



HAL
open science

Coverability and Termination in Recursive Petri Nets

Alain Finkel, Serge Haddad, Igor Khmelnitsky

► **To cite this version:**

Alain Finkel, Serge Haddad, Igor Khmelnitsky. Coverability and Termination in Recursive Petri Nets. Petri Nets 2019 / ACSD 2019 - 40th International Conference on Application and Theory of Petri Nets and Concurrency, Jun 2019, Aachen, Germany. hal-02081019

HAL Id: hal-02081019

<https://inria.hal.science/hal-02081019>

Submitted on 16 Apr 2019

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.

Coverability and Termination in Recursive Petri Nets

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

¹ LSV, ENS Paris-Saclay, CNRS, Université Paris-Saclay, France
² Inria, 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 stack automata, most of the usual property problems are solvable but using non primitive recursive algorithms, even for coverability and termination. 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 and termination 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 for free a more powerful model than Petri net.

Keywords: Recursive Petri nets · Expressiveness · Complexity · Coverability · Termination.

1 Introduction

Verification problems for Petri nets. Petri net is a useful formalism for analysis of concurrent programs for several reasons. From a modelling 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, all 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 are non primitive recursive (see for instance [20]) and it has been proved recently that the reachability problem is non elementary [4]. Fortunately some interesting properties like coverability, termination and boundedness are EXPSPACE-complete [21] and thus still manageable by a tool. So an important research direction consists

* 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)

of extending Petri nets to support new modelling 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 in adding inhibitor arcs which yields undecidability of all verification problems. However adding a single inhibitor arc preserves the decidability of the reachability, coverability, and boundedness problems [22,3,2]. When adding reset arcs, the coverability problem becomes Ackermann-complete and boundedness undecidable [23].

In ν -Petri nets the tokens are coloured where colours are picked in an infinite domain: their coverability problem is double Ackermann time complete [17]. In Petri nets equipped with a stack, the reachability and coverability problems are not only TOWER-hard [4,15] but their decidability status is still unknown. 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 [16] and its decidability is still an open problem while the coverability and the boundedness problems are 2-EXPTIME-complete [6]. In Petri nets with a stack, the reachability problem may be reduced to the coverability problem and both are at least not elementary while their decidability status is still unknown [15]. The analysis of subclasses of Petri nets with a stack is an active field of research [1,19,5,24]. However for none of the above extensions, the coverability and termination problems belong to EXPSPACE.

Recursive Petri nets (RPN). This formalism has been introduced in order to model distributed planning of multi-agent systems for which counters and recursivity were necessary for specifying resources and delegation of subtasks [7]. 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* or *abstract* 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 initialised with the *starting marking* of the transition. When a marking of a thread covers one of the *final markings*, it may perform a *cut* transition pruning its subtree and 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 [10,11] by reducing these properties to reachability problems of Petri nets. So the corresponding algorithms are non elementary. Model checking is undecidable for RPN but becomes decidable for the subclass of sequential RPN [12]. In [13], several modelling 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 [10]). 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. We show that the languages of RPN

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 coverability (as reachability) languages of RPN strictly include the union of context-free languages and Petri net coverability languages. On an other side, we prove that coverability languages of RPN and reachability languages of Petri nets are incomparable. In addition, we establish that the family of languages of RPN is closed by union, homomorphism but not by intersection with a regular language.

From an algorithmic point of view, we show that coverability and termination are EXPSPACE-complete, as for Petri nets. Thus the increasing of expressive power does not entail a corresponding increasing 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). The core of the proof consists in turning an arbitrary covering sequence into a structured one where all threads perform their firings in one shot. 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 is bounded or not along the sequence.

Outline. In section 2, we introduce RPNs and state ordering and establish basic results related to these notions. In section 3, we study the expressiveness of coverability languages. Then in sections 4 and 5, we show that the coverability and termination problems are EXPSPACE-complete. In section 6, we conclude and give some perspectives to this work. All missing proofs can be found in the appendix.

2 Recursive Petri Nets

2.1 Presentation

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 in a finite set of *final markings*. 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 $\mathcal{N} = \langle P, T, W^+, W^-, \Omega, \mathcal{F} \rangle$ where:

- P is a finite set of places;
- $T = T_{el} \uplus T_{ab}$ is a finite set of transitions with $P \cap T = \emptyset$, and T_{el} (respectively T_{ab}) is the subset of elementary (respectively abstract) transitions;

- W^- and W^+ are the $\mathbb{N}^{P \times T}$ backward and forward incidence matrices;
- $\Omega : T_{ab} \rightarrow \mathbb{N}^P$ is a function that labels every abstract transition with a starting marking;
- \mathcal{F} is a finite set of final markings.

Figure 1 graphically describes an example of an RPN with:

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

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 do the same for $W^-(t)$.

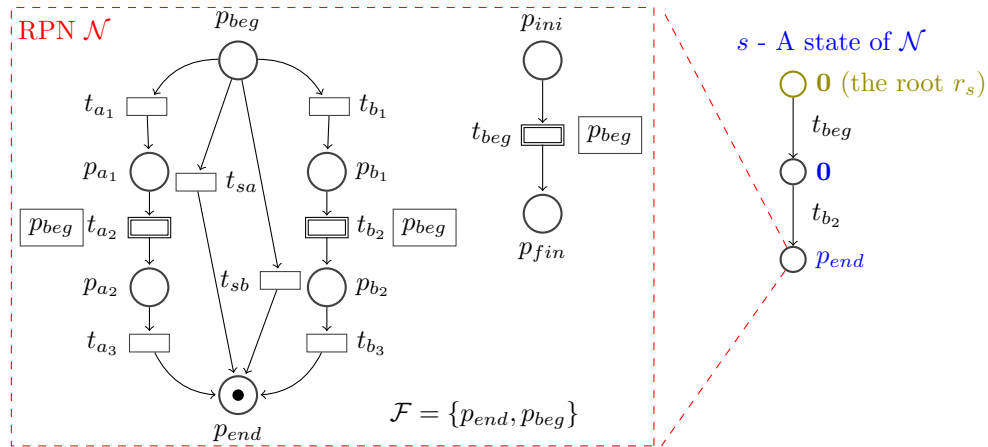


Fig. 1. An example of an RPN with the palindrome language on $\Sigma = \{a, b\}$ (see the proof of Proposition 2).

