



HAL
open science

Coverability, Termination, and Finiteness in Recursive Petri Nets

Alain Finkel, Serge Haddad, Igor Khmelnitsky

► **To cite this version:**

Alain Finkel, Serge Haddad, Igor Khmelnitsky. Coverability, Termination, and Finiteness in Recursive Petri Nets. *Fundamenta Informaticae*, 2022, 183, pp.33 - 66. 10.3233/fi-2021-2081 . hal-04270919

HAL Id: hal-04270919

<https://hal.science/hal-04270919v1>

Submitted on 13 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Coverability, Termination, and Finiteness in Recursive Petri Nets

Alain Finkel^{1,3} *, Serge Haddad^{1,2**}, and Igor Khmelnitsky^{1,2}

¹ Université Paris-Saclay, CNRS, ENS Paris-Saclay, Laboratoire Méthodes Formelles, 91190, Gif-sur-Yvette, France.

{finkel,haddad,khmelnitsky}@lsv.fr

² Inria, France

³ Institut Universitaire de France

Abstract. In the early two-thousands, Recursive Petri nets have been introduced in order to model distributed planning of multi-agent systems for which counters and recursivity were necessary. Although Recursive Petri nets strictly extend Petri nets and context-free grammars, most of the usual problems (reachability, coverability, finiteness, boundedness and termination) were known to be solvable by using non-primitive recursive algorithms. For almost all other extended Petri nets models containing a stack, the complexity of coverability and termination are unknown or strictly larger than EXPSPACE. In contrast, we establish here that for Recursive Petri nets, the coverability, termination, boundedness and finiteness problems are EXPSPACE-complete as for Petri nets. From an expressiveness point of view, we show that coverability languages of Recursive Petri nets strictly include the union of coverability languages of Petri nets and context-free languages. Thus we get a more powerful model than Petri net for free.

Keywords: Recursive Petri nets, Expressiveness, Complexity, Coverability, Termination, Finiteness.

1 Introduction

Verification problems for Petri nets. Petri net is a useful formalism for the analysis of concurrent programs for several reasons. From a modeling point of view (1) due to the locality of the firing rule, one easily models concurrent activities and (2) the (a priori) unbounded marking of places allows to represent a dynamic number of activities. From a verification point of view, most of the usual properties are decidable. However, Petri nets suffer two main limitations: they cannot model recursive features and the computational cost of verification may be very high. More precisely, all the known algorithms solving reachability

* The work of this author was carried out in the framework of ReLaX, UMI2000 and also supported by ANR-17-CE40-0028 project BRAVAS.

** The work of this author was partly supported by ERC project EQualIS (FP7-308087)

are nonprimitive recursive (see for instance [1]) and it has been proved recently that the reachability problem is non elementary [2] but primitive recursive when the dimension is fixed [3]. Fortunately some interesting properties like coverability, termination, finiteness, and boundedness are EXPSPACE-complete [4] and thus still manageable by a tool. So an important research direction consists of extending Petri nets to support new modeling features while still preserving decidability of properties checking and if possible with a "reasonable" complexity.

Extended Petri nets. Such extensions may be partitioned between those whose states are still markings and the other ones. The simplest extension consists of adding inhibitor arcs which yields undecidability of most of the verification problems. However adding a single inhibitor arc preserves the decidability of the reachability, coverability, and boundedness problems [5,6,7]. When adding reset arcs, the coverability problem becomes Ackermann-complete [8] and boundedness undecidable [9].

In ν -Petri nets, the tokens are colored where colors are picked in an infinite domain: their coverability problem is double-Ackermann time complete [10]. In Petri nets with a stack, the reachability problem may be reduced to the coverability problem and both are at least not elementary [2,11] while their decidability status is still unknown [11]. In branching vector addition systems with states (BVASS) a state is a set of threads with associated markings. A thread either fires a transition as in Petri nets or forks, transferring a part of its marking to the new thread. For BVASS, the reachability problem is also TOWER-hard [12] and its decidability is still an open problem while the coverability and the boundedness problems are 2-EXPTIME-complete [13]. The analysis of subclasses of Petri nets with a stack is an active field of research [14,15,16,17]. However, for none of the above extensions, the coverability and termination problems belong to EXPSPACE.

Recursive Petri nets (RPN). This formalism has been introduced to model distributed planning of multi-agent systems for which counters and recursivity were necessary for specifying resources and delegation of subtasks [18]. Roughly speaking, a state of an RPN consists of a tree of *threads* where the local state of each thread is a marking. Any thread fires an *elementary*, *abstract* or *cut* transition. When the transition is elementary, the firing updates its marking as in Petri nets; when it is abstract, this only consumes the tokens specified by the input arcs of the transition and creates a child thread initialized with the *initial marking* of the transition. When a cut transition is fired, the thread and its subtree are pruned, producing in its parent the tokens specified by the output arcs of the abstract transition that created it. In RPN, reachability, boundedness and termination are decidable [19,20] by reducing these properties to reachability problems of Petri nets. So the corresponding algorithms are nonelementary. LTL model checking is undecidable for RPN but becomes decidable for the subclass of sequential RPN [21]. In [22], several modeling features are proposed while preserving the decidability of the verification problems.

Our contribution. We first study the expressive power of RPN from the point of view of coverability languages (reachability languages were studied in [19]). We

first introduce a quasi-order on states of RPN compatible with the firing rule and establish that it is not a well quasi-order. Moreover, we show that there cannot exist a transition-preserving compatible well quasi-order, preventing us to use the framework of Well Structured Transition Systems to prove that coverability is decidable. We show that the RPN languages are *quite close* to recursively enumerable languages since the closure under homomorphism and intersection with a regular language is the family of recursively enumerable languages. More precisely, we show that RPN coverability (as reachability) languages strictly include the union of context-free languages and Petri net coverability languages. Moreover, we prove that RPN coverability languages and reachability languages of Petri nets are incomparable. We prove that RPN coverability languages are a strict subclass of RPN reachability languages. In addition, we establish that the family of RPN languages is closed under union, homomorphism but neither under intersection with a regular language nor under complementation.

From an algorithmic point of view, we show that, as for Petri nets, coverability, termination, boundedness, and finiteness are EXPSpace-complete. Thus the increase of expressive power does not entail a corresponding increase in complexity. In order to solve the coverability problem, we show that if there exists a covering sequence there exists a ‘short’ one (i.e. with a length at most doubly exponential w.r.t. the size of the input). In order to solve the termination problem, we consider two cases for an infinite sequence depending (informally speaking) whether the depth of the trees corresponding to states are bounded or not along the sequence. For the unbounded case, we introduce the abstract graph that expresses the ability to create threads from some initial state. The decidability of the finiteness and boundedness problems are also mainly based on this abstract graph.

Let us mention that this paper is an extended version of [23] that contains new results about expressiveness like the characterization of the RPN coverability languages, decidability and complexity of finiteness and boundedness and we greatly simplified the proofs of coverability, termination, and finiteness. We also provided a more elegant definition of the (now inductive) syntax and the semantics of RPN.

Outline. In section 2, we introduce RPNs and state ordering and establish basic results related to these notions. In section 3, we introduce decision problems and some reductions between them. In section 4, we study the expressiveness of coverability languages. Then in sections 5, 6, and 7 we show that the coverability, termination, boundedness, and finiteness problems are EXPSpace-complete. In section 8, we conclude and give some perspectives to this work.

2 Recursive Petri Nets

2.1 Presentation

The state of an RPN has a structure akin to a ‘directed rooted tree’ of Petri nets. Each vertex of the tree, hereafter *thread*, is an instance of the RPN and possessing

some marking on it. Each of these threads can fire *three* types of transitions. An *elementary* transition updates its own marking according to the usual Petri net firing rule. An *abstract* transition consumes tokens from the thread firing it and creates a new child (thread) for it. The marking of the new thread is determined according to the fired abstract transition. A *cut* transition can be fired by a thread if its marking is greater or equal than some marking. Firing a cut transition, the thread erases itself and all of its descendants. Moreover, it creates tokens in its parent, which are specified by the abstract transition that created it.

Definition 1 (Recursive Petri Net). A Recursive Petri Net is a 6-tuple $\mathcal{N} = \langle P, T, W^+, W^-, \Omega \rangle$ where:

- P is a finite set of places;
- $T = T_{el} \uplus T_{ab} \uplus T_{\tau}$ is a finite set of transitions with $P \cap T = \emptyset$, and T_{el} (respectively T_{ab}, T_{τ}) is the subset of elementary (respectively abstract, cut) transitions;
- W^- is the $\mathbb{N}^{P \times T}$ backward incidence matrix;
- W^+ are the $\mathbb{N}^{P \times (T_{el} \uplus T_{ab})}$ forward incidence matrix;
- $\Omega : T_{ab} \rightarrow \mathbb{N}^P$ is a function that labels every abstract transition with a initial marking;

Figure 1 graphically describes an example of an RPN with:

$$\begin{aligned} P &= \{p_{ini}, p_{fin}, p_{beg}, p_{end}\} \cup \{p_{b_i}, p_{a_i} : i \leq 2\}; \\ T_{el} &= \{t_{b_1}, t_{b_3}, t_{a_1}, t_{a_3}, t_{sa}, t_{sb}\}; T_{ab} = \{t_{beg}, t_{b_2}, t_{a_2}\}; \\ T_{\tau} &= \{t_{\tau_1}, t_{\tau_2}\}. \end{aligned}$$

and for instance $W^-(p_{ini}, t_{beg}) = 1$ and $\Omega(t_{b_2}) = p_{beg}$ (where p_{beg} denotes the marking with one token in place p_{beg} and zero elsewhere).

For brevity reasons, we denote by $W^+(t)$ a vector in \mathbb{N}^P , where for all $p \in P$, $W^+(t)(p) = W^+(p, t)$, and we do the same for $W^-(t)$.

A *concrete state* s of an RPN is a labeled tree representing relations between threads and their associated markings. Every vertex of s is a thread and edges are labeled by abstract transitions. We introduce a countable set \mathcal{V} of vertices in order to pick new vertices when necessary.

Definition 2 (State of an RPN). A concrete state (in short, a state) s of an RPN is a tree over the finite set of vertices $V_s \subseteq \mathcal{V}$, inductively defined as follows:

- either $V_s = \emptyset$ and thus $s = \emptyset$ is the empty tree;
- or $V_s = \{r_s\} \uplus V_1 \uplus \dots \uplus V_k$ with $0 \leq k$ and $s = (r_s, m_0, \{(m_i, s_i)\}_{1 \leq i \leq k})$ is defined as follows:
 - r_s is the root of s labelled by a marking $m_0 \in \mathbb{N}^P$;
 - For all $i \leq k$, s_i is a state over $V_i \neq \emptyset$
and there is an edge $r_s \xrightarrow{m_i} r_{s_i}$ with $m_i \in \{W^+(t)\}_{t \in T_{ab}}$.

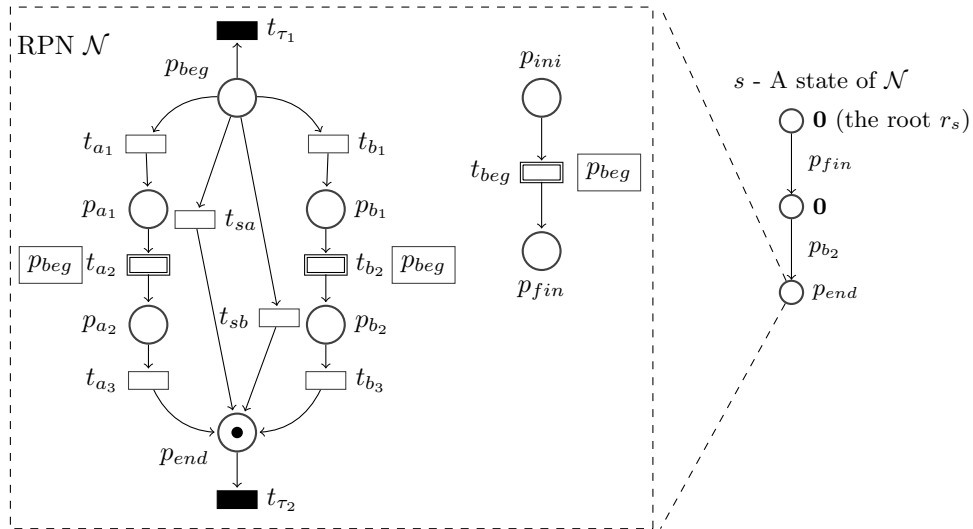


Fig. 1. An example of a marked RPN.

For all $u, v \in V_s$, one denotes $M_s(u)$ the marking labelling u and when $u \xrightarrow{m}_s v$, one writes $\Lambda(u, v) := m$. State s_v is the (maximal) subtree of s rooted in v .

While the set of vertices V_s will be important for analyzing the behavior of a firing sequence in an RPN, one can omit it and get a more abstract representation of the state. Note that contrary to the previous definition where $\{(m_i, s_i)\}_{1 \leq i \leq k}$ was a set, in the following definition we need a multiset $Child_s$.

Definition 3 (Abstract state of an RPN). An abstract state s of an RPN is inductively defined as follows:

- either $s = \emptyset$ is the empty set ;
- or $s = (m_s, Child_s)$ where $m_s \in \mathbb{N}^P$ and $Child_s$ is a finite multiset of pairs (m', s') where $m' \in \{W^+(t)\}_{t \in T_{ab}}$ and s' is an abstract state different from \emptyset .

Given a concrete state s , we denote by $[s]$ its abstract state. Except if explicitly stated, a state is a concrete state.

In the other direction, given an abstract state s , one recovers its set of concrete states by picking an arbitrary set of vertices $V_s \subseteq \mathcal{V}$ of appropriate cardinality and, inductively, arbitrarily splitting V_s between the root and the pairs (m, s') .

For example, on the right side of Figure 1, there is a (concrete) state of the RPN \mathcal{N} . This state consists of three threads with markings $\mathbf{0}$, $\mathbf{0}$, and p_{end} (where $\mathbf{0}$ is the *null marking*) and two edges with the labels $W^+(t_{beg})$ and $W^+(t_{b2})$.

Let s be a state of some RPN. Every thread u different from the root has an unique *parent*, denoted by $prd(u)$. The *descendants* of a thread u consists

of threads in the subtree rooted in u including u itself. We denote this set by $Des_s(u)$. For $m \in \mathbb{N}^P$, denote by $s[r, m] := (r, m, \emptyset)$, the state consisting of a single vertex r whose marking is m . As usual, two markings $m, m' \in \mathbb{N}^P$, over a set of places P , are partially ordered as follows: $m \leq m'$ if for all places $p \in P$, $m(p) \leq m'(p)$.

