, and deadlock-freeness. To start with, an inhibitor arc effectively models a zero test, and hence one can model two-counter machines with two inhibitor arcs, leading to undecidability of all of the above decision problems. However, Reinhardt [20] showed that if we impose a *hierarchy* among places with inhibitor arcs (a single inhibitor arc being a sub-case), we recover decidability of reachability. Recently Bonnet [5] simplified this proof using techniques of Leroux [14] and also showed that termination and coverability are decidable



© S. Akshay, Supratik Chakraborty, Ankush Das, Vishal Jagannath, and Sai Sandeep;
licensed under Creative Commons License CC-BY

28th International Conference on Concurrency Theory (CONCUR 2017).

Editors: Roland Meyer and Uwe Nestmann; Article No. 40; pp. 40:1–40:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

for Petri nets with hierarchical inhibitor arcs. With reset arcs (which remove all tokens from a pre-place) and transfer arcs (which transfer all tokens from a pre-place to a post-place), reachability and deadlock-freeness are known to be undecidable [9], although termination and coverability are decidable [11].

In this paper, we are interested in what happens when *hierarchy* is introduced among all combinations of special arcs. Thus, we specify a hierarchy, or total ordering, of the places, and say that the special arcs respect the hierarchy if whenever there is a special arc from a place p to a transition t , there are also special arcs from every place lower than p in the hierarchy to t . The study of Petri nets extended with hierarchical and non-hierarchical special arcs provides a generic framework that subsumes several existing questions and raises new ones. There are only a handful of results in the literature where hierarchical special arcs have been shown to play an important role. Decidability of reachability for Petri nets with hierarchical inhibitor arcs was shown in [20] and re-visited in a special context in [4], while decidability of termination, coverability and boundedness were shown in [5]. Further, in [2] it was shown that Petri nets with hierarchical zero tests are equivalent to Petri nets with a stack encoding restricted context-free languages. Finally a specific subclass, namely Petri nets with a single inhibitor arc, has received a lot of attention, with results showing decidability of boundedness and termination [10], place-boundedness [6], and LTL model checking [7]. However, in [7], the authors remark that it would not be easy to extend their technique for the last two problems to handle hierarchical arcs. To the best of our knowledge, none of the earlier papers address mixing of reset and transfer arcs within the hierarchy of inhibitor arcs, leaving several interesting questions unanswered. Our primary goal in this paper is to comprehensively fill these gaps. Before delving into the theoretical investigations, we present two examples that illustrate why these models are interesting from a practical point of view.

Our first example is a prioritized job-shop environment in which work stations with possibly different resources are available for servicing jobs. Each job comes with a priority and with a requirement of the count of resources it needs. For simplicity, assume that all resources are identical, and that there is at most one job with any given priority. A work station can service multiple jobs simultaneously subject to availability of resources; however, a job cannot be split across multiple work stations. Additionally, we require that a job with a lower priority must not be scheduled on any work station as long as a job with higher priority is waiting to be scheduled. Once a job gets done, it can either terminate or generate additional jobs with different priorities based on some rules. An example of such a rule could be that a job with priority k and resource requirement m can only generate a new job with priority $\leq k$ and resource requirement $\leq m$. Given such a system, there are several interesting questions one might ask. For example, can too many jobs (above a specified threshold) of the lowest priority be left waiting for a work station? Or, can the system reach a deadlocked state from where no progress can be made? A possible approach to answering these questions is to model this prioritized job-shop environment as a Petri net with hierarchical special arcs, i.e., resets, inhibitors and transfers, and reduce the questions to decision problems (such as coverability or deadlock-freeness) for the corresponding nets.

Our second example builds on work reported in the literature on modeling integer programs with loops using Petri nets [3]. Questions pertaining to termination of such programs can be reduced to decision problems (termination or deadlock-freeness) of the corresponding Petri net model. In Section 6.1, we describe a new reduction of the termination question for integer linear loop programs to the termination problem for Petri nets with hierarchical inhibitor and transfer arcs. This underlines the importance of studying decision problems for these extensions of Petri nets.

Our main contribution is a comprehensive investigation into Petri nets extended with a mix of these special arcs, some of which respect the hierarchy, while others do not. We settle the decidability status of the four decision problems for *all* combinations of hierarchy, inhibitor, reset and transfer arcs, except the termination problem for two combinations. For these cases, we show a reduction from the positivity problem [18, 19] – a long-standing open problem on linear recurrences. We summarize these results in Section 3, after introducing appropriate notations in Section 2. Interestingly, several of our results use completely different constructions and proof techniques, as detailed in Sections 4–6.

2 Preliminaries

We begin by recalling some key definitions and fixing notations. A Petri net, denoted PN, is defined as (P, T, F, M_0) , where P is a set of *places*, T is a set of *transitions*, $M_0 : P \rightarrow \mathbb{N}$ is the *initial marking*, and $F : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ is the *flow relation*. For every $x \in P \cup T$, we define $Pre(x) = \{y \in P \cup T \mid F(y, x) > 0\}$ and $Post(x) = \{y \in P \cup T \mid F(x, y) > 0\}$. For every $t \in T$, we use the following terminology: every $p \in Pre(t)$ is a *pre-place* of t , every $q \in Post(t)$ is a *post-place* of t , every arc (p, t) such that $F(p, t) > 0$ is a *pre-arc* of t , and every arc (t, p) such that $F(t, p) > 0$ is a *post-arc* of t .

A *marking* $M : P \rightarrow \mathbb{N}$ is a function from the set of places to non-negative integers. We say that a transition t is *firable* at marking M , denoted by $M \xrightarrow{t}$, if $\forall p \in Pre(t), M(p) \geq F(p, t)$. If t is firable at M_1 , we say that firing t gives the marking M_2 , where $\forall p \in P, M_2(p) = M_1(p) - F(p, t) + F(t, p)$. This is also denoted as $M_1 \xrightarrow{t} M_2$. We define the sequence of transitions $\rho = t_1 t_2 t_3 \dots t_n$ to be a *run* from marking M_0 , if there exist markings M_1, M_2, \dots, M_n , such that for all i , t_i is firable at M_{i-1} and $M_{i-1} \xrightarrow{t_i} M_i$. Finally, we abuse notation and use \leq to denote the component-wise ordering over markings. Thus, $M_1 \leq M_2$ iff $\forall p \in P, M_1(p) \leq M_2(p)$. A detailed account on Petri nets can be found in [17].

We now define some classical decision problems in the study of Petri nets.

► **Definition 2.1.** Given a Petri net $N = (P, T, F, M_0)$,

- **Termination** (or TERM): Does there exist an infinite run from marking M_0 ?
- **Reachability** (or REACH): Given a marking M , is there a run from M_0 which reaches M ?
- **Coverability** (or COVER): Given a marking M , is there a marking $M' \geq M$ which is reachable from M_0 ?
- **Deadlock-freeness** (or DLFREE): Does there exist a marking M reachable from M_0 , such that no transition is firable at M ?