A 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.

Definition 2 (State of an RPN). A state $s = \langle V, M, E, \Lambda \rangle$ of an RPN $\mathcal{N} = \langle P, T, W^+, W^-, \Omega, \mathcal{F} \rangle$ is a 4-tuple where:

- V is a finite set of vertices;
- $M : V \rightarrow \mathbb{N}^P$ is a function that labels vertices with markings;
- $E \subseteq V \times V$ is a set of edges such that (V, E) is a Λ -labeled directed tree;

- $\Lambda : E \rightarrow T_{ab}$ is a function that labels edges with abstract transitions.

In the following, we denote by: $V_s := V$, $M_s := M$, $E_s := E$ and $\Lambda_s := \Lambda$.

For example, on the the right side of Figure 1 there is a state of the RPN \mathcal{N} . The 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 t_{beg} , t_{b2} . Since a state consists of a directed tree, it has a *root thread* denoted by r_s .

Let s be a state of some RPN. Since a state has the structure of a directed tree every thread u has a *predecessor*, denoted by $prd(u)$, except the root. We call the vertices v for which there exists $(u, v) \in E$ a *child* of u . The *descendants* of a thread u consists of threads in the sub-tree rooted in u including u itself. We denote this set by $Des_s(u)$. Similarly the *ancestors* of u are the threads for which u is a descendant of, i.e. $Asc_s(u) = \{v \mid u \in Des_s(v)\}$. Denote by \perp the empty tree. As usual two markings $m, m' \in \mathbb{N}^P$ over a set of places P are partially order as follows: $m \leq m'$ if for all places $p \in P$, $m(p) \leq m'(p)$.

The RPN moves from one state to another by one of the threads firing an elementary, abstract or cut transition. Let us present the first two kinds:

Definition 3 (operational semantics). *Let $\mathcal{N} = \langle P, T, W^+, W^-, \Omega, \mathcal{F} \rangle$ be an RPN, s be a state with some thread v and t be a transition (elementary or abstract). We say that $t \in T$ is fireable by v from $s = \langle V, M, E, \Lambda \rangle$ if $M(v) \geq W^-(t)$. In this case, its firing leads to the state $s' = \langle V', M', E', \Lambda' \rangle$, denoted $s \xrightarrow{(v,t)} s'$, where s' is defined below:*

- If $t \in T_{el}$ then $s' = \langle V, M', E, \Lambda \rangle$ where $M'(u) = M(u)$ for all $u \in V \setminus \{v\}$ and $M'(v) = M(v) - W^-(t) + W^+(t)$;
- If $t \in T_{ab}$ then:
 - $V' = V \cup \{w\}$ where w is a fresh identifier ($w \notin V$);
 - $M'(u) = M(u)$ for all $u \in V \setminus \{v\}$, $M'(v) = M(v) - W^-(t)$ and $M'(w) = \Omega(t)$;
 - $E' = E \cup \{(v, w)\}$;
 - $\Lambda'(e) = \Lambda(e)$ for all $e \in E$ and $\Lambda((v, w)) = t$.

Figure 2 illustrates the cases of an abstract and elementary transition firing. 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 t_{beg} . The marking of the root is decreased to 0 and v gets the initial marking $\Omega(t_{beg}) = p_{beg}$. The second firing is due to an elementary transition $t_{b1} \in T_{el}$ which is fired by v . Its firing results in a state for which the marking of v is changed to $M'_s(v) = M_s(v) + W^+(t_{b1}) - W^-(t_{b1}) = p_{b1}$.

We now introduce the last type of transition: *cut transition*. Given a state $s = \langle V, M, E, \Lambda \rangle$ and a thread $v \in V$, we denote by $s \setminus v$ the state $s' = \langle V', M', E', \Lambda' \rangle$, with:

- $V' = V \setminus Des_s(v)$;
- M' is the restriction of M on V' , and if $v \neq r$, $M'(prd(v)) = M(prd(v)) + W^+(\Lambda(prd(v), v))$;

- $E' = E \cap (V' \times V')$;
- Λ' is the restriction of Λ on E' .

Note that if v is the root of the tree then $s \setminus v = \perp$.

Definition 4 (τ cut transition). Let $\mathcal{N} = \langle P, T, W^+, W^-, \Omega, \mathcal{F} \rangle$ be an RPN, $s = \langle V, M, E, \Lambda \rangle$ be a state of \mathcal{N} and $v \in V$. We say that τ is fireable by v from s and reaches s' , denoted by $s \xrightarrow{(v, \tau)} s'$, if and only if there exists $m \in \mathcal{F}$ such that $M(v) \geq m$ and $s' = s \setminus v$.

For example, in Figure 2 the fifth transition to be fired is the cut transition τ , 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 its parent has its marking increased by $W^+(t_{b_2}) = p_{b_2}$.

A *firing sequence* is a sequence of transition firings, written in 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 $\sigma = (v_1, t_1)(v_2, t_2) \dots (v_n, t_n)$. Infinite firing sequences are similarly defined. 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 $i_1 < i_2 < \dots < i_k \leq n$ such that: $\sigma' = (v_{i_1}, t_{i_1})(v_{i_2}, t_{i_2}) \dots (v_{i_k}, t_{i_k})$.