Definition 4 (Operational semantics). Let $s = (r, m_0, \{(m_i, s_i)\}_{1 \leq i \leq k})$ be a state. Then the firing rule $s \xrightarrow{(v,t)} s'$ where $v \in V_s$ and $t \in T$ is inductively defined as follows:

- Let $t \in T_{el}$ such that $W^-(t) \leq m_0$, then one has $s \xrightarrow{r,t} (r, m_0 - W^-(t) + W^+(t), \{(m_i, s_i)\}_{i \leq k})$
- Let $t \in T_{ab}$ such that $W^-(t) \leq m_0$, then one has $s \xrightarrow{r,t} (r, m_0 - W^-(t), \{(m_i, s_i)\}_{i \leq k+1})$ where $m_{k+1} = W^+(t)$, $s_{k+1} = s[v, \Omega(t)]$ with $v \in \mathcal{V} \setminus V_s$
- Let $t \in T_\tau$ such that $W^-(t) \leq m_0$, then one has $s \xrightarrow{r,t} \emptyset$
- Let $i \leq k$ such that $s_i \xrightarrow{v_i,t} s'_i$
if $s'_i = \emptyset$ then $s \xrightarrow{v_i,t} (m_0 + m_i, \{(m_j, s_j)\}_{1 \leq j \neq i \leq k})$
else $s \xrightarrow{v_i,t} (m_0, \{m_j, s_j\}_{1 \leq j \neq i \leq k} \cup \{m_i, s'_i\})$

Figure 2 illustrates a sequence of transition firings in the RPN described by Figure 1. The first transition $t_{beg} \in T_{ab}$ is fired by the root. Its firing results in a state for which the root has a new child (denoted by v) and a new outgoing edge with label p_{fin} . The marking of the root is decreased to $\mathbf{0}$ and v is initially marked by $\Omega(t_{beg}) = p_{beg}$. The second firing is due to an elementary transition $t_{b_1} \in T_{el}$ which is fired by v . Its firing results in a state for which the marking of v is changed to $M'_s(v) = M_s(v) + W^+(t_{b_1}) - W^-(t_{b_1}) = p_{b_1}$. The fifth transition to be fired is the cut transition t_{τ_2} , fired by the thread with the marking p_{end} (denoted by w). Its firing results in a state where the thread w is erased, and the marking of its parent is increased by $W^+(t_{b_2}) = p_{b_2}$.

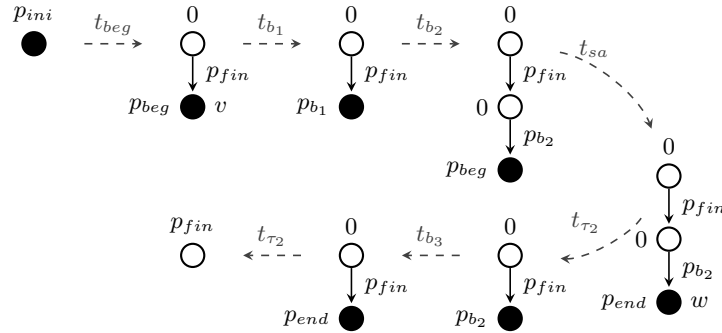


Fig. 2. Firing sequence for the RPN in Figure 1

A *firing sequence* is a sequence of transition firings, written in a detailed way: $s_0 \xrightarrow{(v_1, t_1)} s_1 \xrightarrow{(v_2, t_2)} \dots \xrightarrow{(v_n, t_n)} s_n$, or when the context allows it, in a more concise way like $s_0 \xrightarrow{\sigma} s_n$ for $\sigma = (v_1, t_1)(v_2, t_2) \dots (v_n, t_n)$. Let $\sigma \in T^*$ with $\sigma = t_1 \dots t_n$ and v be a vertex, (v, σ) is an abbreviation for $(v, t_1) \dots (v, t_n)$. When we deal with several nets, we indicate by a subscript in which net, say \mathcal{N} , the firing sequence takes place: $s_0 \xrightarrow{\sigma}_{\mathcal{N}} s_n$. Infinite firing sequences are similarly defined. In a firing sequence, a thread v that has been deleted is *never reused* (which is possible since \mathcal{V} is countable). A thread is *final* (respectively *initial*) w.r.t. σ if it occurs in the final (respectively initial) state of σ . We say that $v \in Des_{\sigma}(u)$ if there exists $i \leq n$ such that $v \in Des_{s_i}(u)$. We call σ' a *subsequence* of σ , denoted by $\sigma' \sqsubseteq \sigma$, if there exists k indexes $i_1, i_2 \dots i_k$ such that $1 \leq i_1 < i_2 < \dots < i_k \leq n$ and $\sigma' = (v_{i_1}, t_{i_1})(v_{i_2}, t_{i_2}) \dots (v_{i_k}, t_{i_k})$.

Remark 1. In the sequel, when we write ‘‘RPN \mathcal{N} ’’, we mean $\mathcal{N} = \langle P, T, W^+, W^-, \Omega \rangle$, unless we explicitly write differently. An RPN \mathcal{N} equipped with an initial state s is a *marked RPN* and denoted (\mathcal{N}, s) . Similarly a *marked Petri net* (\mathcal{N}, m) is a Petri net \mathcal{N} equipped with an initial marking m .

For a marked RPN (\mathcal{N}, s_0) , let $Reach(\mathcal{N}, s_0) = \{[s] \mid \exists \sigma \in T^* \text{ s.t. } s_0 \xrightarrow{\sigma} s\}$ be its *reachability set*, i.e. the set of all the reachable *abstract states*.

2.2 An order for Recursive Petri Nets

We now define a quasi-order \preceq on the states of an RPN. Given two states s, s' of an RPN \mathcal{N} , we say that s is *smaller or equal* than s' , denoted by $s \preceq s'$, if there exists a subtree in s' , which is isomorphic to s , where markings are greater or equal on all vertices and edges.

Definition 5. *Let $s \neq \emptyset$ and s' be states of an RPN \mathcal{N} . Then $s \preceq s'$ if there exists an injective mapping f from V_s to $V_{s'}$ such that for all $v \in V_s$:*

1. $M_s(v) \leq M_{s'}(f(v))$, and,
2. for all $v \xrightarrow{m}_s w$, there exists an edge $f(v) \xrightarrow{m'}_{s'} f(w)$ with $m \leq m'$.

In addition, $\emptyset \preceq s$ for all states s .

When $f(r_s)$ is required to be $r_{s'}$, one denotes this relation $s \preceq_r s'$ with $\emptyset \preceq_r s$ if and only if $s = \emptyset$.

Figure 3 illustrates these quasi-orders.

While this is irrelevant for the results presented here, let us mention that checking whether $s \preceq s'$ can be done in polynomial time by adapting a standard algorithm for the subtree problem (see for instance [24]).

Lemma 1. *The relations \preceq and \preceq_r are quasi-orders.*

Proof. Let, s, s', s'' be states of an RPN \mathcal{N} with $s = (r, m_0, \{(m_i, s_i)\}_{1 \leq i \leq k})$, $s' = (r', m'_0, \{(m'_i, s'_i)\}_{1 \leq i \leq k'})$ and $s'' = (r'', m''_0, \{(m''_i, s''_i)\}_{1 \leq i \leq k''})$. Let us show that the relation \preceq is a quasi-order.

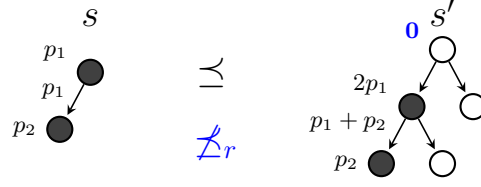


Fig. 3. We have that $s \preceq s'$, but $s \not\preceq_r s'$ because the marking of the root of s' is too small.

1. Reflexivity: the identity function Id on V_s insures that $s \preceq s$.
2. Transitivity: Given $s \preceq s' \preceq s''$, there exist two injective functions $f : V_s \rightarrow V_{s'}$ and $f' : V_{s'} \rightarrow V_{s''}$. Let $g : V_s \rightarrow V_{s''}$ be defined by $g = f' \circ f$. Then g is injective. For any edge $v \xrightarrow{m}_s w$, there exists an edge $f(v) \xrightarrow{m'}_{s'} f(w)$ with $m \leq m'$ and there exists an edge $f'(f(v)) \xrightarrow{m''}_{s''} f'(f(w))$ with $m \leq m' \leq m''$. For all $v \in V_s$, one has $M_s(v) \leq M_{s'}(f(v)) \leq M_{s''}(f'(f(v))) = M_{s''}(g(v))$. Therefore $s \preceq s''$.

The proof for the relation \preceq_r is similar.

Consider the equivalence relation $\simeq := \preceq \cap \preceq^{-1}$. Given a set of states A , one denotes by A/\simeq the quotient set by the equivalence relation \simeq . Observe that $s \simeq s'$ if and only if their abstract representations are equal and that $\simeq = \preceq_r \cap \preceq_r^{-1}$.

A quasi-order \leq on the states of an RPN is *strongly compatible* (as in [25]) if for all states s, s' such that $s \leq s'$ and for all transition firings $s \xrightarrow{(v,t)} s_1$, there exist a state s'_1 and a transition firing $s' \xrightarrow{(v',t')} s'_1$ with $s_1 \leq s'_1$.

Lemma 2. *The quasi-orders \preceq and \preceq_r are strongly compatible.*

Proof. Let $s \preceq s'$ and let f be the mapping associated with the relation \preceq and $s \xrightarrow{(v,t)} s_1$.

Thus $s_v \xrightarrow{(v,t)} s_2$ for some s_2 .

We will exhibit some s'_1 such that $s_1 \preceq s'_1$ with some f' as associated mapping.

Since $M_s(v) \leq M_{s'}(f(v))$, one has $s'_{f(v)} \xrightarrow{(f(v),t)} s'_2$ for some s'_2 and by induction

$s' \xrightarrow{(v,t)} s'_1$ for some s'_1 .

It remains to define f' .

- If $t \in T_{el}$ then $f' = f$;
- If $t \in T_{ab}$ then for all threads u of s , $f'(u) = f(u)$ and if v^* (resp. w^*) is the thread created by the firing (v, t) (resp. $(f(v), t)$) then $f(v^*) = w^*$;
- If $t \in T_\tau$ then f' is equal to f restricted to the remaining vertices.

It is routine to check that the inequalities between corresponding markings of s and s' are fulfilled. The proof for \preceq_r is similar.

These quasi-orders may contain an infinite set of incomparable states (i.e. an infinite *antichain*). For example, see Figure 4 where any two states s_i and s_j are incomparable. Indeed, for any $i < j$: (1) $s_j \not\leq_r s_i$ since $|V_{s_j}| > |V_{s_i}|$ there cannot

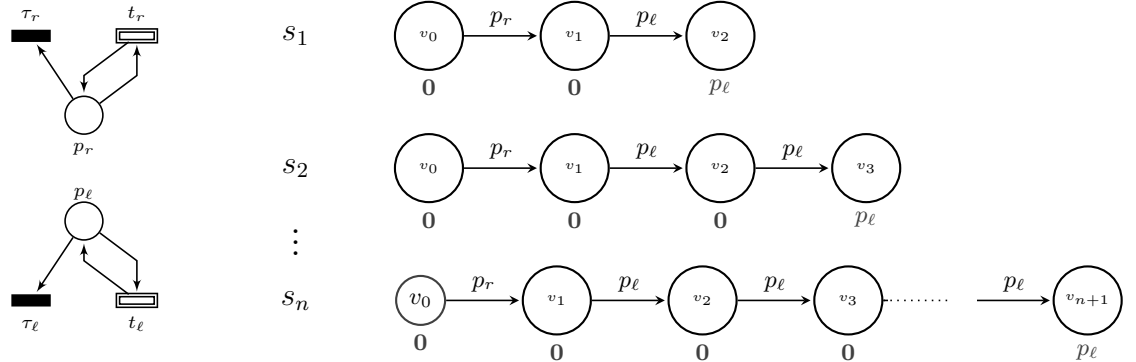


Fig. 4. An RPN with an antichain of states

be any injective function from V_{s_j} to V_{s_i} , and (2) $s_i \not\leq_r s_j$ since for any injective function from V_{s_i} to V_{s_j} , at least one of the edges with the marking p_r would be mapped to an edge with a marking p_l . Since $s \leq_r s'$ implies $s \leq s'$, this is also an antichain for \leq_r .

Observe also that these quasi-orders are not only strongly compatible. They are *transition-preserving compatible* meaning that for all states s, s' such that $s \leq s'$ and for all transition firings $s \xrightarrow{(v,t)} s_1$, there exist s'_1 and a transition firing $s' \xrightarrow{(v',t)} s'_1$ with $s_1 \leq s'_1$. In Petri net, the standard order on \mathbb{N}^P is a well quasi-order which is transition-preserving compatible. The next proposition establishes that such a quasi-order does not exist in RPN.

Proposition 1. *There does not exist a well quasi-order on states of RPN which is transition-preserving compatible.*

Proof. Consider the net of Figure 4 and the family of states $\{s_n\}_{n \geq 1}$. By a simple examination one gets that for all $n \geq 1$, $s_n \xrightarrow{(v_{n+1}, \tau_\ell) \dots (v_1, \tau_\ell)(v_0, \tau_r)} \emptyset$. Moreover for all $n' \neq n$, there does not exist a firing sequence from $s_{n'}$ labelled by $\tau_\ell^{n+1} \tau_r$. Thus for any transition-preserving compatible quasi-order \leq , these states are incomparable establishing that \leq is not a well quasi-order.

Since \leq is not a well quasi-order, RPNs with the relation \leq are not well structured transition systems (WSTS) [25] for which coverability is decidable. Therefore to solve coverability, one needs to find another way.

3 Decision problems and reductions

In this section, we introduce the decision problems that we are going to solve and establish reductions to simpler problems in order to shorten the proofs of subsequent sections.

Let (\mathcal{N}, s_0) be a marked RPN and s_f be a state of \mathcal{N} .

- The *cut problem* asks whether there exists a firing sequence σ such that $s_0 \xrightarrow{\sigma} \emptyset$?
- The *coverability problem* asks whether there exists a firing sequence σ such that $s_0 \xrightarrow{\sigma} s \succeq s_f$?
- The *termination problem* asks whether there exists an infinite firing sequence?
- The *finiteness problem* asks whether $\text{Reach}(\mathcal{N}, s_0)$ is finite?
- The *boundedness problem* asks whether there exists $B \in \mathbb{N}$ such that for all $s \in \text{Reach}(\mathcal{N}, s_0)$ and for all $v \in V_s$, one has $\max(M_s(v)(p))_{p \in P} \leq B$?

Observe that contrary to Petri nets, the finiteness and boundedness problems are different and not equivalent. Indeed, an RPN can be bounded while due to the unbounded number of vertices, its reachability set can be infinite.

We introduce the “rooted” version of the above problems: for these versions, s_0 is required to be some $s[r, m_0]$. In order to establish a reduction from the general problems to their rooted versions, given a marked RPN (\mathcal{N}, s_0) , we build a marked RPN $(\mathring{\mathcal{N}}, s[r, \mathring{m}_0])$ that in a way simulates the former marked RPN. We do this by adding a place p_v for every vertex $v \neq r$ of s_0 and we add an abstract transition t_v that consumes a token from this place and creates a new vertex with initial marking in $M_{s_0}(v) + \sum_{v \xrightarrow{m_{v'}}_{s_0} v'}$. This will allow to create the children of v in s_0 (see Figure 5). In order to similarly proceed in the root, $\mathring{m}_0 = M_{s_0}(r) + \sum_{r \xrightarrow{m_{v'}}_{s_0} v'}$.

Definition 6. Let (\mathcal{N}, s_0) be a marked RPN. Then $(\mathring{\mathcal{N}}, \mathring{s}_0)$ is defined by:

- $\mathring{P} = P \cup \{p_v \mid v \in V_{s_0} \setminus \{r_{s_0}\}\}$;
- $\mathring{T}_{ab} = T_{ab} \cup T_V$, $\mathring{T}_\tau = T_\tau$, $\mathring{T}_{el} = T_{el}$ with $T_V = \{t_v \mid v \in V_s \setminus \{r_s\}\}$;
- for all $t \in T$, one has $\mathring{W}^-(t) = W^-(t)$ and all $t \in T_{ab} \cup T_{el}$, $\mathring{W}^+(t) = W^+(t)$;
- for all $t_v \in T_V$ and $u \xrightarrow{m_v}_{s_0} v$, $\mathring{W}^-(t_v) = p_v$ and $\mathring{W}^+(t_v) = m_v$;
- for all $t \in T_{ab}$, $\mathring{\Omega}(t) = \Omega(t)$;
- for all $t_v \in T_V$, $\mathring{\Omega}(t_v) = M_{s_0}(v) + \sum_{v \xrightarrow{m_{v'}}_{s_0} v'} p_{v'}$;
- $\mathring{s}_0 = s[r, M_{s_0}(r_s) + \sum_{r_{s_0} \xrightarrow{m_v}_{s_0} v} p_v]$.