Since Petri nets are well-structured transition systems (WSTS), the decidability of coverability and termination for Petri nets follows from the corresponding results for WSTS [11]. The decidability of reachability was shown in [13]. Subsequently, there have been several alternative proofs of the same result, viz. [14]. Finally, since deadlock-freeness reduces to reachability in Petri nets [8], all the four decision problems are decidable for Petri nets. In the remainder of the paper, we concern ourselves with these decision problems for Petri nets extended with the following special arcs:

- An **Inhibitor arc** from place p to transition t signifies t is firable only if p has zero tokens.
- A **Reset arc** from place p to transition t signifies that p contains zero tokens after t fires.
- A **Transfer arc** from place p_1 through transition t to place p_2 signifies that on firing transition t , all tokens from p_1 get transferred to p_2 .

For Petri nets with special arcs, we redefine the flow relation as $F : (P \times T) \cup (T \times P) \rightarrow \mathbb{N} \cup \{I, R\} \cup \{S_p \mid p \in P\}$, where $F(p, t) = I$ (resp. $F(p, t) = R$) signifies the presence of an

inhibitor arc (resp. reset arc) from place p to transition t . Similarly, if $F(p, t) = S_{p'}$, then there is a transfer arc from place p to place p' through transition t .

3 Problem statements and main results

We use PN to denote standard Petri nets, and I-PN, R-PN and T-PN to denote Petri nets with inhibitor, reset and transfer arcs, respectively. The following definition subsumes several additional extensions studied in this paper.

► **Definition 3.1.** A Petri net with *hierarchical special arcs* is defined to be a 5-tuple $(P, T, F, \sqsubseteq, M_0)$, where P is a set of places, T is a set of transitions, \sqsubseteq is a total ordering over P encoding the hierarchy, $M_0 : P \rightarrow \mathbb{N}$ is the initial marking, and $F : (P \times T) \cup (T \times P) \rightarrow \mathbb{N} \cup \{I, R\} \cup \{S_p \mid p \in P\}$ is a flow relation satisfying

- $\forall (t, p) \in T \times P, F(t, p) \in \mathbb{N}$, and
- $\forall (p, t) \in P \times T, F(p, t) \notin \mathbb{N} \implies (\forall q \sqsubseteq p, F(q, t) \notin \mathbb{N})$

Thus, all arcs (or edges) from transitions to places are as in standard Petri nets. However, we may have special arcs from places to transitions. These can be inhibitor arcs ($F(p, t) = I$), reset arcs ($F(p, t) = R$), or transfer arcs ($F(p, t) = S_{p'}$, where p and p' are places in the Petri net). Note that all special arcs respect the hierarchy specified by \sqsubseteq . In other words, if there is a special arc from a place p to a transition t , there must also be special arcs from every place p' to t , where $p' \sqsubseteq p$. Depending on the subset of special arcs that are present, we can define sub-classes of Petri nets with hierarchical special arcs as follows. In the following, $\text{Range}(F)$ denotes the range of the flow relation F .

► **Definition 3.2.** The class of Petri nets with hierarchical special arcs, where $\text{Range}(F) \setminus \mathbb{N}$ is a subset of $\{I\}, \{T\}$ or $\{R\}$ is called HIPN, HTPN or HRPN respectively. Similarly, it is called HITPN, HIRPN or HTRPN if $\text{Range}(F) \setminus \mathbb{N}$ is a subset of $\{I, T\}, \{I, R\}$ or $\{T, R\}$ respectively. Finally, if $\text{Range}(F) \setminus \mathbb{N} \subseteq \{I, R, T\}$, we call the corresponding class HIRTPN.

We also study generalizations, in which extra inhibitor, reset and/or transfer arcs that do not respect the hierarchy specified by \sqsubseteq , are added to PN with hierarchical special arcs.

► **Definition 3.3.** Let \mathcal{N} be a class of Petri nets with hierarchical special arcs as in Definition 3.2, and let \mathcal{M} be a subset of $\{I, T, R\}$. We use $\mathcal{M}\text{-}\mathcal{N}$ to denote the class of nets obtained by adding unrestricted special arcs of type \mathcal{M} to an underlying net in the class \mathcal{N} .

For example, R-HIPN is the class of Petri nets with hierarchical inhibitor arcs extended with reset arcs that need not respect the hierarchy. Clearly, if the special arcs in every net $N \in \mathcal{N}$ are from \mathcal{M} , the class $\mathcal{M}\text{-}\mathcal{N}$ is simply the class of Petri nets with unrestricted (no hierarchy) arcs of type \mathcal{M} . Hence we avoid discussing such extensions in the remainder of the paper.

As we show later, all four decision problems of interest to us are either undecidable or not known to be decidable for HIRTPN. A slightly constrained version of HIRTPN, however, turns out to be much better behaved, motivating the following definition.

► **Definition 3.4.** The sub-class HIRcTPN is defined to be HIRTPN with the added restriction that $\forall (p, t, p') \in P \times T \times P, F(p, t) = S_{p'} \implies F(p', t) \in \mathbb{N}$.

Thus, every place p' that has an incoming transfer arc through transition t is constrained to have only a standard PN outgoing arc (if any) to t . This restriction suffices to recover decidability for all four decision problems of interest to us. Since nets in HIRcTPN often suffice to model useful classes of systems, we present results for this class separately.

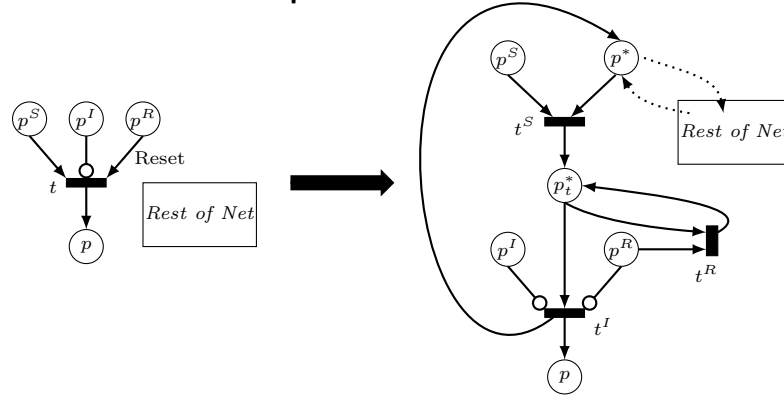
■ **Table 1** Summary of key results; results for all other extensions are subsumed by these results.

	TERM	COVER	REACH	DLFREE
PN	✓ [11]	✓ [11]	✓ [15, 14]	✓ [8, 12]
R/T-PN	✓ [11]	✓ [11]	✗ [9]	✗ [Red. from [9]]
I-PN	✗ [16]	✗ [16]	✗ [16]	✗ [16]
HIPN	✓ [20, 5]	✓ [20, 5]	✓ [20, 5]	✓ [Thm 4.3]
HTPN	✓ [11]	✓ [11]	✗ [Thm 6.1]	✗ [Thm 6.1]
HIRPN	✓ [Thm 4.2]	✓ [Thm 4.2]	✓ [Thm 4.2]	✓ [Thm 4.3]
HITPN	Positivity-Hard [Thm 6.5]	✗ [Cor. 6.2]	✗ [Thm 6.1]	✗ [Thm 6.1]
HIRcTPN	✓ [Thm 4.2]	✓ [Thm 4.2]	✓ [Thm 4.2]	✓ [Thm 4.3]
R-HIPN	✓ [Thm 5.1]	✗ [Thm 5.6]	✗ [[9], [1]]	✗ [Red.frm [9], [1]]
T-HIPN	Positivity-Hard [Thm 6.5]	✗ [Thm 5.6]	✗ [[9], [1]]	✗ [Red.frm [9], [1]]
R-HIRPN	✓ [Thm 5.1, Thm 4.2]	✗ [Thm 5.6]	✗ [[9], [1]]	✗ [Red.frm [9], [1]]

Status of decision problems and our contributions. Table 1 summarizes the decidability status of the four decision problems for some classes of Petri net extensions. A ✓ denotes decidability of the corresponding problem, while ✗ denotes undecidability of the problem. The shaded cells present results (and corresponding citations) already known prior to the current work, while the unshaded cells show results (and corresponding theorems) arising from this paper. Note that the table doesn't list all extensions of Petri nets that were defined above. This has been done deliberately and carefully to improve readability. Specifically, for every Petri net extension that is not represented in the table, e.g., R-HITPN, the status of all four decision problems are inferable from others shown in the table. These are explicitly listed out in a longer version of the paper [1], where we also depict the relative expressiveness of these classes. Thus, our work comprehensively addresses the four decision problems for all classes of Petri net extensions considered above.