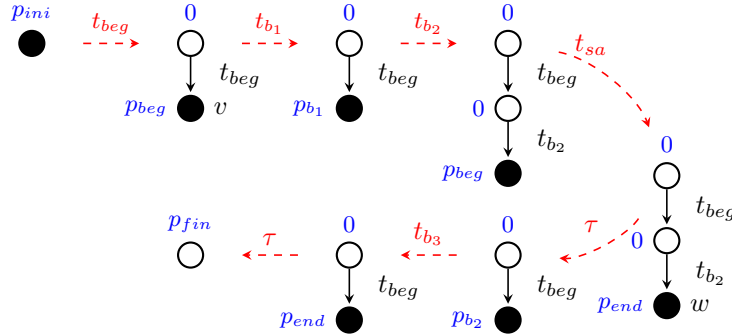


Fig. 2. Firing sequence for the RPN in Figure 1

Remark 1. In the rest of the paper, anywhere we write "RPN \mathcal{N} " we will mean $\mathcal{N} = \langle P, T, W^+, W^-, \Omega, \mathcal{F} \rangle$, unless we explicitly write differently.

2.2 An order for Recursive Petri Nets

We now define a quasi-order 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' if there is a subtree in s'

which is isomorphic to s , where markings are greater or equal on each vertex, and such that the labels on the edges fulfill $W^+(t) \geq W^+(t')$.

Definition 5. Given two states $s = \langle V, M, E, \Lambda \rangle$ and $s' = \langle V', M', E', \Lambda' \rangle$ of an RPN \mathcal{N} , we say that $s \preceq s'$ if and only if there exists an injective total function $f : V \rightarrow V'$ such that:

1. For any edge $(u, v) \in E$, we have $(f(u), f(v)) \in E'$;
2. For any edge $(u, v) \in E$, we have $W^+(\Lambda(u, v)) \leq W^+(\Lambda'(f(u), f(v)))$;
3. For any thread $v \in V$, we have $M(v) \leq M'(f(v))$.

Figure 3 illustrates this quasi-order. The state on the right is greater than the one on the left, if $W^+(t) \geq W^+(t')$.

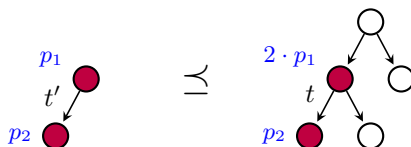


Fig. 3. Example of order between two states.

Lemma 1. The relation \preceq is a quasi-order.

This quasi-order is not a partial order since there could be abstract transitions $t \neq t'$ with $W^+(t) = W^+(t')$.

A quasi-order \preceq on the states of an RPN is *strongly compatible* (as in [8]) if for all states $s \leq s'$ and firing $s \xrightarrow{(v,t)} s_1$ there exist s'_1 and a firing $s' \xrightarrow{(v',t)} s'_1$ with $s_1 \leq s'_1$.

Lemma 2. The quasi-order \preceq is strongly compatible.

Note that, even though this quasi-order is compatible it 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) $s_i \not\preceq s_j$ since $|V_{s_j}| > |V_{s_i}|$ there cannot be any injective function from V_{s_j} to V_{s_i} , and (2) $s_j \not\preceq s_i$ since for any injective function from V_{s_i} to V_{s_j} at least one of the threads with the marking p would be mapped to a thread with marking $\mathbf{0}$.

Since RPNs are not well ordered they are not well-structured transition systems [8] for which coverability is often solved in EXPSPACE. Therefore to solve coverability, one needs to find another way.

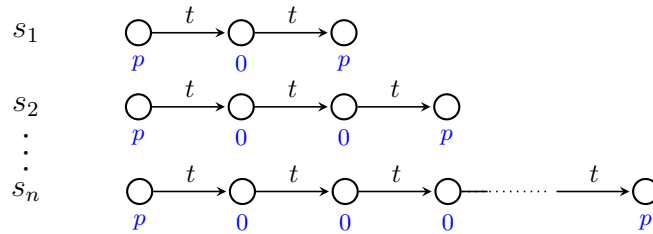


Fig. 4. An example for an antichain of states

3 Expressiveness

Expressiveness of a formalism may be defined by the family of languages that it can generate. In [10], expressiveness of RPNs was studied using reachability languages. In the next section, we are going to establish that the coverability problem for RPNs has a lower complexity than the one of the reachability problem for RPNs. Thus we want to study the coverability languages in order to determine whether this lower complexity has an impact on the expressiveness of the RPN formalism. 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 with the cut transitions labelled by the empty word. Thus given a labelled marked RPN \mathcal{N} and a finite subset of states S_f , the (coverability) language $\mathcal{L}(\mathcal{N}, S_f)$ is defined by:

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

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

As already announced, languages of RPNs are closed by union.

Proposition 1. *Coverability languages of RPNs are closed by union.*

The next theorem has two interesting consequences: the languages of RPNs are not closed by intersection with a regular language and this family is *quite close* to recursively enumerable languages.