Let $m \in \mathbb{N}^{\mathring{P}}$, we denote by $m|_P \in \mathbb{N}^P$ the projection of m on P . Let s be a state of $\mathring{\mathcal{N}}$, we denote $s|_P$ a state of \mathcal{N} obtained by projecting every marking of s on P .

Observations.

1. The encoding size of (\mathcal{N}, s_0) is linear w.r.t. the encoding size of $(\mathring{\mathcal{N}}, \mathring{s}_0)$.

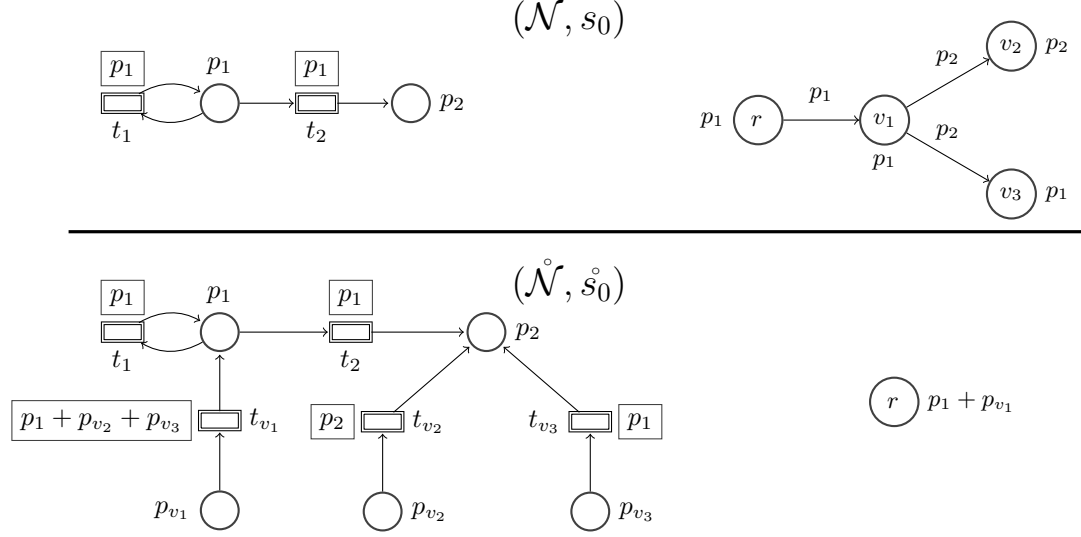


Fig. 5. From a marked RPN to a rooted one

2. Let $e := (v_i)_{0 \leq i \leq k}$ be an enumeration of V_{s_0} such that $v_0 = r_{s_0}$ and for all $0 < i \leq k$, $\text{prd}(v_i) \in \{v_j\}_{j < i}$. Consider $\sigma_{s_0}^e = (\text{prd}(v_i), t_{v_i})_{i=1}^k$. Such an enumeration is called *consistent*. By construction of $\hat{\mathcal{N}}$, $\hat{s}_0 \xrightarrow{\sigma_{s_0}^e} \hat{\mathcal{N}} s'_0$ with $s'_{0|P} = s_0$ and all places of P_V unmarked in s'_0 .
3. Let $\hat{s}_0 \xrightarrow{\sigma} \hat{\mathcal{N}} s$. Then by construction, for all $v \in V_{s_0} \setminus \{r_{s_0}\}$, there is at most one occurrence of t_v which furthermore is fired in $\text{prd}(v)$. Moreover since these firings consume tokens in P_V that were not used for firings of T , they can be pushed at the beginning of σ (denoted by σ_1) and completed by the missing firings of T_V in σ (denoted by σ_2) getting a consistent enumeration e . Summarizing, denoting $\sigma_{|\mathcal{N}}$, σ without the firings of T_V , one gets that:

- (1) $\hat{s}_0 \xrightarrow{\sigma_1 \sigma_{|\mathcal{N}}} \hat{\mathcal{N}} s$,
- (2) $\hat{s}_0 \xrightarrow{\sigma \sigma_2} \hat{\mathcal{N}} s'$ and
- (3) $s_0 \xrightarrow{\sigma_{|\mathcal{N}}} \mathcal{N} s''$ with $s'_{|P} = s''$ and all places of P_V are unmarked in s' .

Due to observation 2, we immediately get that:

Lemma 3. *Let (\mathcal{N}, s_0) be a marked RPN and $s_0 \xrightarrow{\sigma} \mathcal{N} s$. Then for every consistent enumeration e , there exists a firing sequence $\hat{s}_0 \xrightarrow{\sigma_{s_0}^e \sigma} \hat{\mathcal{N}} s'$ with $s'_{|P} = s$ and all places of P_V are unmarked in s' .*

Due to observation 3, we immediately get that:

Lemma 4. *Let (\mathcal{N}, s_0) be a marked RPN and $\hat{s}_0 \xrightarrow{\sigma} \hat{\mathcal{N}} s$. Then there exist a consistent enumeration e and a decomposition $\sigma_{s_0}^e = \sigma_1 \sigma_2$ such that $\hat{s}_0 \xrightarrow{\sigma_1 \sigma_{|\mathcal{N}}} \hat{\mathcal{N}} s$*

$s, \dot{s}_0 \xrightarrow{\sigma\sigma_2}_{\dot{\mathcal{N}}} s'$ and $s_0 \xrightarrow{\sigma|_{\mathcal{N}}}_{\dot{\mathcal{N}}} s''$ with $s'_{|P} = s''$ and all places of P_V are unmarked in s' .

Due to the previous lemmas, we get that:

Proposition 2. *The cut (resp. coverability, termination, finiteness, boundedness) problem is polynomially reducible to the rooted cut (resp. coverability, termination, finiteness, boundedness) problem.*

Proof. Let (\mathcal{N}, s_0) be a marked RPN and s_f be a state of \mathcal{N} . Define \dot{s}_f a state of $\dot{\mathcal{N}}$ be as s_f with in all markings of s_f , all places of $\dot{P} \setminus P$ unmarked.

- Assume that there exists $s_0 \xrightarrow{\sigma}_{\mathcal{N}} \emptyset$. Then by Lemma 3, $\dot{s}_0 \xrightarrow{\sigma_{s_0}^e \sigma}_{\dot{\mathcal{N}}} \emptyset$. Assume that there exists $\dot{s}_0 \xrightarrow{\sigma}_{\dot{\mathcal{N}}} \emptyset$ which means that the last transition is fired in the root and is a cut transition. Then by Lemma 4, $s_0 \xrightarrow{\sigma|_{\mathcal{N}}}_{\dot{\mathcal{N}}} s''$ for some s'' . Since the last firing of $\sigma|_{\mathcal{N}}$ is the cut transition fired in the root $s'' = \emptyset$.
- Assume that there exists $s_0 \xrightarrow{\sigma}_{\mathcal{N}} s \succeq s_f$. Then by Lemma 3, $\dot{s}_0 \xrightarrow{\sigma_{s_0}^e \sigma}_{\dot{\mathcal{N}}} \dot{s}$ with $\dot{s}_{|P} = s$. Thus $\dot{s} \succeq \dot{s}_f$. Assume that there exists $\dot{s}_0 \xrightarrow{\sigma}_{\dot{\mathcal{N}}} s \succeq \dot{s}_f$. Then by Lemma 4, there exists σ_2 a firing sequence of T_V with $\dot{s}_0 \xrightarrow{\sigma\sigma_2}_{\dot{\mathcal{N}}} s'$, $s_0 \xrightarrow{\sigma|_{\mathcal{N}}}_{\dot{\mathcal{N}}} s''$ and $s'_{|P} = s''$. Since σ_2 only creates vertices and deletes tokens from P_V , $s' \succeq \dot{s}_f$. Thus $s'' \succeq s_f$.
- Assume that there exists $s_0 \xrightarrow{\sigma}_{\mathcal{N}}$ with σ infinite. Then by Lemma 3, $\dot{s}_0 \xrightarrow{\sigma_{s_0}^e \sigma}_{\dot{\mathcal{N}}}$. Assume that there exists $\dot{s}_0 \xrightarrow{\sigma}_{\dot{\mathcal{N}}}$ with σ infinite. Then by Lemma 4, $s_0 \xrightarrow{\sigma|_{\mathcal{N}}}_{\dot{\mathcal{N}}}$ with $\sigma|_{\mathcal{N}}$ infinite since there are only a finite number of firings of T_V .
- Assume that $Reach(\mathcal{N}, s_0)$ is infinite. For all $s \in Reach(\mathcal{N}, s_0)$, define \dot{s} a state of $\dot{\mathcal{N}}$ as s with all places of P_V in markings of s unmarked. Due to Lemma 3, $\dot{s} \in Reach(\dot{\mathcal{N}}, \dot{s}_0)$. Since this mapping is injective, $Reach(\dot{\mathcal{N}}, \dot{s}_0)$ is infinite. Assume that $Reach(\dot{\mathcal{N}}, \dot{s}_0)$ is infinite. Let $s \in Reach(\dot{\mathcal{N}}, \dot{s}_0)$. Due to Lemma 4, consider $s \xrightarrow{\sigma_2}_{\dot{\mathcal{N}}} s'$ and $s_0 \xrightarrow{\sigma|_{\mathcal{N}}}_{\dot{\mathcal{N}}} s''$ with $s'_{|P} = s''$ and all places of P_V unmarked in s' . Thus $s'' \in Reach(\mathcal{N}, s_0)$. The mapping from s to s'' is not injective. However, the inverse image of s'' by this mapping is finite since there are a finite number of consistent enumerations and prefixes of such enumerations. Thus $Reach(\mathcal{N}, s_0)$ is infinite.
- Assume that (\mathcal{N}, s_0) is unbounded. For all $s \in Reach(\mathcal{N}, s_0)$, define \dot{s} a state of $\dot{\mathcal{N}}$ as s with all places of P_V in markings of s unmarked. Due to Lemma 3, $\dot{s} \in Reach(\dot{\mathcal{N}}, \dot{s}_0)$. Thus $(\dot{\mathcal{N}}, \dot{s}_0)$ is unbounded. Assume that $(\dot{\mathcal{N}}, \dot{s}_0)$ is unbounded. By construction, the marking of places in P_V is bounded. Let $s \in Reach(\dot{\mathcal{N}}, \dot{s}_0)$. Due to Lemma 4, consider $s \xrightarrow{\sigma_2}_{\dot{\mathcal{N}}} s'$ and $s_0 \xrightarrow{\sigma|_{\mathcal{N}}}_{\dot{\mathcal{N}}} s''$ with $s'_{|P} = s''$ and all places of P_V unmarked in s' . Thus $s'' \in Reach(\mathcal{N}, s_0)$. Since for all vertex v of s , v is also present in s'' and for all $p \in P$, $M_s(v)(p) = M_{s''}(v)(p)$. Then $Reach(\mathcal{N}, s_0)$ is unbounded.

Let σ be a firing sequence. A thread is *extremal* w.r.t. σ if it is an initial or final thread.

Definition 7. Let \mathcal{N} be an RPN. Then $T_{ret} \subseteq T_{ab}$, the set of returning transitions is defined by:

$$\{t \in T_{ab} \mid \exists s[r, \Omega(t)] \xrightarrow{\sigma} \emptyset\}$$

For all $t \in T_{ret}$, we define σ_t to be some arbitrary shortest *returning sequence* (i.e. $s[r, \Omega(t)] \xrightarrow{\sigma_t} \emptyset$). We now introduce $\widehat{\mathcal{N}}$ obtained from \mathcal{N} by adding elementary transitions that mimic the behaviour of a returning sequence. Observe that the size of $\widehat{\mathcal{N}}$ is linear w.r.t. the size of \mathcal{N} .

Definition 8. Let \mathcal{N} be an RPN. Then $\widehat{\mathcal{N}} = \langle P, \widehat{T}, \widehat{W}^+, \widehat{W}^-, \Omega \rangle$ is defined by:

- $\widehat{T}_{ab} = T_{ab}$, $\widehat{T}_\tau = T_\tau$, $\widehat{T}_{el} = T_{el} \uplus \{t^r \mid t \in T_{ret}\}$;
- for all $t \in T$, $\widehat{W}^-(t) = W^-(t)$ and all $t \in T_{ab} \cup T_{el}$, $\widehat{W}^+(t) = W^+(t)$;
- for all $t \in T_{ab}$, $\widehat{\Omega}(t) = \Omega(t)$;
- for all $t \in T_{ret}$, $\widehat{W}^-(t^r) = W^-(t)$ and $\widehat{W}^+(t^r) = W^+(t)$.

Figure 6 has an example of an RPN \mathcal{N} and its $\widehat{\mathcal{N}}$.

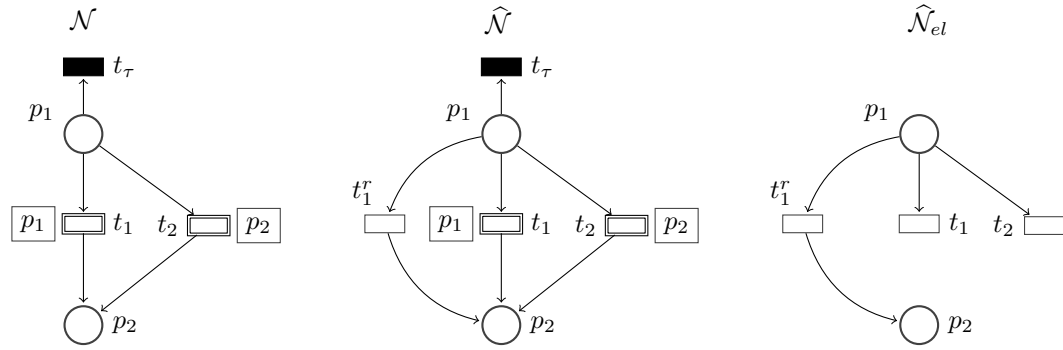


Fig. 6. From \mathcal{N} to $\widehat{\mathcal{N}}$ and $\widehat{\mathcal{N}}_{el}$

Note that since $\widehat{\mathcal{N}}$ enlarges \mathcal{N} by adding transitions and that any firing of t^r in $\widehat{\mathcal{N}}$ can be replaced by the firing of $t\sigma_t$ in \mathcal{N} we get:

Proposition 3. Let (\mathcal{N}, s_0) be a marked RPN. Then $Reach(\mathcal{N}, s_0) = Reach(\widehat{\mathcal{N}}, s_0)$.

We call a firing sequence σ *omniscient* if any thread created during its firing is a final thread.

Proposition 4. Let (\mathcal{N}, s_0) be a marked RPN and $s_0 \xrightarrow{\sigma} s$. Then there exists a firing sequence $s_0 \xrightarrow{\widehat{\sigma}} s$ such that $\widehat{\sigma}$ is omniscient.

Proof. Assume that we have an extremal thread u which fires $t \in T_{ab}$ creating a non final thread v that disappears by a matching cut transition $(v, t_\tau) \in \sigma$ for

$t_\tau \in T_\tau$. One builds σ' by (1) deleting from σ the transition (u, t) , (2) deleting all the firings from $Des_\sigma(v)$ in σ and (3) replacing the transition (v, t_τ) by (u, t') . We claim that $s \xrightarrow{\sigma} s'$. Indeed in u the transition (u, t') has the same incidence in u as the transition (u, t) followed by (v, t_τ) ('anticipating' (v, t_τ) only add tokens in intermediate states) and the other deleted firings are performed by threads in $Des_\sigma(v)$ which do not exist anymore. By taking $\widehat{\sigma}$ the sequence obtained by iterating the process, we get the omniscient sequence.