Interestingly, several of the results use distinct constructions and proof techniques. We point out the salient features of our main results below.

- We *include reset arcs in the hierarchy* of inhibitor arcs in HIPN in Section 4. In Theorem 4.2, we show that we can model reset arcs by inhibitor arcs, while preserving hierarchy. This immediately gives decidability of all problems except DLFREE. Since the reduction may introduce deadlocks, we need a different proof for DLFREE, which we present through in Theorem 4.3. This also proves decidability of DLFREE for HIPN, which to the best of our knowledge, was not known before.
- We *add reset arcs outside the hierarchy* of inhibitor arcs in Section 5. Somewhat counter-intuitively, this class, R-HIPN, does not contain HIRPN and is incomparable to it. This is because all inhibitor arcs in R-HIPN are required to respect the hierarchy, whereas in HIRPN some inhibitor arcs can be replaced by resets, thereby violating the requirement of hierarchy among inhibitor arcs. Using a new and surprisingly simple construction of an extended finite reachability tree (FRT) which keeps track of the hierarchical inhibitor information and modifies the subsumption condition, we show in Theorem 5.1 that termination is decidable for R-HIPN. This result has many consequences. In particular, it implies an arguably simple proof for the special case of a single inhibitor arc, which was solved in [10] (using a different method of extending FRTs). In Theorem 5.6, we use a reduction from two counter machines to show that coverability is undecidable with as few as 2 reset arcs and an inhibitor arc in the absence of hierarchy.
- Finally, we *consider transfer arcs inside and outside the hierarchy* in Section 6. In Theorem 6.1, we show that unlike for reset arcs, including transfer arcs in the hierarchy



■ **Figure 1** Transformation from $N \in \text{HIRPN}_k$ (left) to $N' \in \text{HIRPN}_{k-1}$ (right).

of inhibitor arcs does not necessarily preserve decidability. For both HITPN and T-HIPN, while coverability, reachability and deadlock-freeness are undecidable, we are unable to show such a result for termination. Instead, in Theorem 6.5, we show that we can reduce a long-standing open problem on linear recurrences to this problem.

4 Adding Reset Arcs with Hierarchy to HIPN

In this section, we extend hierarchical inhibitor nets [20] with reset arcs respecting hierarchy. Subsection 4.1 presents a reduction from HIRPN to HIPN that settles the decidability of termination, coverability and reachability for HIRPN. As this reduction does not work for deadlock-freeness (since it introduces new deadlocked markings), we present a separate reduction from deadlock-freeness to reachability for HIRPN in subsection 4.2.

4.1 Reduction from HIRPN to HIPN

Let HIRPN_k be the sub-class of Petri nets in HIRPN with at most k transitions having one or more reset pre-arcs. We first show that termination, reachability and coverability for HIRPN_k can be reduced to the corresponding problems for HIRPN_{k-1} , for all $k > 0$. This effectively reduces these problems for HIRPN to the corresponding problems for HIRPN_0 (or HIPN), which are known to be decidable [20, 5]. In the following, we use $\text{Markings}(N)$ to denote the set of all markings of a net N .

► **Lemma 4.1.** *For every net N in HIRPN_k , there is a net N' in HIRPN_{k-1} and a mapping $f : \text{Markings}(N) \rightarrow \text{Markings}(N')$ that satisfy the following:*

- *For every $M_1, M_2 \in \text{Markings}(N)$ such that M_2 is reachable from M_1 in N , the marking $f(M_2)$ is reachable from $f(M_1)$ in N' .*
- *For every $M'_1, M'_2 \in \text{Markings}(N')$ such that $M'_1 = f(M_1)$, $M'_2 = f(M_2)$ and M'_2 is reachable from M'_1 in N' , the marking M_2 is reachable from M_1 in N .*

Proof Sketch. To see how N' is constructed, consider an arbitrary transition, say t , in N with one or more reset pre-arcs. We replace t by a gadget in N' with no reset arcs, as shown in Figure 1. The gadget has two new places labeled p^* and p_t^* , with every transition in “Rest of Net” having a simple pre-arc from and a post-arc to p^* , as shown by the dotted arrows in Figure 1. The gadget also has a new transition t^S with simple pre-arcs from p^* and from every place p^S that has a simple arc to t in N . It also has a new transition labeled t^R for every reset arc from a place p^R to t in N . Thus, if there are n reset pre-arcs of t in N , the gadget has n transitions t_1^R, \dots, t_n^R . As shown in Figure 1, each such t_i^R has simple pre-arcs

from p_t^R and p_t^* and a post-arc to p_t^* . Finally, the gadget has a new transition labeled t^I with a simple pre-arc from p_t^* and inhibitor pre-arcs from all places p^I (resp. p^R) that have inhibitor (resp. reset) arcs to t in N .

The ordering \sqsubseteq' of places in N' is obtained by extending the ordering \sqsubseteq of N as follows: for each place p in N , we have $p \sqsubseteq' p_t^* \sqsubseteq' p^*$. Clearly, $N' \in \text{HIRPN}_{k-1}$, since it has one less transition (i.e. t) with reset pre-arcs compared to N . It is easy to check that if the reset and inhibitor arcs in N respect \sqsubseteq , then the reset and inhibitor arcs in N' respect \sqsubseteq' as well.

The mapping function $f : \text{Markings}(N) \rightarrow \text{Markings}(N')$ is defined as follows: for every place p in N' , $f(M)(p) = M(p)$ if p is in N ; otherwise, $f(M)(p^*) = 1$ and $f(M)(p_t^*) = 0$. The initial marking of N' is given by $f(M_0)$, where M_0 is the initial marking of N . Given a run in N , it is now easy to see that every occurrence of t in the run can be replaced by the sequence $t^S(t^R)^*t^I$ (the t^R transitions fire until the corresponding place p^R is emptied) and vice-versa. Further details of the construction are given in [1], where it is also shown that N can reach M_2 from M_1 iff N' can reach $f(M_2)$ from $f(M_1)$. ◀

In fact, the above construction can be easily adapted for HIRcTPN as well. Specifically, if we have a transfer arc from place p_x to place p_y through t , we add a new transition $t_{x,y}^T$ with simple pre-arcs from p_t^* and p_x , and with simple post-arcs to p_t^* and p_y to the gadget shown in Figure 1. Furthermore, we add an inhibitor arc from p_x to t^I , like the arc from p^R to t^I in Figure 1. Note that the *constrained* property of transfer arcs is required here, since if we had an inhibitor arc from p_y to t in original net (hence, p_y to t^I in construction), then in the constructed net, t^I cannot be fired, since we would have added tokens in p_y through $t_{x,y}^T$. This allows us to obtain a net in HIRcTPN with at least one less transition with reset pre-arcs or transfer arcs, such that the reachability guarantees in Lemma 4.1 hold.