Theorem 1. *Let \mathcal{L} be a recursively enumerable language. Then there exists an RPN language \mathcal{L}_1 , a regular language \mathcal{R}_2 and a homomorphism h such that $\mathcal{L} = h(\mathcal{L}_1 \cap \mathcal{R}_2)$.*

Proof. The result was stated in Proposition 9 of [9] for reachability languages but it also works for coverability languages since the reachability condition of the proof could easily be transformed into a coverability condition. \square

Obviously the coverability languages of RPNs include the one of PNs. In order to show its expressive power, let us introduce context-free grammars and languages. Let $G = (V, \Sigma, R, S)$ be a context-free grammar defined by V the non

terminal symbols including S , the start symbol and Σ the terminal symbols. The set of rules R is defined by $R = \{r_1, \dots, r_n\}$ such that $r_i = (v_i, u_i)$, with $v_i \xrightarrow{r_i} u_i$, $v_i \in V$ and $u_i \in (V \cup \Sigma)^*$ a word of length n_i . W.l.o.g. one assumes that the start symbol S does not occur in the right-hand side of any production rule of G and that $n_i > 0$ except possibly for a rule $S \rightarrow \varepsilon$. Given a word $\alpha v_i \beta \in (V \cup \Sigma)^*$, an application of rule r_i yields the word $\alpha u_i \beta$, denoted $\alpha v_i \beta \xrightarrow{r_i} \alpha u_i \beta$. A *derivation* from α to β is a consecutive application of a sequence of rules, and is denoted $\alpha \xrightarrow{\sigma} \beta$. The associated language $\mathcal{L}(G)$ is defined by:

$$\mathcal{L}(G) = \{w \in \Sigma^* \mid \exists \sigma S \xrightarrow{\sigma} w\}$$

Proposition 2. *Context-free languages are included in coverability languages of RPNs.*

The next lemma witnesses a Petri net language interesting from an expressiveness point of view.

Proposition 3. *Let $\Sigma = \{a, b, c\}$ and $\mathcal{L}_1 = \{a^m b^n c^p \mid m \geq n \geq p\}$. Then \mathcal{L}_1 is the coverability language of some Petri net and is not a context-free language.*

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

Theorem 2. *Coverability languages of RPNs strictly include the union of coverability languages of PNs and context-free languages.*

Proof. The inclusion is an immediate consequence of Proposition 2. Consider the language $\mathcal{L}_2 = \mathcal{L}_1 \cup \{w\tilde{w} \mid w \in \{d, e\}^*\}$ where \tilde{w} is the mirror of w .

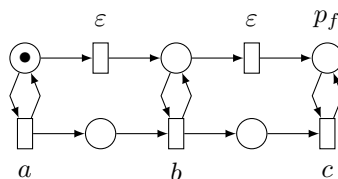
Since (1) by Proposition 1, coverability languages of RPNs are closed by union, (2) \mathcal{L}_1 is a PN language, and (3) the language of palindromes is a context-free language, we deduce that \mathcal{L}_2 is an RPN language.

PN and context-free languages are closed by homomorphism. Since the projection of \mathcal{L}_2 on $\{a, b, c\}$ is the language of Proposition 3, \mathcal{L}_2 is not a context-free language. Since the projection of \mathcal{L}_2 on $\{d, e\}$ is the language of palindromes, \mathcal{L}_2 is not a PN language (see [14]). \square

The next proposition establishes that, as for Petri nets, coverability does not ensure the power of “exact counting”.

Proposition 4. *Let $\Sigma = \{a, b, c\}$ and $\mathcal{L}_3 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$. Then \mathcal{L}_3 is the reachability language of some Petri net and is not the coverability language of any RPN.*

Proof. Consider the net below and p_f be the single marking to be reached. Then its language is \mathcal{L}_3 .



Assume that there exists a pair (\mathcal{N}, S_f) such that $\mathcal{L}_3 = \mathcal{L}(\mathcal{N}, S_f)$. Define the subset of abstract transitions T_ε such that for $t \in T_\varepsilon$ there exists a firing sequence labelled by ε starting from a single thread marked by $\Omega(t)$ that reaches the empty tree. Adding a set of elementary transitions $\{t_\varepsilon \mid t \in T_\varepsilon\}$ where t and t_ε have same incidence does not modify the language of the net. For all n , let σ_n be a coverability sequence 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.

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 states of depth at most B . Observe that $S_0 = \mathbb{N}^P$ and S_B can be identified to $\mathbb{N}^P \times \text{Multiset}(T_{ab} \times S_{B-1})$. Furthermore the (component) order on \mathbb{N}^P and the equality on T_{ab} are well quasi-orders. Since well quasi-ordering is preserved by the multiset operation and the cartesian product, S_B is well quasi-ordered by an quasi-order denoted $<$. By construction, $s \leq s'$ implies $s \preceq s'$. Thus there exist $n < n'$ such that $s_n \preceq s_{n'}$ which entails that $\sigma'_{n'} \sigma''_n$ is a covering sequence with trace $a^{n'} b^n c^n$ yielding a contradiction.

Case 2. The depths of the trees corresponding to s_n are unbounded. Let C be a strict upper bound of the depths of the initial state and the final states. There exists n such that the depth of s_n is greater than $(4|T_{ab}| + 1)C$. Thus in s_n , there are threads v_1, v_2 and v_3 and v_4 in the same branch at levels respectively $i_1 C, i_2 C, i_3 C$ and $i_4 C$ with $0 < i_1 < i_2 < i_3 < i_4$ created along σ_n by the firing of the same abstract transition t . Denote Tr the subtree of the final state of σ_n that matches the state to be covered and Br the branch leading to Tr in the final state. Due to the choice of C , there exists $1 \leq i \leq 4$ such that:

- Br_i , the branch from v_i to v_{i+1} , does not intersect Tr ;
- either Br_i does not intersect Br or Br_i is included in Br .

For $k \in \{i, i+1\}$, consider the trace w_k of the sequence performed in the subtree rooted in v_k by the firings of σ_n .

Case $w_i = w_{i+1}$. Then one can build another covering sequence with trace $a^n b^n c^n$ by mimicking in v_i the behaviour of v_{i+1} leading to another state s_n with less threads yielding a contradiction, since s_n was supposed to have a minimal number of threads.