In order to recover from a sequence in $\widehat{\mathcal{N}}$ a sequence in \mathcal{N} , for every $t \in T_{ret}$ one has to simulate the firings of a transition t^r by sequence σ_t . Therefore bounding the length of σ_t is a critical issue. Recall that in [4], Rackoff showed that the coverability problem for Petri nets belongs to EXPSPACE. More precisely, he proved that if there exists a covering sequence, then there exists a 'short' one:

Theorem 1 (Rackoff [4]). *Let \mathcal{N} be a Petri net, m_{ini}, m_{tar} be markings and σ be a firing sequence such that $m_{ini} \xrightarrow{\sigma} m \geq m_{tar}$. Then there exists a sequence σ' such that $m_{ini} \xrightarrow{\sigma'} m' \geq m_{tar}$ with $|\sigma'| \leq 2^{2^{cn \log n}}$ for some constant c and n being the size of (\mathcal{N}, m_{tar}) .*

A surprising consequence of Rackoff's proof is that the length of the minimal coverability sequence does not depend on the initial marking of the net.

Proposition 5. *Let \mathcal{N} be an RPN and $t \in T_{ret}$. Then the returning sequence σ_t fulfills $|\sigma_t| \leq 2^{2^{dn \log n}}$ for some constant d and $n = \text{size}(\mathcal{N})$.*

Proof. Let us enumerate $T_{ret} = \{t_1, \dots, t_K\}$ in such a way that $i < j$ implies $|\sigma_{t_i}| \leq |\sigma_{t_j}|$. Observe first that the shortest returning sequences do not include firings of abstract transitions not followed by a matching cut transition since it could be omitted as it only deletes tokens in the thread. We argue by induction on $k \leq K$ that:

$$|\sigma_{t_k}| < 2^{k \cdot 2^{cn \log n}} \quad \text{where } c \text{ is the Rackoff constant}$$

For $k = 1$, we know that σ_{t_1} has a minimal length over all returning sequences. Hence there are no cuts in σ_{t_1} except the last one. Due to the above observation, σ_{t_1} only includes firing of elementary transitions. Thus the Rackoff bound of Theorem 1 applies for a covering of some final marking.

Assume that the result holds for all $i < k$. Due to the requirement on lengths, σ_{t_k} only includes cuts from threads created by $t_i \in T_{ret}$ with $i < k$. Thus by Proposition 4 we get a sequence $\widehat{\sigma}_{t_k} \cdot (r, t_\tau)$ in $\widehat{\mathcal{N}}$ (where r is the root and $t_\tau \in T_\tau$). The sequence $\widehat{\sigma}_{t_k}$ consists of only elementary transitions and does not contain any transition t_i^r with $i \geq k$. The marking of r reached by $\widehat{\sigma}_{t_k}$ covers some final marking, hence by Theorem 1 there exists a covering sequence $\widehat{\sigma}'_{t_k}$ such that $|\widehat{\sigma}'_{t_k}| \leq 2^{2^{cn \log n}}$. Since $\widehat{\sigma}_{t_k}$ does not contain firing of t_i^r with $i \geq k$ this also holds for $\widehat{\sigma}'_{t_k}$. Substituting any firing of t_i^r by σ_{t_i} , one gets a corresponding sequence σ'_{t_k} in \mathcal{N} . Using the induction hypothesis, one gets that the length of σ'_{t_k} fulfills:

$$|\sigma'_{t_k}| \leq |\widehat{\sigma}'_{t_k}| 2^{(k-1) \cdot 2^{cn \log n}} \leq 2^{2^{cn \log n}} \cdot 2^{(k-1) \cdot 2^{cn \log n}} \leq 2^{k \cdot 2^{cn \log n}}$$

From minimality of σ_{t_k} , one gets $|\sigma_{t_k}| \leq |\sigma'_{t_k}| \leq 2^{k \cdot 2^{cn \log n}}$ which concludes the proof since

$$\max_{t \in T_{ret}} \{|\sigma_t|\} \leq 2^{|T_{ret}| \cdot 2^{cn \log n}} \leq 2^{n 2^{cn \log n}} \leq 2^{2^{2cn \log n}}.$$

Using the previous proposition, we can compute T_{ret} in exponential space, by enumerating for all abstract transitions, all firing sequences of sufficient length and checking whether they lead to the empty tree.

Below are immediate corollaries from the previous propositions:

Corollary 1. *Let \mathcal{N} be a marked RPN. Then for all $s \xrightarrow{\hat{\sigma}}_{\hat{\mathcal{N}}} s'$, there exists $s \xrightarrow{\sigma}_{\mathcal{N}} s'$ such that $|\sigma| \leq 2^{2^{dn \log n}} |\hat{\sigma}|$ for some constant d and $n = \text{size}(\mathcal{N})$.*

Corollary 2. *Given an RPN \mathcal{N} one can build $\hat{\mathcal{N}}$ in exponential space.*

In order to mimic the behavior of a specific thread in a firing sequence (which will be useful later on), we introduce the Petri net $\hat{\mathcal{N}}_{el}$. The size of $\hat{\mathcal{N}}_{el}$ is also linear w.r.t. the size of \mathcal{N} .

Definition 9. *Let \mathcal{N} be an RPN. Then the Petri net $\hat{\mathcal{N}}_{el} = \langle P, \hat{T}_{el}, \hat{W}_{el}^+, \hat{W}_{el}^- \rangle$ is defined by:*

- $\hat{T}_{el} = \hat{T} \setminus T_{\tau}$;
- For all $t \in \hat{T}_{el} \setminus T_{ab}$, $\hat{W}_{el}^-(t) = \hat{W}^-(t)$ and $\hat{W}_{el}^+(t) = \hat{W}^+(t)$;
- For all $t \in T_{ab}$, $\hat{W}_{el}^-(t) = \hat{W}^-(t)$ and $\hat{W}_{el}^+(t) = 0$.

Figure 6 has an example of an RPN \mathcal{N} and its $\hat{\mathcal{N}}_{el}$.

As for $\hat{\mathcal{N}}$, one can build $\hat{\mathcal{N}}_{el}$ in exponential space.

Observation. The main (straightforward) property of $\hat{\mathcal{N}}_{el}$ is the following one. Let $\sigma \in \hat{T}_{el}^*$ with n_t the number of occurrences of t in σ . Then $m_0 \xrightarrow{\sigma}_{\hat{\mathcal{N}}_{el}} m$ if and only if $s[r, m_0] \xrightarrow{(r, \sigma)}_{\hat{\mathcal{N}}} s$ with $V_s = \{r\} \cup \bigcup_{t \in T_{ab}} \{v_{t,1}, \dots, v_{t,n_t}\}$, $M_s(r) = m$ and for all v_{t_i} , $r \xrightarrow{W^+(t)}_s v_{t_i}$ and $M_s(v_{t_i}) = \Omega(t)$.

4 Expressiveness

The expressiveness of a formalism may be defined by the family of languages that it can generate. In [19], the expressiveness of RPNs was studied using reachability languages. However, using reachability languages as specification languages has an inconvenience since the emptiness problem for these languages is not elementary [26] for Petri nets, so it is also not elementary, at least, for RPN. We propose to characterize the expressive power of RPN by studying the family of coverability languages which is sufficient to express most of the usual reachability properties since many of them reduce to check that no reachable state may cover a bad marking in a thread.

The characterization of the expressive power by means of covering languages has been done for Petri nets (studied in the book of Peterson [27]), and more recently, for Well Structured Transition Systems (WSTS) [28] and for monotonic extensions of Petri nets like reset-transfer Petri nets, ν -Petri nets, unordered Petri nets [29,30]. More properties are decidable for VASS covering languages than for VASS reachability languages. For instance, universality for reachability languages is undecidable for 1-VASS [31] and then co-finiteness is also undecidable but these two properties are both decidable for VASS covering languages [32]; moreover, it is Ackermann-complete for 1-VASSs [33]. Generally, the universality of both reachability and coverability of WSTS languages is undecidable [28].

So we equip any transition t with a *label* $\lambda(t) \in \Sigma \cup \{\varepsilon\}$ where Σ is a finite alphabet and ε is the empty word. The labelling is extended to transition sequences in the usual way. Thus given a labelled marked RPN (\mathcal{N}, s_0) and a finite subset of states S_f , the (coverability) language $\mathcal{L}_C(\mathcal{N}, s_0, S_f)$ is defined by:

$$\mathcal{L}_C(\mathcal{N}, s_0, S_f) = \{\lambda(\sigma) \mid \exists s_0 \xrightarrow{\sigma} s \succeq s_f \wedge s_f \in S_f\}$$

i.e. the set of labellings for sequences covering some state of S_f in \mathcal{N} .

We now study the family of RPN coverability languages both from the point of view of expressiveness and closure under multiple operations.

Proposition 6. *The family of coverability languages of RPNs is closed under union.*

Proof. We closely follow the classic proof that the family of Petri net languages is closed under union, i.e. adding a place and two extra transitions that have to be fired in the beginning of the firing sequence in order to decide in which of the Petri net one fires. Due to the correspondence between firing sequences of (\mathcal{N}, s_0) and those of $(\tilde{\mathcal{N}}, \tilde{s}_0)$, established in the previous section, one can assume w.l.o.g. that the initial markings of the RPNs have a single vertex. Consider two labelled marked RPNs with final states $(\mathcal{N}, s[r, m_0], S_f)$ and $(\mathcal{N}', s[r', m'_0], S'_f)$. Let us define $\tilde{\mathcal{N}}$ as follows. Its set of places is the disjoint union of P and P' with three additional places p_0 , p and p' . Its set of transitions is the disjoint union of T and T' with four additional elementary transitions t_b , t_c , t'_b and t'_c .

- For all $t \in T$, $\tilde{W}^-(t) = W^-(t) + p$ and when $t \notin T_\tau$ $\tilde{W}^+(t) = W^+(t)$
- For all $t \in T'$, $\tilde{W}^-(t) = W'^-(t) + p'$ and when $t \notin T'_\tau$ $\tilde{W}^+(t) = W'^+(t)$
- For all $t \in T_{ab}$, $\tilde{\Omega}(t) = \Omega(t) + p$
- For all $t \in T'_{ab}$, $\tilde{\Omega}(t) = \Omega'(t) + p'$
- $\tilde{W}^-(t_b) = \tilde{W}^-(t'_b) = p_0$, $\tilde{W}^+(t_b) = m_0 + p$, $\tilde{W}^+(t'_b) = m'_0 + p'$
- $\tilde{W}^-(t_c) = p$, $\tilde{W}^+(t_c) = 2p$, $\tilde{W}^-(t'_c) = p'$, $\tilde{W}^+(t'_c) = 2p'$
- \tilde{S}_f is obtained from the union $S_f \cup S'_f$ by adding a token in place p (resp. p') of all markings of states of S_f (respectively S'_f).
- For all $t \in T$, $\tilde{\lambda}(t) = \lambda(t)$ and for all $t \in T'$, $\tilde{\lambda}(t) = \lambda'(t)$

- For all $t \in \{t_b, t_c, t'_b, t'_c\}$, $\tilde{\lambda}(t) = \varepsilon$.
- The initial state of $\tilde{\mathcal{N}}$ is $s[\tilde{r}, p_0]$.

Let us prove that $\mathcal{L}(\mathcal{N}, s[r, m_0], S_f) \cup \mathcal{L}(\mathcal{N}', s[r', m'_0], S'_f) \subseteq \mathcal{L}(\tilde{\mathcal{N}}, s[\tilde{r}, p_0], \tilde{S}_f)$. Let σ be a coverability sequence of $(\mathcal{N}, s[r, m_0], S_f)$. The corresponding coverability sequence $\tilde{\sigma}$ of $L(\tilde{\mathcal{N}}, s[\tilde{r}, p_0], \tilde{S}_f)$ is built as follows. Initially, one fires $(\tilde{r}, t_b)(\tilde{r}, t_c)^{\ell_r}$ where ℓ_r is the number of abstract transition firings occurring in σ triggered by r . Then after the creation of a thread v , one inserts $(v, t_c)^{\ell_v}$ firings where ℓ_v is the number of abstract transition firings occurring in σ triggered by v . It is routine to check that $\tilde{\sigma}$ is coverability sequence. The proof for $\mathcal{L}(\mathcal{N}', s[r', m'_0], S'_f)$ is similar.

Let us prove that $\mathcal{L}(\tilde{\mathcal{N}}, s[\tilde{r}, p_0], \tilde{S}_f) \subseteq \mathcal{L}(\mathcal{N}, s[\tilde{r}, m_0], S_f) \cup \mathcal{L}(\mathcal{N}', s[r', m'_0], S'_f)$. Observe that any firing sequence must start by a firing of t_b or t'_b . Let $t_b\tilde{\sigma}$ be a coverability sequence of $(\tilde{\mathcal{N}}, s[\tilde{r}, p_0], \tilde{S}_f)$. Consider the sequence σ obtained by deleting all the firings of t_c in $\tilde{\sigma}$. It is routine to check that σ is a coverability sequence for $(\mathcal{N}, s[r, m_0], S_f)$. The case of a coverability sequence starting by t'_b is similar.

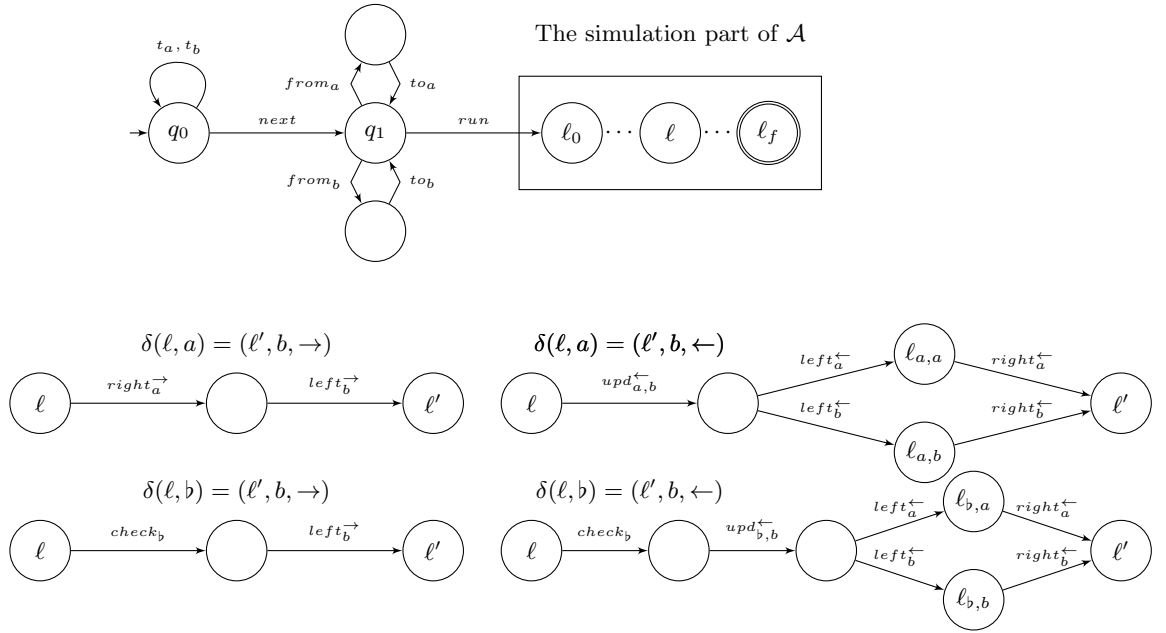
The next theorem has two interesting consequences: the family of RPN coverability languages is not closed under intersection with the family of regular languages. But the family obtained by this intersection is *quite close* to the family of recursively enumerable languages. The result was already stated in Proposition 9 of [34] for the family of RPN reachability languages but the proof was only sketched.