Finally, by repeatedly applying Lemma 4.1, we have,

► **Theorem 4.2.** *Termination, reachability and coverability for HIRPN and HIRcTPN are decidable.*

4.2 Reducing Deadlock-freeness to Reachability in HIRPN

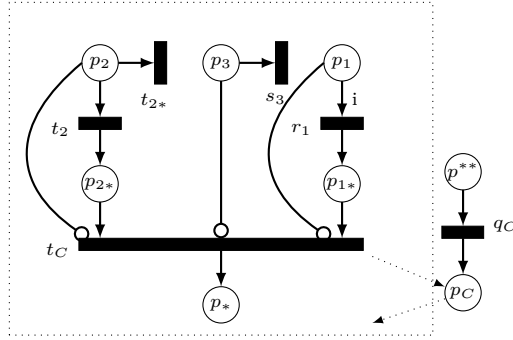
The overall idea behind our reduction is to add transitions that check whether the net is deadlocked, and to put a token in a special place, say p^* , if this is indeed the case. Note that for a net to be deadlocked, the firing of each of its transitions must be disabled. Intuitively, if M denotes a marking of a net and if T denotes the set of transitions, then $\text{Deadlock}(M) = \bigwedge_{t_i \in T} \text{NotFire}_i(M)$, where $\text{Deadlock}(M)$ is a predicate indicating if the net is deadlocked in M , and $\text{NotFire}_i(M)$ is a formula representing the enabledness of transition t_i in M . For a transition t to be disabled, atleast one of its pre-places p must fail the condition on that place for t to fire. There are three cases to consider here.

- $F(p, t) \in \mathbb{N}$: For t to be disabled, we must have $M(p) < F(p, t)$
- $F(p, t) = I$: For t to be disabled, we must have $M(p) > 0$.
- $F(p, t) = R$: Place p cannot disable t

Suppose we define $\text{Exact}_j(p) \equiv (M(p) = j)$ and $\text{AtLeast}(p) \equiv (M(p) > 0)$. Clearly, $\text{NotFire}_i(M) = \bigvee_{(p,t) \in F} \text{Check}(p)$, where $\text{Check}(p) = \text{AtLeast}(p)$ if $F(p, t) = I$, and $\text{Check}(p) = \bigvee_{j < k} \text{Exact}_j(p)$ if $F(p, t) = k \in \mathbb{N}$. The formula for $\text{Deadlock}(M)$ (in CNF above) is now converted into DNF by distributing conjunctions over disjunctions. Given a HIRPN, we now transform the net, preserving hierarchy, so as to reduce checking $\text{Deadlock}(M)$ in DNF in the original net to a reachability problem in the transformed HIRPN.

Every conjunctive clause in the DNF of $\text{Deadlock}(M)$ is a conjunction of literals of the form $\text{AtLeast}(p)$ and $\text{Exact}_j(p)$. Let S_C be the set of all literals in a conjunctive clause C ,

and let P be the set of all places in the net. Define $B_i^C = \{p \in P \mid \text{Exact}_i(p) \in S_C\}$ and $A^C = \{p \in P \mid \text{AtLeast}(p) \in S_C\} \setminus \bigcup_{i \geq 1} B_i^C$. We only need to consider conjunctive clauses where the sets B_i^C are pairwise disjoint (other clauses can never be true). Similarly, we only need to consider conjunctive clauses where B_0^C and A^C are disjoint. We add a transition for each conjunctive clause that satisfies the above two properties. By definition, A^C and B_i^C are disjoint for all $i \geq 1$. Thus, the sets A^C and B_i^C ($i \geq 0$) are pairwise disjoint for every conjunctive clause we consider.



Given the original HIRPN net, for each conjunctive clause considered, we perform the construction as shown in the above figure. For every place $p_a \in A^C$, we add a construction as for p_2 in above diagram. For every place $p_i \in B_i^C$, we add a construction as for p_1 in above diagram. For all places $p \notin A^C \cup \bigcup_i B_i^C$, we add a construction as for p_3 in above diagram. We call the transition t_C in the above diagram as the "Check Transition", and refer to the set of transitions $r_i, s_i, t_i, t_{i*}, t_C$ (excluding q_C) as *transitions for clause C*. Note that for any $p_i \in P$, exactly one of r_i, s_i, t_i exist since the sets A^C and the sets B_i^C are all pairwise disjoint.

Our construction also adds two new places, p_C and p^{**} , and one new transition q_C such that

- there is a pre-arc and a post-arc of weight 1 from p^{**} to every transition in the original net. Thus, transitions in original net can fire only if p^{**} has a token.
- there is a pre-arc of weight 1 from p_c to each transition for clause C (within dotted box).
- there is a post-arc of weight 1 to p_c from every transition for clause C (within dotted box), except from t_C to p_C .

Note that hierarchy is preserved in the transformed net, since the only new transitions which have inhibitor/reset arcs are the check transitions, which have inhibitor arcs from all places in the original net. Let N be the original net in HIRPN with P being its set of places, and let N' be the transformed net, also in HIRPN, obtained above. Define a mapping $f : \text{Markings}(N) \rightarrow \text{Markings}(N')$ as follows: $f(M)(p) = M(p)$ if $p \in P$; $f(M)(p^{**}) = 1$ and $f(M)(p) = 0$ in all other cases. If M_0 is the initial marking in N , define $M'_0 = f(M_0)$ to be the initial marking in N' .

► **Claim 4.1.** The marking M'_* of N' , defined as $M'_*(p) = 1$ if $p = p^*$ and $M'_*(p) = 0$ otherwise, is reachable from M'_0 in N' iff there exists a deadlocked marking reachable from M_0 in N .

From, the above reduction (see [1] for proof details) and from the decidability of reachability for HIRPN we then have the following.

► **Theorem 4.3.** *Deadlock-freeness for HIRPN is decidable.*

5 Adding Reset Arcs without Hierarchy

The previous section dealt with extension of Petri nets where reset arcs were added within the hierarchy of the inhibitor arcs. This section discusses the decidability results when we add reset arcs outside the hierarchy of inhibitor arcs.

5.1 Termination in R-HIPN

Our main idea here is to use a modified finite reachability tree (FRT) construction to provide an algorithm for termination in R-HIPN. The usual FRT construction [11] for Petri nets does not extend to Petri nets with even a single (and hence also hierarchical) inhibitor arc.

► **Theorem 5.1.** *Termination is decidable for R-HIPN.*

Consider a R-HIPN net $(P, T, F, \sqsubseteq, M_0)$. We start by introducing a few definitions. For any place $p \in P$, we define the *index of the place p* ($Index(p)$) as the number of places $q \in P$ such that $q \sqsubseteq p$. The definition of Index over places induces an Index among transitions too: For any transition $t \in T$, its *index* is defined as $Index(t) = \max_{F(p,t)=I} Index(p)$. By convention, if there is no such place, then $Index(t) = 0$. Given markings M_1 and M_2 and $i \in \mathbb{N}$, we say that M_1 and M_2 are *i -Compatible* (denoted $Compat_i(M_1, M_2)$) if $\forall p \in P, (Index(p) \leq i \implies M_1(p) = M_2(p))$.