Case $w_i \neq w_{i+1}$. Let $w \neq \varepsilon$ 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_{i+1} . Then one can build another covering sequence σ by mimicking in v_{i+1} the behaviour of v_i . The trace of σ is an interleaving of $a^n b^n c^n$ and w 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_{i+1} are performed in one shot which implies that its trace is $\dots a^q a^q w_{i+1} b^q c^q b^q c^q \dots$ yielding a contradiction. \square

The following corollary shows that extending coverability languages of Petri nets substituting (1) coverability by reachability or (2) Petri nets by RPNs are somewhat “orthogonal”.

Corollary 1. *The families of reachability languages of Petri nets and the coverability languages of RPNs are incomparable.*

Proof. One direction is a consequence of Proposition 4 while the other direction is a consequence of Proposition 2 observing that the language of palindromes is not the reachability language of any Petri net. \square

The next corollary exhibits a particular feature of RPNs languages (e.g. not fulfilled by Petri nets or context-free languages)

Corollary 2. *Coverability languages of RPNs are not closed by intersection with a regular language.*

Proof. Due to Proposition 4, coverability languages of RPNs are strictly included in recursively enumerable languages. Since it is closed by homomorphism, Theorem 1 implies that it is not closed by intersection with a regular language. \square

4 Coverability Is EXPSPACE-Complete

Let \mathcal{N} be an RPN and s_{ini}, s_{tar} be two states of \mathcal{N} . The *coverability problem* asks whether there exists a firing sequence $s_{ini} \xrightarrow{\sigma} s \succeq s_{tar}$. Such a sequence σ with initial and target states, is called a *covering sequence*. The section is devoted to establishing that this problem is EXPSPACE-complete. The EXPSPACE-hardness follows immediately from the EXPSPACE-hardness of the coverability problem for Petri nets [18].

In [21], 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 3 (Rackoff [21]). *Let \mathcal{N} be a Petri net, m_{ini}, m_{tar} be markings and σ be a firing sequence such that $m_{ini} \xrightarrow{\sigma} m \succeq m_{tar}$. Then there exists a sequence σ' such that $m_{ini} \xrightarrow{\sigma'} m' \succeq m_{tar}$ with $|\sigma'| \leq 2^{2^{cn \log n}}$ for some constant c and n being the size of (\mathcal{N}, m_{tar}) .*

So to solve the coverability problem on Petri nets, one guesses a sequence of length at most $2^{2^{cn \log n}}$, checking at the same time whether it is a covering sequence in exponential space. Which shows that the coverability problem belongs to NEXPSPACE = EXPSPACE by Savitch’s theorem.

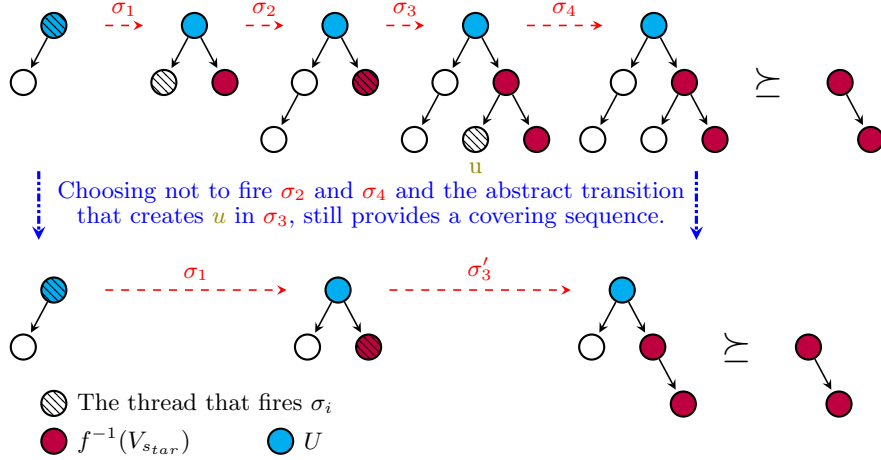
We follow a similar line and more specifically, we show that if there exists a covering sequence $s_{ini} \xrightarrow{\sigma} s \succeq s_{tar}$ in an RPN \mathcal{N} , then there exists a ‘short’ covering sequence σ' .

First, we establish that the final state of a covering sequence can be chosen with a limited number of threads (Proposition 5). Then we enlarge the RPN \mathcal{N} with new elementary transitions getting $\widehat{\mathcal{N}}$, leaving the coverability problem unchanged. The interest of $\widehat{\mathcal{N}}$ is that a covering sequence (when it exists) can be chosen with a particular form that we call *well-sequenced* without increasing its length (Proposition 6). In order to come back to \mathcal{N} , we establish that the firing of an additional transition of $\widehat{\mathcal{N}}$ can be simulated by a short sequence in \mathcal{N} (Proposition 7). Proposition 8 combines these intermediate results to get an upper bound for a short covering sequence.

Let σ be a firing sequence. A thread is *extremal* w.r.t. σ if it is an initial or final thread. We show that we can bound the number of extremal threads in a covering sequence. In the sequel, the size of the input of the coverability problem is denoted by η , i.e. the accumulated size of the RPN, the initial and target states. Recall that $Asc_s(v)$ is the set of ancestors of v in s .