Theorem 2. *Let \mathcal{L} be a recursively enumerable language. Then there exist an RPN language \mathcal{L}' , a regular language \mathcal{R} and a homomorphism h such that $\mathcal{L} = h(\mathcal{L}' \cap \mathcal{R})$.*

Proof. Let $\mathcal{M} = (\Sigma, L, \delta)$ be a Turing machine with its set of states L including ℓ_0 (resp. ℓ_f) the initial (resp. final) state and its transition function δ from $L \times \Sigma \cup \{b\}$ to $L \times \Sigma \times \{\leftarrow, \rightarrow\}$ where b is the blank character.

Let us define a labeled marked RPN \mathcal{N} and an automaton \mathcal{A} . Their common alphabet is the set of transitions of \mathcal{N} and the labeling of the transitions of the RPN is the identity mapping. The intersection of their languages is thus the language of the synchronized product of the two devices. The single final state of \mathcal{N} (to be covered) is the empty tree.

The automaton \mathcal{A} is depicted below (with $\Sigma = \{a, b\}$). In q_0 it allows \mathcal{N} to generate the representation of any word $w \in \Sigma^*$, input of \mathcal{M} . However, this intermediate representation is not suitable for mimicking \mathcal{M} . Thus in q_1 , the intermediate representation is translated into an appropriate one. Once this representation is obtained, it mimics any transition of \mathcal{M} by triggering the firing of several transitions of \mathcal{N} . We will detail this simulation after the specification of \mathcal{N} .



\mathcal{N} is defined as follows. Its set of places is $P = \{p_a \mid a \in \Sigma\} \cup \{root, right, left, start, ret\}$. We now define the set of transitions T . The first subset corresponds to the generation of a representation of the input word of \mathcal{M} .

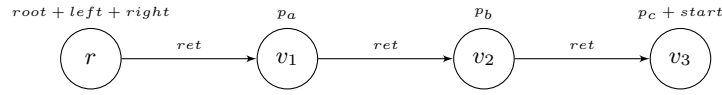
- For all $a \in \Sigma$, $t_a \in T_{ab}$ with $W^-(t_a) = start$, $W^+(t_a) = ret$ and $\Omega(t_a) = start + p_a$;
- $next \in T_{el}$ with $W^-(next) = start$ and $W^+(next) = ret$;
- For all $a \in \Sigma$, $from_a \in T_{\tau}$ $W^-(from_a) = ret + p_a$;
- For all $a \in \Sigma$, $to_a \in T_{ab}$ with $W^-(to_a) = right$, $W^+(to_a) = right$ and $\Omega(to_a) = right + p_a$;
- $run \in T_{el}$ with $W^-(run) = root + ret$ and $W^+(run) = root$

The second subset corresponds to the simulation of \mathcal{M} .

- For all $a \in \Sigma$, $right_a^{\rightarrow} \in T_{\tau}$ with $W^-(right_a^{\rightarrow}) = right + p_a$;
- For all $a \in \Sigma$, $left_a^{\rightarrow} \in T_{ab}$ with $W^-(left_a^{\rightarrow}) = W^+(left_a^{\rightarrow}) = left$ and $\Omega(left_a^{\rightarrow}) = left + p_a$;
- For all $a, b \in \Sigma$, $upd_{a,b}^{\leftarrow} \in T_{el}$ with $W^-(upd_{a,b}^{\leftarrow}) = right + p_a$ and $W^+(upd_{a,b}^{\leftarrow}) = right + p_b$
- For all $a \in \Sigma$, $left_a^{\leftarrow} \in T_{\tau}$ with $W^-(left_a^{\leftarrow}) = left + p_a$
- For all $a \in \Sigma$, $right_a^{\leftarrow} \in T_{ab}$ with $W^-(right_a^{\leftarrow}) = W^+(right_a^{\leftarrow}) = right$ and $\Omega(right_a^{\leftarrow}) = right + p_a$
- $check_b \in T_{el}$ with $W^-(check_b) = W^+(check_b) = right + root$;
- For all $b \in \Sigma$, $upd_{b,b}^{\leftarrow} \in T_{ab}$ with $W^-(upd_{b,b}^{\leftarrow}) = right$, $W^+(upd_{b,b}^{\leftarrow}) = right$ and $\Omega(upd_{b,b}^{\leftarrow}) = right + p_b$.

The initial state is $s[r, root + start + left + right]$.

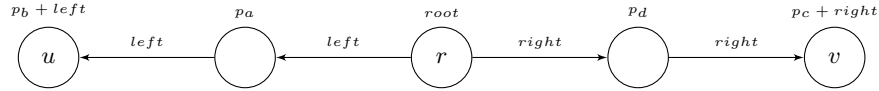
Let us explain how the simulation works. Let abc be the word on the tape of \mathcal{M} . Then firing $(r, t_a)(v_1, t_b)(v_2, t_c)$ one gets:



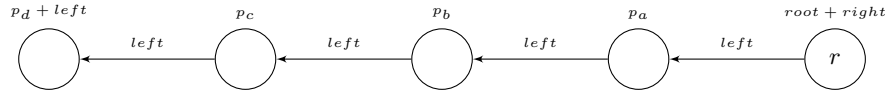
After firing $(v_3, next)(v_3, from_c)(r, to_c)(v_2, from_b)(u_1, to_b)(v_1, from_a)(u_2, to_a)(r, run)$ one gets:



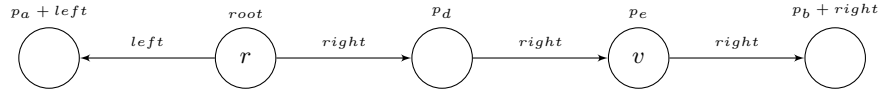
Let us describe the two cases of tape simulation. Assume that the content of the tape is $abcd^{\omega}$ and that the head of \mathcal{M} is over c then the corresponding state is the following one. The “left” branch contains the content of the tape on the left of the head while descending to the leaf and the “right” branch contains the relevant content of the tape on the right of the head (including the cell under the head) while ascending from the leaf. Thus the token in place $right$ points to the thread corresponding to the cell under the head while the token in place $left$ points to the thread corresponding to the cell immediately on the left of the head. The state of \mathcal{M} is the state of \mathcal{A} .



Assume that the content of the tape is $abcd^{\omega}$ and that the head of \mathcal{M} is over the first b then the corresponding state is the following one.



It is routine to check that the simulation works. Let us illustrate it with one example. Assume that the content of the tape is $abcd^{\omega}$, the head of \mathcal{M} is over c and the current state is ℓ . Let $\delta(\ell, c) = (\ell', e, \leftarrow)$. Then after firing $(v, upd_{c,e}^{\leftarrow})(u, left_b^{\leftarrow})(v, right_b^{\leftarrow})$, one gets:



For all $a \in \Sigma$, the homomorphism h maps t_a to a and for all $t \notin \{t_a\}_{a \in \Sigma}$, h maps t to ε .

Obviously, the family of RPNs coverability languages include the family of PN coverability languages. In [19], Proposition 1 establishes that the family of context-free languages is included in family of reachability languages for RPNs. The proof relies on simulating the leftmost derivations of a context-free grammar

within particular two places b_X and e_X per nonterminal symbol X where a token in b_X means that X must be derived and a token in e_X means that the derivation of X into a word has been achieved. In order to adapt this result for the family of coverability languages for RPNs, it is enough to consider w.l.o.g. that the initial symbol I never appears on the right hand side of a rule and to specify $s[r, e_I]$ as final state. We refer the reader to [19] for more details.

Proposition 7. *The family of Context-free languages is included in the family of coverability languages of RPNs.*

Since universality is undecidable for the family of context-free languages, we deduce that universality of the family of RPN coverability languages is undecidable.

Let $\mathcal{L}_1 = \{a^m b^n c^p \mid m \geq n \geq p\}$. Denote by $\mathcal{L}_2 = \{w\tilde{w} \mid w \in \{d, e\}^*\}$ where \tilde{w} is the mirror of w . Let $\mathcal{L}_3 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$. Observe that given the final marking pf we get that the net in Figure 7 has \mathcal{L}_1 as its coverability language, and \mathcal{L}_3 its reachability language.

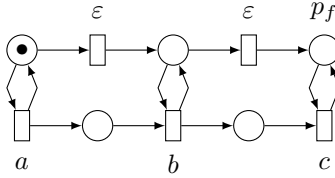


Fig. 7. A Petri net for the languages \mathcal{L}_1 and \mathcal{L}_3

The next proposition witnesses a Petri net language interesting from an expressiveness point of view. A similar result can be found page 179 in Peterson's book [27].

Proposition 8. *\mathcal{L}_1 is the coverability language of some Petri net but it is not a context-free language.*

Proof. Let us recall (a weak version of) Ogden lemma [35]. For any context-free language \mathcal{L} there exists an integer N such for any word $w \in \mathcal{L}$ with N marked positions, there exists a decomposition $w = w_1 w_2 w_3 w_4 w_5$ such that $w_2 w_4$ contains at least a marked position and for all $n \geq 0$, $w_1 w_2^n w_3 w_4^n w_5 \in \mathcal{L}$. The proof that \mathcal{L}_1 is not a context-free language is similar to the proof of the folk result that \mathcal{L}_3 is not a context-free language. Assume that \mathcal{L}_1 is a context-free language and consider the word $w = a^N b^N c^N$ with all c positions marked. So let $w = w_1 w_2 w_3 w_4 w_5$ with the decomposition fulfilling the requirements of Ogden lemma. Since $w' = w_1 w_2^2 w_3 w_4^2 w_5 \in \mathcal{L}_1$, w_2 and w_4 are mono-letter words. Furthermore one of these words is equal to c^q for some $q > 0$. If $w_2 = c^q$ then

$w_4 = c^{q'}$ and thus w' contains too much c 's to belong to \mathcal{L}_1 . If $w_4 = c^q$ then either $w_2 = a^{q'}$, $w_2 = b^{q'}$ or $w_2 = c^{q'}$. Whatever the case, w' misses either a 's or b 's to belong to \mathcal{L}_1 . As mentioned before the coverability language for the net in Figure 7 with final marking p_f is \mathcal{L}_1 .

Using the previous results, the next theorem emphasises the expressive power of coverability languages of RPNs.

Theorem 3. *The family of coverability languages of RPNs strictly include the union of the family of coverability languages of PNs and the family of context-free languages.*

Proof. The inclusion is an immediate consequence of Proposition 7. Consider the language $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

Since (1) by Proposition 6, the family of coverability languages of RPNs is closed under union, (2) \mathcal{L}_1 is a PN language, and (3) the language of palindromes is a context-free language, we deduce that \mathcal{L} is an RPN language.

PN and context-free languages are closed under homomorphism. Since the projection of \mathcal{L} on $\{a, b, c\}$ is the language of Proposition 8, \mathcal{L} is not a context-free language. The projection of \mathcal{L} on $\{d, e\}$ is the language of palindromes. Since it was seen in [36] that the language of (2 letters) palindromes is not a coverability language for any PN we are done.

The next propositions show that the family of coverability languages of an RPN is a particular family of reachability languages of an RPN : the family of *cut languages*. A cut language of an RPN is a reachability language with a single final state \emptyset .

Proposition 9. *The family of cut languages of RPNs is included in the family of coverability languages of RPNs.*

Proof. Due to the correspondence between firing sequences of (\mathcal{N}, s_0) and those of (\mathcal{N}, \hat{s}_0) , established in the previous section, one can assume w.l.o.g. that the initial markings of the RPNs have a single vertex. Let $\mathcal{L}_R(\mathcal{N}, s[r, m_0], \{\emptyset\})$ be such a reachability language.

\mathcal{N}' is obtained by adding places *todo* and *done* and a transition $start \in T'_{ab}$ with:

$$\lambda'(start) = \varepsilon, W'^-(start) = todo, W'^+(start) = done, \Omega'(start) = m_0.$$

Then it is routine to check that $\mathcal{L}_C(\mathcal{N}', s[r, todo], \{s[r, done]\}) = \mathcal{L}_R(\mathcal{N}, s[r, m_0], \{\emptyset\})$.

Establishing the converse inclusion is more intricate.

Proposition 10. *The family of coverability languages of RPNs is included in the family of cut languages of RPNs.*

Proof. Due to the correspondence between firing sequences of (\mathcal{N}, s_0) and those of (\mathcal{N}, \hat{s}_0) , established in the previous section, one can assume w.l.o.g. that the initial markings of the RPNs have a single vertex. Let $\mathcal{L}_C(\mathcal{N}, s[r, m_0], S_f)$ be a coverability RPN language.

Case $\emptyset \in S_f$. Observe that in this case we can reduce S_f to $\{\emptyset\}$. Then \mathcal{N}' is obtained from \mathcal{N} by adding a place $root$ and a cut transition t_{root} with $\lambda'(t_{root}) = \varepsilon$ and $W'^-(t_{root}) = root$. It is routine to check that the reachability language $\mathcal{L}_R(\mathcal{N}', s[r, m_0 + root], \{\emptyset\}) = \mathcal{L}_C(\mathcal{N}, s[r, m_0], \{\emptyset\})$.

Case $\emptyset \notin S_f$. Consider the net \mathcal{N}^* obtained from \mathcal{N} by adding two places $start$ and run with $m_0^* = start$, transitions $t_{run} \in T_{el}$ and $t_{start} \in T_{ab}$ with $\lambda^*(t_{run}) = \lambda^*(t_{start}) = \varepsilon$ and:

$$W^{*-}(t_{run}) = run, W^{*+}(t_{run}) = 2run,$$

$$W^{*-}(t_{start}) = start, W^{*+}(t_{start}) = \mathbf{0} \text{ and } \Omega^*(t_{start}) = m_0 + run.$$

- For all $t \in T_{el}$, $W^{*-}(t) = W^{*-}(t) + run$ and $W^{*+}(t) = W^+(t)$;
- For all $t \in T_{ab}$, $\Omega^*(t) = \Omega(t) + run$, $W^{*-}(t) = W^-(t) + run$ and $W^{*+}(t) = W^+(t) + run$;
- For all $t \in T_\tau$, $W^{*-}(t) = W^-(t) + run$.

Let S_f^* be S_f where all markings are increased by run .

Then it is routine to check that: $\mathcal{L}_C(\mathcal{N}^*, s[r, m_0^*], S_f^*) = \mathcal{L}_C(\mathcal{N}, s[r, m_0], S_f)$.

Furthermore (1) the empty tree is not reachable in $(\mathcal{N}^*, s[r, m_0^*])$ and (2) for any coverability sequence $s[r, m_0^*] \xrightarrow{\sigma} s \succeq s_f \in S_f^*$, r does not belong to the image of the corresponding mapping f . Thus in the rest of the proof we assume that $(\mathcal{N}, s[r, m_0], S_f)$ fulfills these properties. We also assume w.l.o.g. that all vertices in S_f are distinct. We denote V_f this set of vertices.

Let \mathcal{N}' obtained as follows.

One adds places $todo, done, cut, \{p_v \mid v \in V_f\}, \{p_{u,v} \mid s \in S_f, u \xrightarrow{m_v}_s v\}$.