► **Definition 5.2.** Consider a run $M_2 \xrightarrow{\rho} M_1$. Let $t^* = \operatorname{argmax}_{t \in \rho} Index(t)$. We define $Subsume(M_2, M_1, \rho) = M_2 \leq M_1 \wedge \left(Compat_{Index(t^*)}(M_1, M_2) \right)$

To understand this definition note that if ρ can be fired at M_2 and reaches M_1 and if $Subsume(M_2, M_1, \rho)$ is true, then ρ can be fired at M_1 again. In classical Petri nets without inhibitor arcs, $Subsume(M_2, M_1, \rho) = M_2 \leq M_1$, and hence this is the classical monotonicity condition. However, this condition may differ in the presence of even a single inhibitor arc.

Given a net $N = (P, T, F, \sqsubseteq, M_0)$ in R-HIPN, we define the *Extended Reachability Tree* $ERT(N)$ as a directed unordered tree where the nodes are labelled by markings $M : P \rightarrow \mathbb{N}$, rooted at $n_0 : M_0$ (initial marking). If $M_1 \xrightarrow{t} M_2$ for some markings M_1 and M_2 and transition $t \in T$, then a node marked by $n' : M_2$ is a child of the node $n : M_1$. Consider any node labelled M_1 . If along the path from root $n_0 : M_0$ to $n : M_1$, there is a marking $n' : M_2$ ($n \neq n'$), such that the path from $n' : M_2$ to $n : M_1$ corresponds to a run ρ and $Subsume(M_2, M_1, \rho)$ is true, then M_1 is made a leaf node. We call this a *subsumed* leaf node. Note that leaf nodes in this tree are of two types: either leaf nodes caused by subsumption as above or leaf nodes due to deadlock, where no transition is fireable.

► **Lemma 5.3.** *For any net $N = (P, T, F, \sqsubseteq, M_0)$ in R-HIPN, $ERT(N)$ is finite.*

Proof. Assume the contrary. By Konig's Lemma, there is an infinite path. Let the infinite path correspond to a run $\rho = M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \dots \xrightarrow{t_i} M_i \dots$

Let $t \in T$ be the transition which has maximum index among the transitions which are fired infinitely often in run ρ . Thus all transitions having higher index than $Index(t)$ fire only finitely many times. Let b be chosen such that $\forall i \geq b, Index(t_i) \leq Index(t)$ (i.e. b is chosen after the last position where any transition with index higher than $Index(t)$ fires). This exists by the definition of t . Since t is fired infinitely often, the sequence $\{M_i \mid i > b \wedge t_{i+1} = t\}$ is an infinite sequence. As \leq over markings is a well-quasi ordering, there exist two markings M_i and M_j , such that both belong to the above sequence (i.e. $t_{i+1} = t_{j+1} = t$), $M_i \leq M_j$ and $i < j$. Since $t_{i+1} = t_{j+1} = t$, and t fires at M_i and M_j , we have $\forall p \in P, Index(p) \leq$

$Index(t) \implies M_i(p) = M_j(p) = 0$. Thus, $Compat_{Index(t)}(M_i, M_j)$ is true. Note that t is the maximum index transition fired in the run from M_i to M_j , since no higher index transition fires after position b and $j > i > b$. Hence, $Subsume(M_i, M_j, \rho')$ is true, where ρ' is the run from M_i to M_j . But then, the path would end at M_j , giving a contradiction. ◀

Thus, we have shown that the ERT is always finite. Next, we will show a crucial property of $Compat_i$, which will allow us to check for a non-terminating run.

► **Lemma 5.4.** *Consider markings M_1 and M_2 such that $M_1 \leq M_2$. Let $i \in \mathbb{N}$ be such that we have $Compat_i(M_1, M_2)$. Then for any run ρ over the set of transitions $T_i = \{t \mid t \in T \wedge Index(t) \leq i\}$, if $M_1 \xrightarrow{\rho} M'_1$, then $M_2 \xrightarrow{\rho} M'_2$, where $M'_1 \leq M'_2$ and $Compat_i(M'_1, M'_2)$.*

Proof. We prove this by induction. We first prove that t is fireable at M_2 . If $F(p, t) \in \mathbb{N}$, then $M_2(p) \geq M_1(p) \geq F(p, t)$. If $F(p, t) = I$, i.e., it is an inhibitor arc, then $Index(p) \leq Index(t) \leq i$. But now, since $Compat_i(M_1, M_2)$ holds and t is fireable at M_1 , we obtain $M_2(p) = M_1(p) = 0$. Finally, if $F(p, t) = R$, i.e., it is a reset arc, then t can fire regardless of the value of $M_2(p)$. Hence, t is fireable at M_2 .

Now let $M_2 \xrightarrow{t} M'_2$. Then, for all $p \in P$, $M'_2(p) = M_2(p) - F(p, t) + F(t, p)$ and $M'_1(p) = M_1(p) - F(p, t) + F(t, p)$. Since $F(t, p)$ is constant and $F(p, t)$ can depend only on number of tokens in place p (so, if $M_1(p)$ and $M_2(p)$ were equal before firing, they remain equal now), we obtain that $Compat_i(M'_1, M'_2)$ and $M'_1 \leq M'_2$. ◀

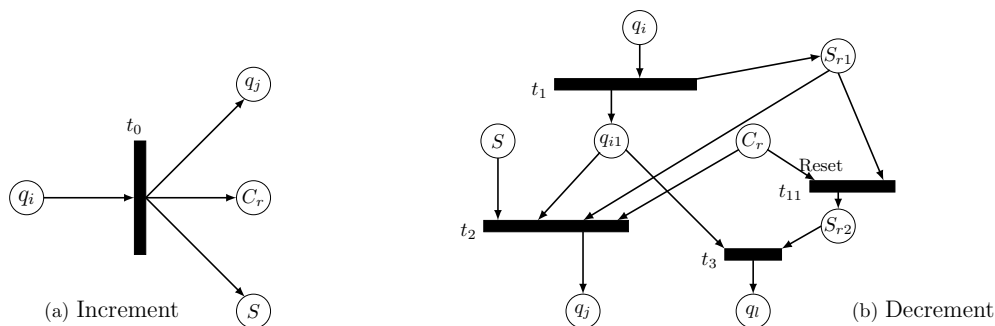
► **Lemma 5.5.** *A net N in R-HIPN has a non-terminating run iff $ERT(N)$ has a subsumed leaf node.*

Proof. In the forward direction, consider a non-terminating run. This run has a finite prefix in $ERT(N)$. This prefix ends in a leaf that is not a deadlock (as some transition is fireable). Thus it is a subsumed leaf node. In the reverse direction, we now show that if $ERT(N)$ has a subsumed leaf node, then N has a non-terminating run. To see this, consider any subsumed leaf node labeled by marking M_2 . Let M_1 be a marking along the path M_0 to M_2 , and ρ be the run from M_1 to M_2 , such that $Subsume(M_2, M_1, \rho)$ is true. Hence, we have $M_1 \xrightarrow{\rho} M_2$. Take $t^* = \operatorname{argmax}_{t \in \rho} Index(t)$ and $i = Index(t^*)$. Since $Subsume(M_1, M_2, \rho)$ is true, we have $M_1 \leq M_2$ and $Compat_i(M_1, M_2)$ is true. We also have that ρ is a run over $T_i = \{t \mid t \in T \wedge Index(t) \leq i\}$ (by definition of i). Thus, by Lemma 5.4, we have $M_2 \xrightarrow{\rho} M_3$, where $M_2 \leq M_3$ and $Compat_i(M_2, M_3)$ is true. Hence, ρ can be fired again at M_3 and so on, resulting in a non-terminating run. ◀

Finally, we observe that checking $Subsume(M_2, M_1, \rho)$ is also easily doable. Thus, for any net in R-HIPN, one can construct its extended reachability tree and decide the termination problem using the ERT. This completes the proof of the theorem. We observe here that this construction cannot be immediately lifted to checking boundedness due to the presence of reset arcs. However, we can lift this to check for termination in HIRPN and R-HIRPN as well as to check boundedness in HIPN.