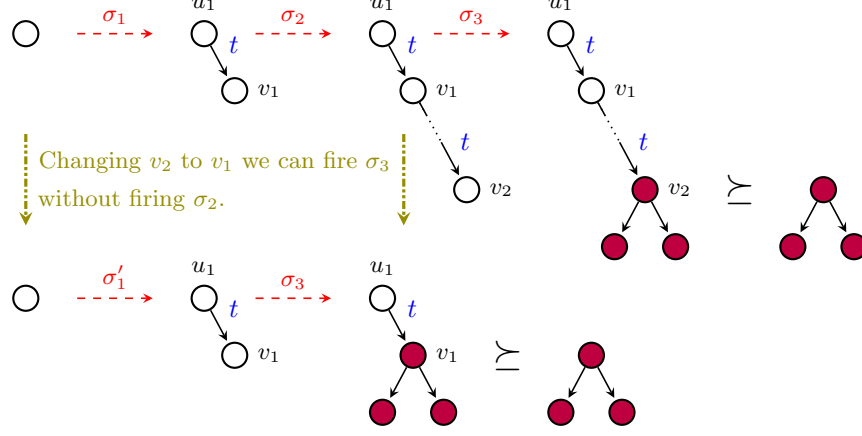
Proposition 5. *Let \mathcal{N} be an RPN and $s_{ini} \xrightarrow{\sigma} s \succeq s_{tar}$ be a covering sequence. Then there is a sequence $s_{ini} \xrightarrow{\sigma'} s' \succeq s_{tar}$ such that $|V_{s'}| \leq 3\eta$.*

Proof. If $s_{tar} = \perp$ then $\sigma' = \varepsilon$ is the appropriate sequence. Otherwise denote $f : V_{s_{tar}} \rightarrow V_s$ the injective mapping associated with $s_{tar} \preceq s$. Let $U = Asc_s(f(r_{s_{tar}})) \setminus \{f(r_{s_{tar}})\}$ be the branch in s leading to the vertex corresponding to the root of s_{tar} . Consider the set $V = V_s \setminus (U \cup f(V_{s_{tar}}))$. Then one can delete in σ all transitions fired from threads in $Des_\sigma(V)$ and those that created the threads of V and still get a covering sequence.



Now assume that on the branch U , two edges (u_1, v_1) and (u_2, v_2) are labelled by the same transition where u_2 is a descendent of u_1 and $v_1 \notin V_{s_{ini}}$. Then one can delete all transitions fired in the subbranch from v_1 to u_2 and substitute

transitions (v_2, t) by transitions (v_1, t) and still get a covering sequence. So $|U \setminus V_{s_{ini}}| \leq |T_{ab}|$.



Thus: $|V_s| \leq |V_{s_{ini}}| + |U \setminus V_{s_{ini}}| + |V_{s_{tar}}| \leq |V_{s_{ini}}| + |T_{ab}| + |V_s| \leq 3\eta$ \square

Let $T_{ret} \subseteq T_{ab}$, the set of *returning transitions* be defined by: $t \in T_{ret}$ if there exists a firing sequence (called a *return sequence*): $s_t \xrightarrow{\sigma_t} \perp$, where $V_{s_t} = \{v_t\}$, $M_{s_t}(v_t) := \Omega(t)$, and $E_{s_t} = A_{s_t} = \emptyset$. For any $t \in T_{ret}$, we define σ_t to be some arbitrary shortest return sequence.

As mentioned before, we get $\widehat{\mathcal{N}}$ from \mathcal{N} by adding elementary transitions as follows.

Definition 6 ($\widehat{\mathcal{N}}$). Let $\mathcal{N} = \langle P, T, W^+, W^-, \Omega, \mathcal{F} \rangle$ be an RPN. Then $\widehat{\mathcal{N}}$ is an RPN where $\widehat{\mathcal{N}} = \langle P, \widehat{T}, \widehat{W}^+, \widehat{W}^-, \Omega, \mathcal{F} \rangle$, where \widehat{T} is T with additional elementary transitions, and $\widehat{W}^+, \widehat{W}^-$ are updated accordingly: $\widehat{T}_{el} = T_{el} \uplus \{t^r : t \in T_{ret}\}$, and for any new transition t^r , $\widehat{W}^-(t^r) = W^-(t)$ and $\widehat{W}^+(t^r) = W^+(t)$.

The key ingredient of the existence of a short sequence is that in $\widehat{\mathcal{N}}$ every sequence can be turned into a *well-sequenced* sequence reaching the same state. Along such a sequence, (1) there are only extremal threads, (2) firings are performed in one shot by threads, and (3) only initial threads disappear and final threads perform firings of abstract transitions.

Definition 7. Let \mathcal{N} be an RPN and σ be a firing sequence. Then σ is well-sequenced if $\sigma = \sigma_1(v_1, \tau)\sigma_2(v_2, \tau) \dots \sigma_\ell(v_\ell, \tau)\sigma_{\ell+1}^{ab} \dots \sigma_k \sigma_k^{ab}$ where:

- The threads v_i are initial for $1 \leq i \leq \ell$;
- The threads v_i are final for $\ell + 1 \leq i \leq k$;
- The firing sequence $\sigma_i \in (\{v_i\} \times T_{el})^*$ for $1 \leq i \leq \ell$;
- The firing sequence $\sigma_i^{ab} \in (\{v_i\} \times T_{ab})^*$ for $\ell + 1 \leq i \leq k$.

Proposition 6. Let \mathcal{N} be an RPN and $s \xrightarrow{\sigma} s'$ be a firing sequence. There exists a well-sequenced firing sequence $s \xrightarrow{\widehat{\sigma}} s'$ in $\widehat{\mathcal{N}}$, with $|\widehat{\sigma}| \leq |\sigma|$.

Proof (sketch, full proof in the appendix). By construction, σ is fireable from s to s' in $\widehat{\mathcal{N}}$. Therefore all we are left with turning σ into a well-sequenced sequence.

Assume that an extremal thread u fires $t \in T_{ab}$ which creates a non-extremal thread v disappearing by the cut transition (v, τ) occurring in σ . For all such v 's let us (1) delete from σ the step (u, t) , and all the firings from $Des_\sigma(v)$ and (2) replacing the step (v, τ) by (u, t^r) . After this operation, no cut transition matches the firing of an abstract transition. Assume that an initial and not final thread u fires abstract transitions. Then one deletes these firings and all firings in the descendants of u . So along σ , there are only extremal threads.