- For all $t \in T_{el}$, $W'^-(t) = W^-(t)$ and $W'^+(t) = W^+(t)$;
- For all $t \in T_{ab}$, $W'^-(t) = W^-(t)$, $W'^+(t) = W^+(t)$ and $\Omega'(t) = \Omega(t) + cut$;
- For all $t \in T_\tau$, $W'^-(t) = W^-(t) + cut$.

For all $t \in T_{ab}$, one adds the following abstract transitions:

- one adds $t_{Br} \in T'_{ab}$ with $\lambda'(t_{Br}) = \lambda(t)$ and $W'^-(t_{Br}) = W^-(t) + todo$, $W'^+(t_{Br}) = done$, $\Omega'(t_{Br}) = \Omega(t) + todo$;
- For all r_s with $s \in S_f$ one adds $t_{r_s} \in T'_{ab}$ with $\lambda'(t_{r_s}) = \lambda(t)$ and $W'^-(t_{r_s}) = W^-(t) + todo$, $W'^+(t_{r_s}) = done$, $\Omega'(t_{r_s}) = \Omega(t) + (|\{r_s \xrightarrow{m_w}_s w\}| + 1)p_{r_s}$;
- For all $v \in V_s \setminus \{r_s\}$ with $s \in S_f$ and $u \xrightarrow{m_v}_s v$ such that $W^+(t) \geq m_v$, one adds $t_v \in T'_{ab}$ with $\lambda'(t_v) = \lambda(t)$ and $W'^-(t_v) = W^-(t) + p_u$, $W'^+(t_v) = p_{u,v}$, $\Omega'(t_v) = \Omega(t) + (|\{v \xrightarrow{m_w}_s w\}| + 1)p_v$.

One adds the following cut transitions:

- One adds $\tau_{done} \in T_\tau$ with $W'^-(\tau_{done}) = done$ and $\lambda'(\tau_{done}) = \varepsilon$.
- For all $v \in V_s$ with $s \in S_f$ one adds $\tau_v \in T'_\tau$ with $\lambda'(\tau_v) = \varepsilon$ and $W'^-(\tau_v) = M_s(v) + p_v + \sum_v \xrightarrow{m_w}_s w p_{v,w}$.

Let us prove that $\mathcal{L}_R(\mathcal{N}', s[r, m_0 + todo], \{\emptyset\}) = \mathcal{L}_C(\mathcal{N}, s[r, m_0], S_f)$.

• $\mathcal{L}_C(\mathcal{N}, s[r, m_0], S_f) \subseteq \mathcal{L}_R(\mathcal{N}', s[r, m_0 + todo], \{\emptyset\})$. Consider in \mathcal{N} a coverability sequence $s[r, m_0] \xrightarrow{\sigma} s \succeq s_f \in S_f$ with f the mapping from V_{s_f} to V_s . Let Br be the branch in s from r to $f(r_{s_f})$, excluding $f(r_{s_f})$. We build a sequence σ' as follows.

- Let $v \in Br \setminus \{r\}$ and (u, t) be the firing in σ that creates v .
Then we substitute (u, t) by (u, t_{Br}) ;
- Let (u, t) be the firing in σ that creates $f(r_{s_f})$. Then we substitute (u, t) by $(u, t_{r_{s_f}})$;
- Let $v \in V_{s_f} \setminus \{r_{s_f}\}$ and (u, t) be the firing in σ that creates $f(v)$.
Then we substitute (u, t) by $(u, t_{r_{s_f}})$.

Then σ' is a firing sequence of $(\mathcal{N}', s[r, m_0 + todo])$ that leads to s' with the same tree structure (and vertices) as the one of s and where the markings labelling s' are defined as follows.

- For all $v \in V_{s'} \setminus (Br \cup f(V_{s_f}))$, $M_{s'}(v) = M_s(v) + cut$,
and all $u \xrightarrow{m'_v}_{s'} v$ and $u \xrightarrow{m_v}_s v$, one has $m'_v = m_v$;
- For all $v \in Br$, $M_{s'}(v) = M_s(v)$. For all $v \xrightarrow{m'_w}_{s'} w$ with $w \in Br \cup \{f(r_{s_f})\}$,
 $m'_w = done$;
- For all $v \in V_{s_f}$, $M_{s'}(f(v)) = M_s(f(v)) + p_v$. For all $f(v) \xrightarrow{m'_w}_{s'} f(w)$,
 $m'_w = p_{v,w}$.

Observe that $\lambda(\sigma') = \lambda(\sigma)$. Then one completes σ' by firing $\{(f(v), \tau_v)\}_{v \in V_{s_f}}$ bottom up followed by firing $\{(v, \tau_{done})\}_{v \in Br}$ bottom up leading to \emptyset .

• $\mathcal{L}_R(\mathcal{N}', s[r, m_0 + todo], \{\emptyset\}) \subseteq \mathcal{L}_C(\mathcal{N}, s[r, m_0], S_f)$. Observe that in $(\mathcal{N}', s[r, m_0 + todo])$ the only way to reach \emptyset is to fire τ_{done} since in r (by induction) only abstract transitions of T_{ab} , $\{t_{Br} \mid t \in T_{ab}\}$ and $\{t_{r_s} \mid t \in T_{ab} \wedge s \in S_f\}$ are fireable and places cut and $\{p_v\}_{v \in V_f}$ are initially unmarked. Furthermore a single firing $\{t_{Br} \mid t \in T_{ab}\}$ and $\{t_{r_s} \mid t \in T_{ab} \wedge s \in S_f\}$ is at most possible in r since no transition can produce tokens for $todo$ in r .

So consider in \mathcal{N}' a firing sequence $s[r, m_0 + todo] \xrightarrow{\sigma'} \emptyset$. Due to the previous observation before the firing (r, τ_{done}) ending σ' , there has been in σ' a firing of (r, t_{Br}) or (r, t_{r_s}) for some $t \in T_{ab}$ and $s \in S_f$ creating a vertex v_1 followed by the firing of a cut transition in v_1 . Since $\Omega'(t_{Br}) = \Omega(t) + todo$, if v_1 has been created by (r, t_{Br}) then the only cut transition that can be fired in v_1 is τ_{done} . Since $\lambda'(\tau_{done}) = \varepsilon$ and $W'^+(t_{Br}) = done$, this firing can be delayed in σ' just before the firing of (r, τ_{done}) .

Furthermore there must have been before this firing, the firing of (v_1, t_{Br}) or (v_2, t_{r_s}) for some $t \in T_{ab}$ and $s \in S_f$ creating a vertex v_2 followed by the firing of a cut transition in v_2 . Since this iterated reasoning must end, there must be some v_k created by the firing of (v_{k-1}, t_{r_s}) (with $v_0 = r$) for some $t \in T_{ab}$ and $s \in S_f$. We denote by $f(r_s)$ the vertex v_k .

Since $\Omega'(t_{r_s}) = \Omega(t) + (|\{r_s \xrightarrow{m_w}_s w\}| + 1)p_{r_s}$, the only cut transition that can be fired in $f(r_s)$ is τ_{r_s} . Since $\lambda'(\tau_{r_s}) = \varepsilon$ and $W'^+(t_{r_s}) = done$, this firing can be delayed in σ' just before the firing of (v_{k-1}, τ_{done}) . Furthermore the firing of this cut transition must have been preceded for all $r_s \xrightarrow{m_w}_s w$ by the firing of some abstract transition (v_k, t_w) creating a vertex denoted $f(w)$ followed by the firing of a cut transition in $f(w)$.

Applying the same reasoning for $f(w)$ as the one for $f(r_s)$, one gets that the only cut transition that can be fired in $f(w)$ is τ_w and that all the firings related to these w 's can be delayed before the firing ($f(r_s), \tau_r s$).

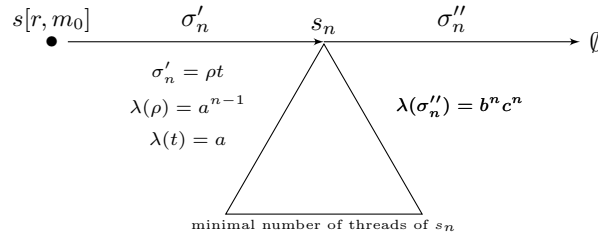
Iterating this process, one obtains that σ' can be reordered as $\sigma''\sigma_\tau$ with $\lambda'(\sigma'') = \lambda'(\sigma')$, and σ_τ is a sequence of cut transition firings with $\lambda(\sigma_\tau) = \varepsilon$.

Let s'' be the state of reached by σ'' : it includes a branch created by the firings among $\{t_{Br}\}_{t \in T_{ab}}$ followed by a tree whose set vertices is $f(V_s)$ and every vertex $f(v)$ has been created by the firing of some transition in $\{t_v\}_{t \in T_{ab}}$. Observe that due to our observations on $(\mathcal{N}', s[r, m_0 + todo])$ all other firings of σ'' are firings of transitions in T . By substituting in σ'' all t_{Br} by t and all t_v by t , one gets a firing sequence σ of $(\mathcal{N}, s[r, m_0])$ with $\lambda(\sigma) = \lambda'(\sigma')$ that covers s .

The transformation presented in the above proof can be performed in polynomial time and this will be used in the next section. The next proposition establishes that, as for Petri nets, coverability does not ensure the power of “exact counting”. The proof is interesting by itself since it combines an argument based on WSTS (case 1) and an argument *à la* Ogden (case 2).

Proposition 11. \mathcal{L}_3 is the reachability language of the Petri net of Figure 7 but it is not the coverability language of any RPN.

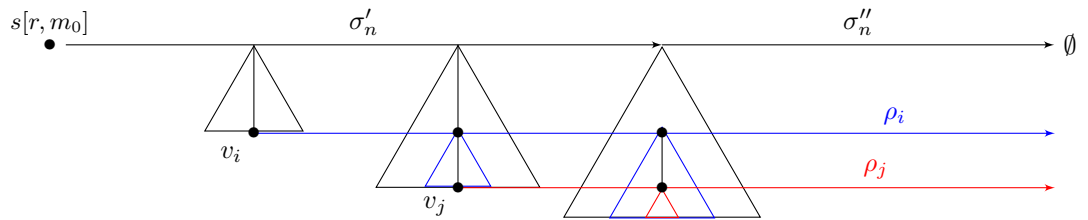
Proof. Due to Proposition 10, it is enough to prove that there does not exist $(\mathcal{N}, s[r, m_0])$ such that $\mathcal{L}_3 = \mathcal{L}_R(\mathcal{N}, s[r, m_0], \{\emptyset\})$. Assume by contradiction that there exists such $(\mathcal{N}, s[r, m_0])$. For all n , let σ_n be a firing sequence reaching \emptyset such that $\lambda(\sigma_n) = a^n b^n c^n$ and σ'_n be the prefix of σ_n whose last transition corresponds the last occurrence of a . Denote s_n the state reached by σ'_n and the decomposition by $\sigma_n = \sigma'_n \sigma''_n$. Among the possible σ_n , we select one such that s_n has a minimal number of threads. Let $Post$ be the finite set of \mathbb{N}^P defined by: $Post = \{W^+(t)\}_{t \in T_{ab}}$.



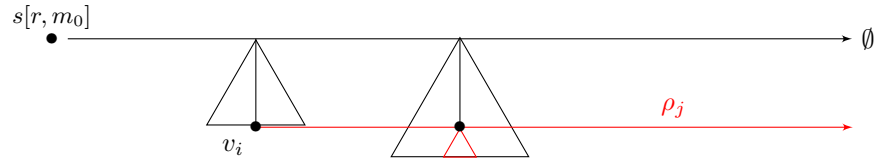
Case 1. There exists a bound B of the depths of the trees corresponding to $\{s_n\}_{n \in \mathbb{N}}$. Let S_B be the set of abstract states of depth at most B and different from \emptyset . Observe that S_0 can be identified to \mathbb{N}^P and S_B can be identified to $\mathbb{N}^P \times \text{Multiset}(Post \times S_{B-1})$. Furthermore the (component) order on \mathbb{N}^P and the equality on $Post$ are well quasi-orders. Since well quasi-order is preserved by the multiset operation and the cartesian product, S_B is well quasi-ordered by a quasi-order denoted $<$. By construction, $s \leq s'$ implies $s \preceq_r s'$. Thus there exist $n < n'$ such that $s_n \preceq_r s_{n'}$ which entails that $\sigma'_n \sigma''_{n'}$ is a firing sequence with trace $a^{n'} b^n c^n$ reaching \emptyset yielding a contradiction.

Case 2. The depths of the trees corresponding to $\{s_n\}_{n \in \mathbb{N}}$ are unbounded. There exists n such that the depth of s_n is greater than $(2|Post| + 1)$. Thus in s_n for $1 \leq j \leq 3$, there are edges $u_j \xrightarrow{m}_{s_n} v_j$ and denoting i_j the depth of v_j , one has $0 < i_1 < i_2 < i_3$.

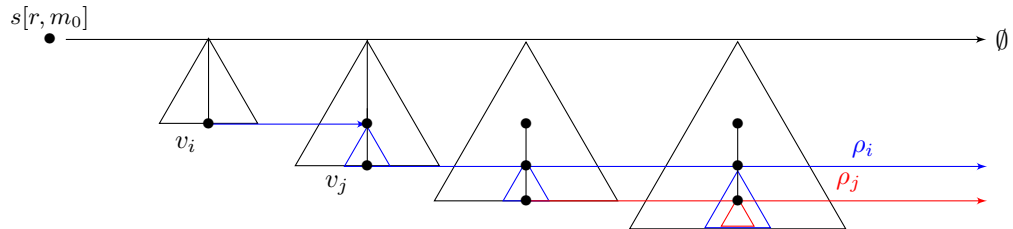
For $k \in \{1, 2, 3\}$, consider of the sequence ρ_k performed in the subtree rooted in v_k by the firings of σ_n . Among these three firing sequences two of them either (1) both finish by a cut transition in v_k or (2) both do not finish by a cut transition in v_k . Let us call i, j with $i < j$ the indices of these sequences and w_i and w_j their traces. We have illustrated the situation below.



One can build two firing sequences that still reach \emptyset and thus whose labels belong to the language. The first one consists of mimicking the “behavior” of the subtree rooted in v_j starting from v_i , which is possible due to the choice of i and j , as illustrated below.



The second one consists of mimicking the “behavior” of the subtree rooted in v_i starting from v_j as illustrated below.



Case $w_i = w_j$. Then the firing sequence reaching \emptyset obtained by mimicking in v_i the behaviour of v_j has trace $a^n b^n c^n$ and leads to another state s_n with

less threads yielding a contradiction, since s_n was supposed to have a minimal number of threads.

Case $w_i \neq w_j$. Let $w \neq \varepsilon$ be the trace of the sequence performed in the subtree rooted in v_i without the trace of the sequence performed in the subtree rooted in v_j . Let us consider the firing sequence σ reaching \emptyset obtained by mimicking in v_j the behaviour of v_i . The trace of σ is an interleaving of $a^n b^n c^n$ and w and it belongs to \mathcal{L}_3 which implies that $w = a^q b^q c^q$ for some $q > 0$. Furthermore σ can be chosen in such a way that the firing subsequences in the subtrees rooted at v_i and v_j are performed in one shot which implies that its trace is $\dots a^q a^q w_j b^q c^q b^q c^q \dots$ yielding a contradiction.

The following corollary shows that extending the family of coverability languages of PN's by substituting either (1) coverability by reachability or (2) PN's by RPN's is somewhat "orthogonal".

Corollary 3. *The families of reachability languages of Petri nets and the family of coverability languages of RPN's are incomparable.*

Proof. One direction is a consequence of Proposition 11 while the other direction is a consequence of Proposition 7 observing that the language of palindromes is not the reachability language of any Petri net.