5.2 Coverability in R-HIPN

While termination turned out to be decidable, reachability is undecidable for R-HIPN in general (since it subsumes reset Petri nets). Indeed, it is shown in [9] that reachability is undecidable for Petri nets with 2 reset arcs. Using a similar strategy, in [1], we tighten the undecidability result to show that reachability in Petri nets with one inhibitor arc and one



■ **Figure 2** Modeling a Minsky Machine (unlabelled edges have weight 1).

reset arc is undecidable. Further, we can modify the construction presented to show that deadlock-freeness in Petri nets with one reset arc and one inhibitor arc is also undecidable.

Next we turn our attention to the coverability problem.

► **Theorem 5.6.** *Coverability is undecidable for Petri nets with two reset/transfer arcs and an inhibitor arc.*

The rest of this section proves the above theorem. To do this, we construct a Petri net with two reset arcs and one inhibitor arc that simulates a two-counter Minsky Machine. A Minsky Machine M has a finite set of instructions q_i for $0 \leq i \leq n$, where q_0 is the initial instruction and q_n is the final instruction i.e. there are no transition rules from q_n . There are two counters C_1 and C_2 and we have two types of instructions. For each counter $r \in \{1, 2\}$,

1. $\text{INC}(r, j)$: Increase C_r by 1 and go to q_j .
2. $\text{JZDEC}(r, j, l)$: If C_r is zero, go to q_l , else decrease C_r by 1 and go to q_j .

We construct a Petri net P such that the runs of P encode the runs of the Minsky Machine M . We use places q_i ($0 \leq i \leq n$) to encode each instruction. The place q_i gets a token when we simulate instruction i in the Minsky Machine. We use two places C_1 and C_2 to store the number of tokens corresponding to counter values in C_1 and C_2 in the counter machine. We use a special place S which stores the sum of C_1 and C_2 . Figure 2(a) shows the construction for the increment instruction “Increase C_r by 1 and go to q_j ”. When q_i gets a token, the transition is fired, C_r and S are incremented by 1 and q_j gets the token to proceed.

Next, we show how to simulate the decrement (along with zero check) instruction “if $C_r = 0$, then go to q_l , else decrease C_r by 1 and go to q_j ” by introducing non-determinism in the Petri net. The gadget for this is shown in Figure 2(b). When we reach a decrement with zero check instruction, we guess whether C_r is zero, and if so, fire t_{11} and then t_3 . Else, we decrement C_r by 1 and fire t_2 . We have two cases. **Case 1:** If C_r is actually zero, it runs correctly as t_2 would not fire. The transition t_3 fires and C_r remains zero. In addition, q_l gets the token. **Case 2 :** If C_r has non-zero tokens, both transitions can fire. But runs in which t_3 fires are “wrong” runs. We call such transitions as *incorrect*. The crucial point is that in runs with incorrect transitions, S is not decremented whereas C_r is decremented. Hence $M(S) \neq M(C_1) + M(C_2)$ in markings reached by runs with incorrect transitions.

Note that in any run of the Petri net P , q_i and only q_i gets a token when the instruction numbered i is being simulated. The following lemma proves the correctness of the reduction.

► **Lemma 5.7.** *In every run of P reaching marking M , $M(S) \geq M(C_1) + M(C_2)$. Furthermore, $M(S) = M(C_1) + M(C_2)$ iff there are no incorrect transitions.*

If the Minsky Machine reaches instruction q_n , the Petri net P gets a token in the place q_n . But, if the Minsky Machine doesn't reach q_n , it is possible that the place q_n in Petri net P gets a token because of incorrect transitions. By the above lemma, to check if there had been any incorrect transitions along the run, we just check at the end (at q_n) if $M(S) = M(C_1) + M(C_2)$. This can be done using an inhibitor arc. Thus q_{n+1} gets tokens iff the Minsky Machine reaches the instruction q_n . Hence reaching instruction q_n in Minsky Machine is equivalent to asking if we can cover the marking in which all places except q_{n+1} have 0 tokens and q_{n+1} has 1 token. Since checking reachability in Minsky machines is undecidable, this shows that checking coverability in Petri nets with 2 reset and an inhibitor arc is undecidable.

Note that the above proof also shows undecidability of coverability in Petri nets with 2 transfer arcs and an inhibitor arc. Additional details, the inhibitor arc construction and the extension to transfer arcs can be found in the long version of this paper [1]. Finally, the problem of coverability in Petri nets with 1 inhibitor arc and 1 reset arc is open.

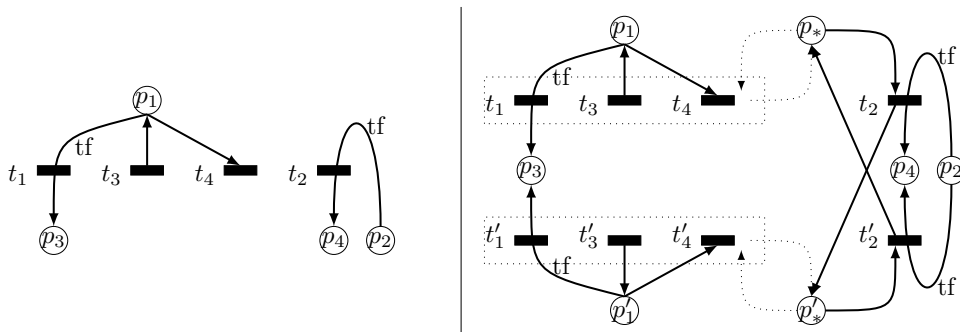
► Problem 1. Is coverability in Petri nets with 1 reset arc and 1 inhibitor arc decidable ?

6 Adding Transfer Arcs within and without Hierarchy

We show a reduction from Petri nets with 2 (non-hierarchical) transfer arcs to HTPN preserving reachability and deadlock-freeness. Since reachability and deadlock-freeness in Petri nets with 2 transfer arcs are undecidable [9], they are undecidable in HTPN too.

► **Theorem 6.1.** *Reachability and deadlock-freeness are undecidable in HTPN.*

Proof. Let N be a Petri net with 2 transfer arcs as shown in the Figure below. Here, one transfer arc is from p_1 to p_3 via t_1 , while the other is from p_2 to p_4 via t_2 . The transitions t_3 and t_4 are representative of other transitions to and from p_1 . W.l.o.g. we assume that there is no arc from p_1 to t_2 . If this is not the case, we can add a place and transition in between to create an equivalent net without adding any deadlocked reachable marking (see [1] for details). The construction of the corresponding HTPN is shown in the diagram below.