Let us establish the other requirements on σ by induction on the number of extremal threads. There are three cases when adding a new thread with maximal depth:

1. This thread is an initial thread and a leaf in the final state. Then we can push the sequence of firings it performs to the end of the sequence.
2. The thread is final and not initial. Hence the subsequence of firings it performs in σ consists of elementary transition firings that can be fired at the end of σ . Furthermore, the abstract transition that created it can be fired at the end of the firing sequence of its parent.
3. This thread is initial and not final (i.e. in $\{v_1, \dots, v_\ell\}$). Hence the subsequence of firings it performs in σ consists of elementary transition firings possibly ended by a cut transition. If the cut transition occurs, the subsequence can be fired immediately. Otherwise, it can be omitted.

This concludes the proof. □

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 critical.

Proposition 7. *Let \mathcal{N} be an RPN and $t \in T_{ret}$. Then the returning sequence σ_t fulfills $|\sigma_t| \leq 2 \cdot 2^{dn \log n}$ 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 3 applies for a covering of some final marking.

Assume that the result holds for $k - 1$. 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 6 we get a sequence $\widehat{\sigma}_{t_k} \cdot (v_{t_k}, \tau)$ in $\widehat{\mathcal{N}}$. The sequence $\widehat{\sigma}_{t_k}$ consists of only elementary steps and does not contain any transition t_i^r with $i \geq k$. The marking reached by $\widehat{\sigma}_{t_k}$ covers some final marking, hence by Theorem 3 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 the sequence σ_{t_i} , one gets a corresponding sequence σ'_{t_k} in \mathcal{N} . Using the induction hypothesis, one gets that the length of this sequence:

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

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

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

□

Combining all previous results, we can now bound the length of a shortest covering sequence:

Proposition 8. *Let \mathcal{N} be an RPN, and $s_{ini} \xrightarrow{\sigma} s \succeq s_{tar}$. Then there exists a covering sequence of length shorter than $2^{2^{e\eta \log \eta}}$, where e is some constant and $\eta = \text{size}(\mathcal{N}, s_{ini}, s_{tar})$.*

Proof. Using Proposition 5 we can assume that $|V_{s_{ini}} \cup V_s| \leq 3\eta$. Using Proposition 6 one gets a well-sequenced sequence $s_{ini} \xrightarrow{\widehat{\sigma}} s$ in $\widehat{\mathcal{N}}$, such that:

$$\widehat{\sigma} = \sigma_1(v_1, \tau) \sigma_2(v_2, \tau) \dots \sigma_\ell(v_\ell, \tau) \sigma_{\ell+1} \sigma_{\ell+1}^{ab} \dots \sigma_k \sigma_k^{ab},$$

where $\sigma_i^{ab} = (v_i, t_{i,1}) \dots (v_i, t_{i,n_i})$. Observe that $k \leq |V_{s_{ini}} \cup V_s|$.

We now show that there is a short covering sequence in $\widehat{\mathcal{N}}$. Let $f : V_{s_{tar}} \rightarrow V_s$ the function associated with $s \succeq s_{tar}$. Each of the σ_i is a sequence whose final marking of v_i covers some marking:

1. For $i \leq \ell$, a final marking of the net;
2. For $i > \ell$ and $v_i \notin f(V_{s_{tar}})$, $\sum_j W^-(t_{i,j})$;
3. For $i > \ell$ and $v_i \in f(V_{s_{tar}})$, $\sum_j W^-(t_{i,j}) + M_{s_{tar}}(f^{-1}(v_i))$.

Since all σ_i contain only elementary steps, using Theorem 3, one gets as sequence σ'_i with $|\sigma'_i| \leq 2^{2^{e\eta \log \eta}}$ covering the marking specified by the three cases above.

Define the sequence $s \xrightarrow{\widehat{\sigma}'} s'$ where each σ_i is replaced by σ'_i . Using case 3, for all $v \in V_{s_{tar}}$ $M_{s'}(f(v)) \geq M_{s_{tar}}(v)$. Therefore $s' \succeq s_{tar}$, and the length of $\widehat{\sigma}'$ is at most:

$$|\widehat{\sigma}'| = \sum_{i=1}^k |\sigma'_i| + \sum_{i=1}^{\ell} |(v_i, \tau)| + \sum_{i=\ell+1}^k |\sigma_i^{ab}| \leq 3\eta 2^{2^{e\eta \log \eta}} + 3\eta + 3\eta \leq 2^{2^{2e\eta \log \eta}}.$$

Substituting any firing of t_i^r by σ_{t_i} in $\widehat{\sigma}'$ we get a covering sequence σ' in \mathcal{N} . Using Proposition 7, its length fulfills:

$$|\sigma'| \leq |\widehat{\sigma}'| \cdot 2^{2^{dn \log n}} \leq 2^{2^{en \log n}}$$

for some constant e . □

Using Proposition 8 we establish the complexity of the coverability problem.

Theorem 4. *The coverability problem for RPNs is EXPSPACE-complete.*

Proof. According to Proposition 8, if there is a covering sequence then there is one with length at most $2^{2^{e\eta \log \eta}}$ and no more than 4η threads. Hence one guesses a sequence of at most this length and checks simultaneously whether it is a covering sequence in exponential space. This shows that the coverability problem belongs to NEXPSPACE = EXPSPACE by Savitch theorem. □

5 Termination Is EXPSPACE-Complete

Let \mathcal{N} be an RPN and s_{ini} be an initial state of \mathcal{N} . The *termination problem* asks whether there exists an infinite firing sequence starting from s_{ini} . In [21] Rackoff showed that the termination problem on Petri net is solvable in exponential space:

Theorem 5 (Rackoff). *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 [18].