The next corollary exhibits a particular feature of RPN's languages (e.g. Petri nets or context-free languages are closed under intersection with a regular language).

Corollary 4. *The family of coverability languages of RPN's is not closed under intersection with a regular language and under complementation.*

Proof. Due to Proposition 11, the family of coverability languages of RPN's is strictly included in the family of recursively enumerable languages. Since the former family is closed under homomorphism, Theorem 2 implies that it is not closed under intersection with a regular language and a fortiori with another coverability language. Since intersection can be obtained by union and complementation and since the family of RPN coverability languages is closed under union, they are not closed under complementation.

Combining Propositions 9, 10 and 11, one gets the following theorem.

Theorem 4. *The family of coverability languages of RPN's is strictly included in the family of reachability languages of RPN's.*

Figure 8 illustrates the hierarchy of the languages presented in this work.

5 Coverability is EXPSPACE-Complete

The section is devoted to establishing that the coverability problem is EXPSPACE-complete. The EXPSPACE-hardness follows immediately from the EXPSPACE-hardness of the coverability problem for Petri nets [37].

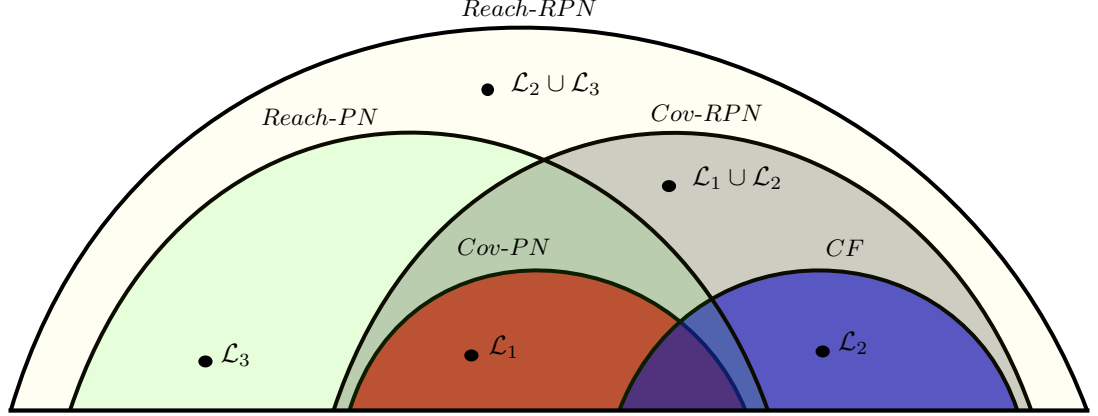


Fig. 8. $\mathcal{L}_1 = \{a^m b^n c^p \mid m \geq n \geq p\}$; $\mathcal{L}_2 = \{w \in \{d, e\}^* \mid w = \tilde{w}\}$; $\mathcal{L}_3 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$

Observe that the coverability problem is equivalent to the emptiness problem of the coverability language of an RPN. In Section 4 we have shown that the families of coverability languages and cut languages for RPN are equal and that the transformation from one to another is performed in polynomial time (proposition 9 and 10). Therefore we will establish the complexity result for the cut problem getting as a corollary the same result for the coverability problem.

Theorem 5. *The cut problem is EXPSPACE-complete.*

Proof. Let (\mathcal{N}, s_0) be a marked RPN and η the accumulated size of the RPN and the initial state. By Proposition 2 we can assume that V_{s_0} is a singleton $\{r\}$. Assume there exists a firing sequence $s_0 \xrightarrow{\sigma} \emptyset$. Using Proposition 4 one gets an omniscient sequence $s_0 \xrightarrow{\tilde{\sigma}} \emptyset$ such that $\tilde{\sigma} = (r, \sigma_1)(r, t)$ for some $t \in T_r$. The (omniscient) sequence (r, σ_1) contains only elementary transitions. Thus $m_0 \xrightarrow{\sigma_1}_{\mathcal{N}_{el}} m$ with $m \geq W^-(t)$. By Theorem 1, there exists σ'_1 with $|\sigma'_1| \leq 2^{2^{c\eta \log \eta}}$ covering $W^-(t)$. Using Corollary 1 there $s \xrightarrow{\sigma'} \emptyset$ with $|\sigma'| \leq 2^{2^{e\eta \log \eta}}$ for some constant e .

Therefore if there is a cut sequence then there is one with length at most $2^{2^{e\eta \log \eta}}$. Hence one guesses a sequence with at most this length and simultaneously checks whether it is a cut sequence in exponential space. This shows that the cut problem belongs to NEXPSPACE which is equivalent to EXPSPACE by Savitch's theorem.

The EXPSPACE hardness of the coverability problem in Petri nets entails EXPSPACE hardness of the coverability problem in RPNs which in turn entails the EXPSPACE hardness of the cut problem in RPNs.

The next theorem is an immediate corollary of the previous one.

Theorem 6. *The coverability problem for RPNs is EXPSPACE-complete.*

6 Termination is EXPSPACE-Complete

In this section we tackle the termination problem for RPN. Let (\mathcal{N}, s_0) be a marked RPN. We denote the size of the input of the termination problem by η . In [4] Rackoff showed that the termination problem for Petri net is solvable in exponential space:

Theorem 7 (Rackoff[37,4]). *The termination problem for Petri nets is EXPSPACE-complete.*

We aim to show that the termination problem for RPN is EXPSPACE-complete. EXPSPACE-hardness follows immediately from EXPSPACE-hardness of the termination problem for Petri nets [37]. By Proposition 2 we can assume that $V_{s_0} = \{r\}$. Hence for the rest of the section, we will assume that $s_0 = s[r, m_0]$ for some marking m_0 .

A main ingredient of the proof is the construction of an *abstract graph* related to the firing of abstract transitions.

Definition 10 (abstract graph). *Let (\mathcal{N}, s_0) be a marked RPN. Let $G_{\mathcal{N}, s_0} = (V_a, E_a, M_a)$ be a labeled directed graph defined inductively as follows:*

1. $r \in V_a$ and $M_a(r) = m_0$;
2. For any $v \in V_a$ and $t \in T_{ab}$, if there exists $s[v, M_a(v)] \xrightarrow{\sigma(v,t)}$ then $v_t \in V_a$, $(v, v_t) \in E_a$ and $M_a(v) = \Omega(t)$.

Observe that an edge (v, v_t) means that from state $s[v, M_a(v)]$, the thread v can fire t in the future and by induction that $v_t \in V_a$ if and only if t is fireable in the marked RPN. Observe that the size of $G_{\mathcal{N}, s_0}$ is linear w.r.t. the size of (\mathcal{N}, s_0) .

Lemma 5. *Let (\mathcal{N}, s_0) be a marked RPN. Then one can build its abstract graph in exponential space.*

Proof. First note that $|V_a| \leq |T_{ab}| + 1$. Then for any vertex v already in V_a and any $t \in T_{ab}$ checking whether $s[v, M_a(v)] \xrightarrow{\sigma(v,t)}$ is fireable is equivalent to solving the covering problem $M_a(v) \xrightarrow{\sigma} m \succeq W^-(t)$ in $\hat{\mathcal{N}}_{el}$ (recall Definition 9) which can be done in exponential space due to Rackoff's coverability theorem for Petri nets.

While we will not prove it, using a reduction from the Petri net coverability problem, one can show that we cannot use less than an exponential space to build the abstract graph.

Let us illustrate the abstract graph in Figure 9 corresponding to the RPN of Figure 1. Here the initial state is $s[r, p_{ini}]$. For clarity, we have renamed the abstract transitions as follows: $t := t_{beg}$, $ta := t_{a_2}$, $tb := t_{b_2}$. For instance, the existence of the edge from v_t to v_{ta} is justified by the firing sequence $(v_t, t_{a_1})(v_t, ta)$.

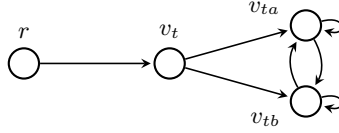


Fig. 9. An abstract graph for the RPN in Figure 1

Let σ be an infinite firing sequence. We say that σ is *deep* if it visits a state s whose depth is strictly greater than $|T_{ab}|$. Otherwise, we say that σ is *shallow*. To solve the termination problem it suffices to show whether the RPN has such an infinite sequence, either shallow or deep.

The next lemma establishes that *lassos* of the abstract graph are witnesses of deep infinite sequences in an RPN:

Lemma 6. *Let (\mathcal{N}, s_0) be a marked RPN. Then there is a deep infinite sequence starting from s_0 if and only if there is a cycle in $G_{\mathcal{N}, s_0}$.*

Proof. • Assume that σ is a deep sequence. Hence, it reaches a state \tilde{s} whose tree has a path γ starting from the root, with $|\gamma| > |T_{ab}|$. Let us denote it by $\gamma = (v_i)_{i=1}^m$. For all $i \leq m$ denote by t_i the abstract transition that creates v_i . Using γ , one builds a path $\gamma_a = v_1 v_2 \dots v_m$ in $G_{\mathcal{N}, s_0}$ as follows. First $v_1 = r$ and $m_r = M_a(r)$. Since along σ the thread r fires t_1 to create v_2 , there is an edge between r to v_{t_2} in $G_{\mathcal{N}, s_0}$. For any $1 < i \leq m$ the thread v_i is created with the marking $\Omega(t_i) = M_a(v_{t_i})$. Since v_{i+1} is a child of v_i , somewhere on the sequence σ the thread v_i fires t_{i+1} . Therefore there is an edge from v_{t_i} to $v_{t_{i+1}}$ in $G_{\mathcal{N}, s_0}$. The length of the path γ_a strictly greater than $|T_{ab}|$, and since $V_a \leq |T_{ab}| + 1$ there is a cycle in γ_a .

• Conversely assume that there is a cycle in $G_{\mathcal{N}, s_0}$. Then there is an infinite path $\gamma_a = \{v_i\}_{i=0}^\infty$ in $G_{\mathcal{N}, s}$ starting from r , where for any $i \geq 1$ denote by t_i the abstract transition associated the vertex v_i . We now translate this infinite path to an deep sequence on \mathcal{N} with initial state s_0 . Note that $v_0 = r$ and that $m_r = M_a(r)$. By definition of E_a there is a sequence $s \xrightarrow{\sigma_1} s'_0$ where the abstract transition t_1 is fireable from v_0 in s'_0 . We get $s \xrightarrow{\sigma_1} s'_0 \xrightarrow{(v_0, t_1)} s_2$. Denote by v_1 the thread created by t_1 . The threads marking has $M_{s_1}(v_1) = M_a(v_1)$, therefore

one continues translating the path γ_a in the same way as the first edge. Since for any (v_i, v_{i+1}) in γ_a we create a new thread from v_i one gets a deep sequence.

We now show that for any shallow σ there is a thread v which fires infinitely many times in σ .

Lemma 7. *Let (\mathcal{N}, s_0) be a marked RPN and σ be a shallow sequence. Then there is a thread v that fires infinitely many times in σ .*

Proof. If the root r fires infinitely often then we are done. Otherwise, r has finitely many children, and the firing subsequence of σ of the subtree of (at least) one child, say v , must be infinite. If v fires infinitely often then we are done. Otherwise, we proceed inductively up to $|T_{ab}|$ where some thread must fire infinitely often.

We now show that given some state $s[r, m_0]$ one can check in exponential space the existence of a shallow sequence in which r fires infinitely many times.

Lemma 8. *Let (\mathcal{N}, s_0) be a marked RPN. Then one can check in exponential space, whether there exists an infinite sequence starting with r firing infinitely many times.*

Proof. We first show that there is a sequence where r fires infinitely many times if and only if there is a infinite firing sequence in the marked Petri net $(\widehat{\mathcal{N}}_{el}, m_0)$.

- Assume there exists such σ in $(\mathcal{N}, s[r, m_0])$. Then the sequence σ is also fire-able in $(\widehat{\mathcal{N}}, s[r, m_0])$. In $\widehat{\mathcal{N}}$, one eliminates in σ the cut transitions by increasing occurrence order as follows. Let (v, t) be a cut transition and (v', t') be the firing that creates v . Then one deletes all the firings performed by the descendants of v and replaces (v', t') by (v', t'') . Let σ' be the sequence obtained after this transformation. In σ' , the root still fires infinitely often since no firing performed by the root has been deleted (but sometimes substituted by an elementary firing). Moreover, σ' has no more cut transitions. Consider the still infinite firing sequence (r, σ'') where in σ' all firings in other vertices than r have been deleted. Observe now that by definition, σ'' is also an infinite sequence of $\widehat{\mathcal{N}}_{el}$.

- Conversely, assume there exists an infinite firing sequence σ of $(\widehat{\mathcal{N}}_{el}, m_0)$. Then (r, σ) is an infinite firing sequence of $(\widehat{\mathcal{N}}, s[r, m_0])$ (with only root firings) entailing the existence of an infinite firing sequence of $(\mathcal{N}, s[r, m_0])$.

By Theorem 7, one can check in exponential space whether there exists an infinite sequence of $(\widehat{\mathcal{N}}_{el}, m_0)$.

Summing up the results for shallow and deep sequences we get:

Theorem 8. *The termination problem of RPN is EXPSPACE-complete.*

Proof. The algorithm proceeds as follows. It builds in EXPSPACE (by Lemma 5) the abstract graph and checks whether there is a deep infinite sequence using the characterization of Lemma 6. In the negative case, it looks for a shallow infinite sequence. To this aim, it checks in exponential space for any reachable vertex v from r in $G_{\mathcal{N}, s_0}$, whether there exists an infinite sequence starting from $s[v, M_a(v)]$ with the root firing infinitely many times. The complexity follows from Lemma 8 while the correctness follows from Lemma 7.

7 Finiteness and boundedness are EXPSPACE-complete

In this section we will show that the finiteness and boundedness problems for RPNs are EXPSPACE-complete w.r.t. $\eta = size(\mathcal{N}, s_0)$, i.e. the accumulated size of the RPN and the initial state. For Petri nets the finiteness problem, which is equivalent to the boundedness problem, has been shown to be EXPSPACE-complete:

Theorem 9 ([37,4]). *The finiteness problem for Petri nets is EXPSPACE-complete.*

EXPSPACE-hardness follows immediately from EXPSPACE-hardness of the finiteness problem for Petri nets [37].

Moreover by applying Proposition 2 like in previous sections we will assume that $s_0 = s[r, m_0]$. Given two vertices u, v in a graph \mathcal{G} , the *distance* between them $dist_{\mathcal{G}}(u, v)$ is the length of a shortest path going from one to the other.

Lemma 9. *Let (\mathcal{N}, s_0) be a marked RPN and $G_{\mathcal{N}, s_0} = (V_a, E_a, M_a)$ be its abstract graph. Then for all $v \in V_a$, there exists $s \in Reach(\mathcal{N}, s_0)$ and $u \in V_s$ such that $M_s(u) = M_a(v)$.*

Proof. We show the lemma by induction on $dist_{G_{\mathcal{N}, s_0}}(r, u)$. If $dist_{G_{\mathcal{N}, s_0}}(r, v) = 0$ then $v = r$ and $M_a(r) = m_0$. Assume that we have shown the lemma for any v such that $dist_{G_{\mathcal{N}, s_0}}(r, v) < n$, and pick $v \in V_a$ such that $dist_{G_{\mathcal{N}, s_0}}(r, v) = n$. Since $dist_{G_{\mathcal{N}, s_0}}(v, r) > 0$, $v = v_t$ for some $t \in T_{ab}$. Moreover there is some $(u, v_t) \in E_a$ such that $dist_{G_{\mathcal{N}, s_0}}(r, u) = n - 1$ and by the induction hypothesis there is a sequence $s_0 \xrightarrow{\sigma_u} s_u$ and some $w \in V_{s_u}$ such that $M_{s_u}(w) = M_a(u)$. From the definition of $G_{\mathcal{N}, s_0}$ there is a fireable sequence $s[w, M_a(u)] \xrightarrow{\sigma_t(w, t)}$. Combining these sequences, we get $s_0 \xrightarrow{\sigma_u} s_u \xrightarrow{\sigma_t(w, t)} s_{v_t}$, where the newly created thread w' fulfills $M_{s_{v_t}}(w') = \Omega(t) = M_a(v_t)$.