Six transfer arcs have not been shown in the construction above for clarity. These are:

- From p_1 to p_3 through t'_1 .
- From p_1 to p'_1 through t_2 .
- From p_1 to p'_1 through t'_2 .
- From p'_1 to p_3 through t_1 .
- From p'_1 to p_1 through t_2 .
- From p'_1 to p_1 through t'_2 .

These arcs ensure that hierarchy is respected by the transfer arcs, where $p_1 < p'_1 < p_2$ in the hierarchy. The dotted arc from p_* to the upper dotted box represents a pre-arc from p_* to every transition in the box. Similarly, we have an arc from every transition in the upper dotted box to p_* . Dotted arcs between the lower dotted box and p'_* are interpreted similarly. The intuitive idea behind the construction is to represent the place p_1 in the original net by two places p_1 and p'_1 in the modified net. At every marking, p_1 of the original net is represented by one of the two places p_1 or p'_1 in the modified net. Places p'_* and p_* are used to keep track of which place represents p_1 in the current marking. Everytime the transition t_2 fires, the representative place swaps. Let the original net be (P, T, F) and the constructed net be (P', T', F') . The initial marking M'_0 is given by $M'_0(p_*) = 1$, $M'_0(p'_*) = M'_0(p'_1) = 0$, and $M'_0(p) = M_0(p)$ for all other $p \in P$. Now, given marking M of original net, we define the set $S^{ext} = \{A_M, B_M\}$, where,

$$A_M(p) = \begin{cases} M(p) & p \notin \{p_1, p'_1, p'_*, p_*\} \\ 1 & p = p_* \\ 0 & p \in \{p'_*, p'_1\} \\ M(p_1) & p = p_1 \end{cases}$$

$$B_M(p) = \begin{cases} M(p) & p \notin \{p_1, p'_1, p'_*, p_*\} \\ 1 & p = p'_* \\ 0 & p \in \{p_*, p_1\} \\ M(p_1) & p = p'_1 \end{cases}$$

► **Claim 6.1.** Marking A_M or B_M is reachable from M'_0 in the constructed net iff marking M is reachable from M_0 in the original net.

The proof of the claim is given in [1]. The proof of Theorem 6.1 follows from this claim ◀

► **Corollary 6.2.** *Coverability is undecidable in HITPN.*

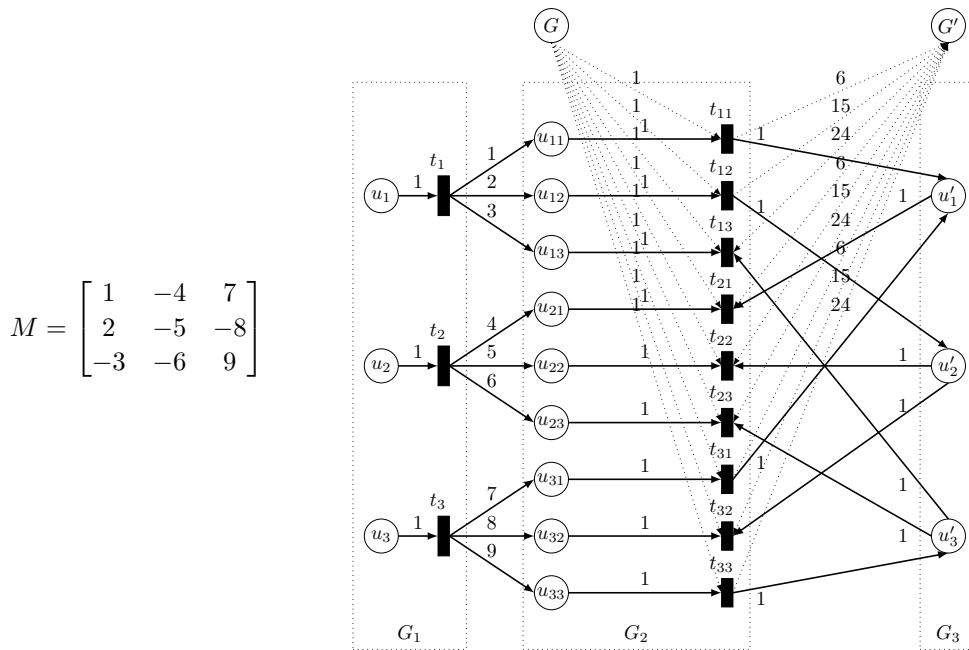
Proof. From Theorem 5.6, it follows that coverability is undecidable in Petri nets with two transfer arcs and one inhibitor arc. Now, we can perform a construction similar to the one above to reduce the coverability of this net to the coverability problem of a net in HITPN. ◀

6.1 Hardness of Termination in HITPN

Termination in HIRcTPN was shown to be decidable in Section 4.1. Termination in HTPN is also decidable, as it is known that termination in transfer Petri nets is decidable. However, termination in HITPN, which subsumes the above two problems, is as hard as the positivity problem – a long standing open problem about linear recurrent sequences ([19],[18]). We prove this by reducing the positivity problem to termination in HITPN.

► **Definition 6.3** (Positivity Problem). Given a matrix $M \in \mathbb{Z}^{n \times n}$ and a vector $v_0 \in \mathbb{Z}^n$, is $M^k v_0 \geq 0$ for all $k \in \mathbb{N}$?

Given matrix $M \in \mathbb{Z}^{n \times n}$ and a vector $v_0 \in \mathbb{Z}^n$, we construct a net $N \in \text{HITPN}$ such that N does not terminate the coverability problem of a net in HITPN iff $M^k v_0 \geq 0$ for all $k \in \mathbb{N}$. Consider the following while loop program $v = v_0$; **while** ($v \geq 0$) $v = Mv$. Clearly, this program is non-terminating iff $M^k v_0 \geq 0$ for all k . We construct a net N which simulates this linear program. The net N contains two phases, a forward phase that has the effect of multiplying v by M , and a backward phase that mimics assigning the new vector $M \cdot v$

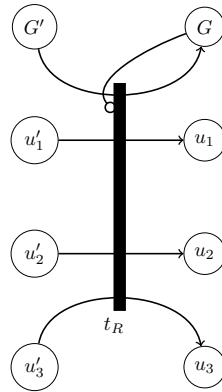


■ **Figure 3** Forward phase (right) simulating matrix M (left).

computed in the forward phase back to v . We also check for non-negativity in the backward phase, and design the net N to terminate if any component becomes negative.

Forward Phase. The construction of the forward phase Petri net for a general matrix is explained below. An example of the construction is shown in Figure 3. We have n places, u_1, u_2, \dots, u_n , corresponding to the n components of vector v . Each place u_i is connected to a transition t_i with a pre-arc of weight 1. Each t_i also has a post-arc to a new place u_{ij} for $1 \leq i, j \leq n$ with a weight $|M_{ji}|$, i.e. the absolute value of the $(j, i)^{th}$ entry of matrix M , corresponding to v_i 's contribution to the new value of v_j . Finally, we have places u'_1, u'_2, \dots, u'_n , corresponding to the n components of the new value of vector v . Each place u'_j is connected to place u_{ij} by a transition t_{ij} , with both the arcs being weighted 1. If $M_{ji} \geq 0$, we make use of the fact that u_{ij} has a pre-arc to t_{ij} and t_{ij} has a post-arc to u'_j . This has the effect of adding the value of u_{ij} to u'_j . On the other hand, if $M_{ji} < 0$, then we make use of the fact that both u_{ij} and u'_j have pre-arcs to t_{ij} to subtract u_{ij} from u'_j . The above run simulates the forward phase, in effect multiplying the vector v , represented by the column of u_i in Figure 3, by M , and storing the new components in the column on u'_i . To simulate the while loop program, we need to copy back each u'_i to u_i , while performing the check that each u'_i is non-negative.