The following lemma shows that we can simulate the behaviour of every thread by a Petri net.

Lemma 10. *Let (\mathcal{N}, s_0) be a marked RPN and $G_{\mathcal{N}, s_0} = (V_a, E_a, M_a)$ be its abstract graph. Then:*

$$\bigcup_{s \in Reach(\mathcal{N}, s_0)} \{M_s(v)\}_{v \in V_s} = \bigcup_{u \in V_a} Reach(\widehat{\mathcal{N}}_{el}, M_a(u)).$$

Proof. • Let $m \in \bigcup_{s \in Reach(\mathcal{N}, s_0)} \{M_s(u)\}_{u \in V_s}$. There exists $s_0 \xrightarrow{\sigma} s$ with some $v \in V_s$ such that $M_s(v) = m$. By Proposition 4 there is an omniscient sequence in $s_0 \xrightarrow{\widehat{\sigma}}_{\widehat{\mathcal{N}}} s$. We split $\widehat{\sigma}$ into $s_0 \xrightarrow{\widehat{\sigma}_1}_{\widehat{\mathcal{N}}} s_v \xrightarrow{\widehat{\sigma}_2}_{\widehat{\mathcal{N}}}$ where s_v is the state where the thread v first appears. Note that there is $u \in V_a$ for which $M_{s_v}(v) = M_a(u)$. Let $(v, \widehat{\sigma}'_2)$ consisting of all firings of v in $\widehat{\sigma}_2$. $(v, \widehat{\sigma}'_2)$ is fireable from s_v since $\widehat{\sigma}_2$ is omniscient implying that there will be not cut transition fired by a child of v . By construction of $\widehat{\mathcal{N}}_{el}$, the sequence $\widehat{\sigma}'_2$ is a firing sequence of $(\widehat{\mathcal{N}}_{el}, M_a(u))$ thus $m \in Reach(\widehat{\mathcal{N}}_{el}, M_a(u))$.

• Let $u \in V_a$ and $m \in \text{Reach}(\widehat{\mathcal{N}}_{el}, M_a(u))$, i.e. $M_a(u) \xrightarrow{\sigma}_{\widehat{\mathcal{N}}_{el}} m$ for some $n \in \mathbb{N}$. First by Lemma 9 there exists $s_0 \xrightarrow{\sigma_u}_{\mathcal{N}} s_u$ where for some $v \in V_{s_u}$ we have $M_{s_u}(v) = M_a(u)$. By construction of $\widehat{\mathcal{N}}$ we also have $s_0 \xrightarrow{\sigma_u}_{\widehat{\mathcal{N}}} s_u$. By construction of $\widehat{\mathcal{N}}_{el}$ we get that $s_u \xrightarrow{(v, \sigma)}_{\widehat{\mathcal{N}}} s$ where $M_s(v) = m$. By Proposition 3, $s \in \text{Reach}(\mathcal{N}, s_0)$, which concludes the proof.

Using the previous Lemma and Rackoff's Theorem we establish the complexity of the boundedness problem:

Proposition 12. *The boundedness problem of RPN is EXPSPACE-complete.*

Proof. Hardness of the problem comes from hardness of Petri nets. Let (\mathcal{N}, s_0) be a marked RPN. First by Corollary 2 we can assume that $s_0 = s[r, m_0]$. By Lemma 10 checking whether \mathcal{N}, s_0 is bounded is equivalent to whether for $v \in V_a$, $(\widehat{\mathcal{N}}_{el}, M_a(u))$ is bounded which, due to Rackoff, can be performed in exponential space.

Let (\mathcal{N}, s_0) be a marked RPN. If $s_0 = \emptyset$ then the number of reachable states is finite (one), hence from now on we assume that $s_0 \neq \emptyset$. Next, if there exists $t \in T_{ab}$ with $W^-(t) = 0$ then there are infinitely many reachable states since one can fire t repeatedly which provides us with a sequence of states with an unbounded number of threads. Therefore from now on we assume that for all $t \in T_{ab}$, $W^-(t) > 0$.

We now establish a connection between the boundedness of $\widehat{\mathcal{N}}_{el}$ and the maximal number of children of the root in \mathcal{N} :

Lemma 11. *Let \mathcal{N} be an RPN such that $(\widehat{\mathcal{N}}_{el}, m_0)$ is bounded. Then:*

$$\sup_{s' \in \text{Reach}(\mathcal{N}, s[r, m_0])} |\{v \in V_{s'} \mid r_{s'} \rightarrow_{s'} v\}| < \infty$$

Proof. Assume that there exists a family of sequences $\{\sigma_n\}_{n \in \mathbb{N}}$ such that $s[r, m_0] \xrightarrow{\sigma_n}_{\mathcal{N}} s_n$ and the number of children of r in s_n is greater than n . By Proposition 4 for all σ_n there exists an omniscient sequence $\widehat{\sigma}_n$ in $\widehat{\mathcal{N}}$ from $s[r, m]$ reaching s_n . We remove from $\widehat{\sigma}_n$ all the transitions not fired from the root getting $(r, \widehat{\sigma}'_n)$ which is also fireable from $s[r, m]$ and which leads to a state where the root has a number of children greater than n . Since an abstract transition consumes tokens from the root (for all $t \in T_{ab}$, $W^-(t) > 0$) one can remove them from $(r, \widehat{\sigma}'_n)$ and get $(r, \widehat{\sigma}''_n)$ for which $s \xrightarrow{(r, \widehat{\sigma}''_n)}_{\widehat{\mathcal{N}}} s''_n$ and $\sum_{p \in P} M_{s''_n}(r)(p) > n$. Since $\widehat{\sigma}''_n$ is fireable from m in $\widehat{\mathcal{N}}_{el}$ this contradicts the hypothesis of the lemma.

Combining the above results, we get a characterization of the finiteness problem:

Proposition 13. *Let (\mathcal{N}, s_0) be a marked RPN. Then $\text{Reach}(\mathcal{N}, s_0)$ is finite if and only if both of the following assertions hold:*

1. *There is no loop in $G_{\mathcal{N}, s_0} = (V_a, E_a, M_a)$;*
2. *For all $v \in V_a$, $(\widehat{\mathcal{N}}_{el}, M_a(v))$ is bounded.*

Proof. • Assume that assertions 1 and 2 hold. Due to Assertion 1 and Lemma 6 any reachable state has its depth bounded by some constant. Due to Assertion 2 and Lemmas 10 and 11 each thread in any reachable state has a bounded number of children, and a bounded number of different reachable markings. Therefore $Reach(\mathcal{N}, s_0)$ is finite.

• Assume that Assertion 1 does not hold. By Lemma 6 there is a deep infinite sequence. Hence there is an infinite sequence of states with growing depth. Therefore $Reach(\mathcal{N}, s_0)$ is not finite.

• Assume that Assertion 2 does not hold for some vertex v . By Lemma 9 there exists a state $s \in Reach(\mathcal{N}, s_0)$ and a vertex $u \in V_s$ such that $M_s(u) = M_a(v)$. By the definition of $\widehat{\mathcal{N}}_{el}$, for any $m \in Reach(\widehat{\mathcal{N}}_{el}, M_s(v))$, there exists a firing sequence (r, σ') in $\widehat{\mathcal{N}}$ such that $s \xrightarrow{(r, \sigma')}_{\widehat{\mathcal{N}}} s'$ with $M_{s'}(v) = m$. Therefore $Reach(\widehat{\mathcal{N}}, s) \subseteq Reach(\widehat{\mathcal{N}}, s_0)$. Due to Proposition 3, $Reach(\widehat{\mathcal{N}}, s_0) = Reach(\mathcal{N}, s_0)$.

Theorem 10. *The finiteness problem of RPN is EXPSPACE-complete.*

Proof. The algorithm proceeds by checking Assertions 1 and 2 of Proposition 13. It builds in exponential space (by Lemma 5) the abstract graph and checks whether there is no loop in $G_{\mathcal{N}, s_0}$. In the negative case, it checks in exponential space for any vertex $v \in V_a$, whether the marked Petri net $(\widehat{\mathcal{N}}_{el}, M_a(v))$ is bounded.

8 Conclusion

We have proven that RPN is a strict generalisation of both Petri nets and context-free grammars without increasing the complexity of coverability, termination, boundedness and finiteness problems. It remains several open problems about languages of RPN and decidability/complexity of checking properties. Here is a partial list of open problems:

- How to decide whether a word belongs to a coverability or reachability language of a RP?
- Since the quasi-order possesses an infinite antichain, but there exist short witnesses for coverability, does there exist an effective finite representation of the downward closure of the reachability set?
- Does there exist a relevant fragment of LTL decidable for RPN?

References

1. Mayr EW. An Algorithm for the General Petri Net Reachability Problem. *SIAM J. Comput.*, 1984. **13**(3):441–460.
2. Czerwinski W, Lasota S, Lazic R, Leroux J, Mazowiecki F. The reachability problem for Petri nets is not elementary. In: Proceedings of STOC 19. 2019 pp. 24–33.

3. Leroux J, Schmitz S. Reachability in Vector Addition Systems is Primitive-Recursive in Fixed Dimension. In: Proceedings of LICS 19. 2019 pp. 1–13.
4. Rackoff C. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 1978. **6(2)**:223 – 231.
5. Reinhardt K. Reachability in Petri Nets with Inhibitor Arcs. *Electr. Notes Theor. Comput. Sci.*, 2008. **223**:239–264.
6. Bonnet R, Finkel A, Leroux J, Zeitoun M. Model Checking Vector Addition Systems with one zero-test. *LMCS*, 2012. **8(2)**:11).
7. Bonnet R. The Reachability Problem for Vector Addition System with One Zero-Test. In: MFCS 2011, Warsaw, Poland, volume 6907 of *LNCS*. 2011 pp. 145–157.
8. Schnoebelen Ph. Revisiting Ackermann-Hardness for Lossy Counter Machines and Reset Petri Nets. In: MFCS 2010, Brno, Czech Republic, volume 6281 of *LNCS*. 2010 pp. 616–628.
9. Dufourd C, Finkel A, Schnoebelen Ph. Reset Nets between Decidability and Undecidability. In: ICALP’98, volume 1443 of *LNCS*. Springer, Aalborg, Denmark, 1998 pp. 103–115.
10. Lazić R, Schmitz S. The Complexity of Coverability in ν -Petri Nets. In: LICS 2016. ACM Press, New York, United States, 2016 pp. 467–476.
11. Lazić R. The reachability problem for vector addition systems with a stack is not elementary. *CoRR*, 2013. **abs/1310.1767**. 1310.1767.
12. Lazić R, Schmitz S. Non-elementary complexities for branching VASS, MELL, and extensions. In: CSL-LICS 2014, Vienna, Austria. ACM, 2014 pp. 61:1–61:10.
13. Demri S, Jurdziński M, Lachish O, Lazić R. The covering and boundedness problems for branching vector addition systems. *Journal of Computer and System Sciences*, 2012. **79(1)**:23–38.
14. Atig MF, Ganty P. Approximating Petri Net Reachability Along Context-free Traces. In: FSTTCS 2011, Mumbai, India, volume 13 of *LIPICs*. 2011 pp. 152–163.
15. Mavrankulov G, Othman M, Turaev S, Selamat MH, Zhumabayeva L, Zhukabayeva T. Concurrently controlled grammars. *Kybernetika*, 2018. **54(4)**:748–764.
16. Dassow J, Turaev S. Petri Net Controlled Grammars: the Case of Special Petri Nets. *J. UCS*, 2009. **15(14)**:2808–2835.
17. Zetsche G. The Emptiness Problem for Valence Automata or: Another Decidable Extension of Petri Nets. In: RP 2015, Warsaw, Poland, volume 9328 of *LNCS*. 2015 pp. 166–178.
18. El Fallah Seghrouchni A, Haddad S. A Recursive Model for Distributed Planning. In: ICMAS 1996, Kyoto, Japan. 1996 pp. 307–314.
19. Haddad S, Poitrenaud D. Theoretical Aspects of Recursive Petri Nets. In: ICATPN 1999, Williamsburg, Virginia, USA, volume 1639 of *LNCS*. 1999 pp. 228–247.
20. Haddad S, Poitrenaud D. Modelling and Analyzing Systems with Recursive Petri Nets. In: WODES 2000, Ghent, Belgium, volume 569 of *The Springer International Series in Engineering and Computer Science*. 2000 pp. 449–458.
21. Haddad S, Poitrenaud D. Checking Linear Temporal Formulas on Sequential Recursive Petri Nets. In: TIME 2001, Cividale del Friuli, Italy. IEEE Computer Society, 2001 pp. 198–205.
22. Haddad S, Poitrenaud D. Recursive Petri nets. *Acta Inf.*, 2007. **44(7-8)**:463–508.
23. Finkel A, Haddad S, Khmelnitsky I. Coverability and Termination in Recursive Petri Nets. In: PETRI NETS’19, volume 11522 of *LNCSs*. Springer, Aachen, Germany, 2019 pp. 429–448.
24. Stadel M. A remark on the time complexity of the subtree problem. *Computing*, 1978. **19(4)**:297–302.

25. Finkel A, Schnoebelen P. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, 2001. **256**(1-2):63–92.
26. Czerwinski W, Lasota S, Lazic R, Leroux J, Mazowiecki F. The reachability problem for Petri nets is not elementary. In: STOC 2019. ACM, 2019 pp. 24–33.
27. Peterson JL. Petri net theory and the modeling of systems / James L. Peterson. Prentice-Hall Englewood Cliffs, N.J, 1981. ISBN 0136619835.
28. Geeraerts G, Raskin J, Begin LV. Well-structured languages. *Acta Informatica*, 2007. **44**(3-4):249–288.
29. Bonnet R, Finkel A, Haddad S, Rosa-Velardo F. Ordinal Theory for Expressiveness of Well-Structured Transition Systems. *Information and Computation*, 2013. **224**:1–22.
30. Delzanno G, Rosa-Velardo F. On the coverability and reachability languages of monotonic extensions of Petri nets. *Theor. Comput. Sci.*, 2013. **467**:12–29.
31. Valk R, Vidal-Naquet G. Petri Nets and Regular Languages. *J. Comput. Syst. Sci.*, 1981. **23**(3):299–325.
32. Figueira D. Co-finiteness of VASS coverability languages, 2019. Working paper or preprint, URL <https://hal.archives-ouvertes.fr/hal-02193089>.
33. Hofman P, Totzke P. Trace Inclusion for One-Counter Nets Revisited. In: RP 2014, volume 8762 of *LNCS*. Springer, 2014 pp. 151–162.
34. Haddad S, Poitrenaud D. Decidability and undecidability results for recursive Petri nets. Technical Report 019, LIP6, Paris VI University, 1999.
35. Ogden W. A helpful result for proving inherent ambiguity. *Mathematical systems theory*, 1968. **2**(3):191–194.
36. Lambert J. A Structure to Decide Reachability in Petri Nets. *Theor. Comput. Sci.*, 1992. **99**(1):79–104.
37. Lipton RJ. The Reachability Problem Requires Exponential Space. Technical Report 062, Yale University, Department of Computer Science, 1976.