Backward phase. The copy back in the backward phase (Figure 4) is effected by a transfer arc from u'_i to u_i via transition t_R . To ensure that the backward phase starts only after the forward phase completes (else, partially computed values would be copied back), we introduce a new place G . The place G stores as many tokens as the total number of times each transition t_{ij} fires, and has a pre-arc weighted 1 to each transition t_{ij} . The emptiness of G ensures that each t_{ij} has completed its firings in the current loop iteration. An inhibitor arc from G to t_R ensures that the forward phase completes before t_R fires. We introduce place G' which computes the initial value of G for the next loop iteration. G' has an arc



■ **Figure 4** Backward phase: Arc from G to t_R is an inhibitor arc, rest are transfer arcs.

connected to t_{ij} with weight $\sum_{k=1}^n |M_{kj}|$. If u'_j has a pre-arc to t_{ij} , then G' has a pre-arc to t_{ij} , while if t_{ij} has a post-arc to u'_j , then it also has a post-arc to G' . Finally, there is a transfer arc from G' to G via t_R . Once the forward phase finishes, the place G becomes empty. Hence, the only transition that can fire is t_R , which completes the backward phase in one firing. Combining the forward and backward phases, we obtain a net N which simulates the while loop program. The initial marking assigns $(v_0)_i$, i.e. the i -th component of vector v_0 , to place u_i , and $\sum_{1 \leq i \leq n} (\sum_{1 \leq j \leq n} |M_{ji}|)(v_0)_i$ tokens to G , while all other places are assigned 0 tokens. The following lemma (see [1] for details) relates termination of N with the Positivity problem.

► **Lemma 6.4.** *There exists a non-terminating run in N iff $M^k v_0 \geq 0$ for all $k \in \mathbb{N}$.*

Note that the net N constructed above has only one transition with inhibitor and transfer arcs; hence N is in T-HIPN as well as in HIPN. Thus, we have,

► **Theorem 6.5.** *Termination in HIPN as well as T-HIPN is as hard as positivity problem.*

7 Conclusion

In this paper, we investigated the effect of hierarchy on Petri nets extended with not only inhibitor arcs (as classically considered), but also reset and transfer arcs. For four of the standard decision problems, we settled the decidability for almost all these extensions using different reductions and proof techniques. As future work, we are interested in questions of boundedness and place-boundedness in these extended classes. We would also like to explore further links to problems on linear recurrences. We leave open one technical question of coverability for Petri nets with 1 reset and 1 inhibitor arc (without hierarchy).

Acknowledgments. We thank Alain Finkel and Mohamed Faouzi Atig for interesting and insightful discussions and pointers to earlier results and reductions.

References

- 1 S. Akshay, S. Chakraborty, A. Das, V. Jagannath, and S. Sandeep. On Petri Nets with Hierarchical Special Arcs. *ArXiv e-prints*, July 2017. arXiv:1707.01157.
- 2 Mohamed Faouzi Atig and Pierre Ganty. Approximating Petri net reachability along context-free traces. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2011, December 12-14, 2011, Mumbai, India*, pages 152–163, 2011.
- 3 Amir M. Ben-Amram, Samir Genaim, and Abu Naser Masud. On the termination of integer loops. *ACM Trans. Program. Lang. Syst.*, 34(4):16:1–16:24, December 2012.
- 4 Rémi Bonnet. The reachability problem for vector addition system with one zero-test. In *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings*, pages 145–157, 2011.
- 5 Rémi Bonnet. Theory of well-structured transition systems and extended vector-addition systems. *These de doctorat, ENS Cachan, France*, 2013.
- 6 Rémi Bonnet, Alain Finkel, Jérôme Leroux, and Marc Zeitoun. Place-boundedness for vector addition systems with one zero-test. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India*, pages 192–203, 2010.
- 7 Rémi Bonnet, Alain Finkel, Jérôme Leroux, and Marc Zeitoun. Model checking vector addition systems with one zero-test. *Logical Methods in Computer Science*, 8(2), 2012.
- 8 Allan Cheng, Javier Esparza, and Jens Palsberg. Complexity results for 1-safe nets. *Theor. Comput. Sci.*, 147(1&2):117–136, 1995. doi:10.1016/0304-3975(94)00231-7.
- 9 Catherine Dufourd, Alain Finkel, and Philippe Schnoebelen. Reset nets between decidability and undecidability. In *Automata, Languages and Programming, 25th International Colloquium, ICALP'98, Aalborg, Denmark, July 13-17, 1998, Proceedings*, pages 103–115, 1998. doi:10.1007/BFb0055044.
- 10 Alain Finkel and Arnaud Sangnier. Mixing coverability and reachability to analyze VASS with one zero-test. In *SOFSEM 2010: Theory and Practice of Computer Science, 36th Conference on Current Trends in Theory and Practice of Computer Science, Spindleruv Mlýn, Czech Republic, January 23-29, 2010. Proceedings*, pages 394–406, 2010.
- 11 Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, 256(1-2):63–92, 2001.
- 12 Michel Hack. The recursive equivalence of the reachability problem and the liveness problem for Petri nets and vector addition systems. In *15th Annual Symposium on Switching and Automata Theory, New Orleans, Louisiana, USA, October 14-16, 1974*, pages 156–164, 1974. doi:10.1109/SWAT.1974.28.
- 13 S. Rao Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 267–281, New York, NY, USA, 1982. ACM. doi:10.1145/800070.802201.
- 14 Jérôme Leroux. Vector addition systems reachability problem (A simpler solution). In *Turing-100 - The Alan Turing Centenary, Manchester, UK, June 22-25, 2012*, pages 214–228, 2012. URL: <http://www.easychair.org/publications/?page=1673703727>.
- 15 Ernst W. Mayr. An algorithm for the general Petri net reachability problem. *SIAM J. Comput.*, 13(3):441–460, 1984. doi:10.1137/0213029.
- 16 Marvin L Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., 1967.
- 17 Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
- 18 Joël Ouaknine and James Worrell. Positivity problems for low-order linear recurrence sequences. In *Proceedings of the Twenty-fifth Annual ACM-SIAM Symposium on Discrete*

- Algorithms*, SODA '14, pages 366–379, Philadelphia, PA, USA, 2014. Society for Industrial and Applied Mathematics. URL: <http://dl.acm.org/citation.cfm?id=2634074.2634101>.
- 19 Joël Ouaknine and James Worrell. On linear recurrence sequences and loop termination. *SIGLOG News*, 2(2):4–13, 2015. doi:10.1145/2766189.2766191.
 - 20 Klaus Reinhardt. Reachability in Petri nets with inhibitor arcs. *Electr. Notes Theor. Comput. Sci.*, 223:239–264, 2008. doi:10.1016/j.entcs.2008.12.042.