We first introduce and solve the *constrained termination problem* which asks whether there exists an infinite firing sequence starting from s_{ini} which does not delete any threads of s_{ini} . Accordingly, a *constrained firing sequence* is a firing sequence that does not delete any initial thread. The size of the input of the termination problem is denoted by η .

A main ingredient of the proof is a construction of an *abstract graph* related to the firing of abstract transitions. To this aim, let m be a marking then $s[m]$ is the state with a single thread whose marking is m .

Definition 8 (abstract graph). *Let \mathcal{N} be an RPN and s_{ini} be an initial state. Let $G_{\mathcal{N}, s_{ini}} = (V_a, E_a, M_a)$ be a labelled directed graph defined by:*

1. $V_a = \{v_t \mid t \in T_{ab}\} \uplus V_{s_{ini}}$;
2. $M_a : V_a \rightarrow \mathbb{N}^P$ where for all v in $V_{s_{ini}}$, $M_a(v) = M_{s_{ini}}(v)$ and for all t in T_{ab} , $M_a(v_t) := \Omega(t)$;
3. $E_a \subset V_a \times V_a$ such that for all $t \in T_{ab}$ and $v \in V_a$, $(v, v_t) \in E_a$ if there exists a firing sequence σ from the state $s[M_a(v)]$ ending by a firing (v, t) .

Equivalently assertion 3 means that the edge (v, v_t) belongs to E_a if there exists a covering sequence $s[M_a(v)] \xrightarrow{\sigma} s \succeq s[W^-(t)]$. Thus building the abstract graph amounts to solving a quadratic number of coverability problems. Using Theorem 4, one can build it in EXPSPACE.

Let us illustrate the abstract graph in Figure 5 corresponding to the RPN of Figure 1. Here the initial state is $s[p_{ini}]$. 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 $(v_t, t_{a_1})(v_t, ta)$.

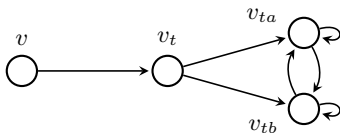


Fig. 5. An abstract graph for the RPN in Figure 1

Let σ be an infinite firing sequence. We say that σ is *deep* if it reaches a state s whose depth is greater than $|T_{ab}| + |V_{s_{ini}}|$. Otherwise, we say that σ is *shallow*. To solve the constrained 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 constrained infinite sequences in an RPN:

Lemma 3. *Let \mathcal{N} be an RPN and s_{ini} be an initial state. Then there is a deep constrained infinite sequence starting from s_{ini} if and only if there is a path from $u \in V_{s_{ini}}$ to a cycle in $G_{\mathcal{N}, s_{ini}}$.*

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

Lemma 4. *Given \mathcal{N} an RPN with an initial state s , and σ a shallow sequence. Then there is a thread v that fires infinitely many times in σ .*

Next, show that given a state consisting of one thread, one can check in EXPSPACE the existence of a shallow sequence which fires infinitely many times from this (root) thread.

Lemma 5. *Let \mathcal{N} be an RPN and m be a marking. Then one can check whether there exists an infinite sequence starting from $s[m]$ with the root firing infinitely many times.*

Proof. There is a sequence firing infinitely many times from the root in \mathcal{N} starting from $s[m]$ if and only if there is one in $\hat{\mathcal{N}}$ starting from $s[m]$. Define $\hat{\mathcal{N}}^e$ to be the Petri net with the same places as $\hat{\mathcal{N}}$, and whose set of transitions is the union of the elementary transitions of $\hat{\mathcal{N}}$ and the set $\{t^- \mid t \in T_{ab}\}$ where

$W^-(t^-) = W^-(t)$ and $W^+(t^-) = 0$. We claim that there is a sequence firing infinitely many times from the root in $\widehat{\mathcal{N}}$ starting from $s[m]$ if and only if there is an infinite sequence in $\widehat{\mathcal{N}}^e$ starting from m .

For one direction, assume there exists such σ in $\widehat{\mathcal{N}}$. One eliminates in σ the cut steps by increasing occurrence order as follows. Let (v_i, τ) be a cut step and (v_j, t_j) be the step that creates v_i . Then one deletes all the steps performed by the descendants of v_i and replaces (v_j, t_j) by (v_j, t_j^-) . 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 steps. One eliminates in σ' the abstract firings by increasing occurrence order as follows. Let (r, t_i) be an abstract firing that creates thread v . Then one deletes all the steps performed by the descendants of v and replaces (r, t_i) by (r, t_i^-) . 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 only elementary steps. So it is an infinite sequence of $\widehat{\mathcal{N}}^e$.

The other direction is immediate. By Theorem 5, one can check in EXPSPACE whether there exists an infinite sequence on $\widehat{\mathcal{N}}^e$ with initial marking m . \square

Summing the results for shallow and deep sequences we get:

Proposition 9. *The constrained termination problem of RPN belongs to EXPSPACE.*

Proof. The algorithm proceeds as follows. It builds in EXPSPACE the abstract graph and it checks whether there is a constrained deep infinite sequence using the characterisation of Lemma 3. In the negative case, it looks for a constrained shallow infinite sequence. To this aim, it checks in EXPSPACE for any reachable vertex v from $V_{s_{ini}}$ in $G_{\mathcal{N}, s_{ini}}$, whether there exists an infinite sequence starting from $s[M_a(v)]$ with the root firing infinitely many times. The complexity follows from Lemma 5 while its correctness follows from Lemma 4. \square

We now prove that the termination problem is EXPSPACE by reducing it in EXPSPACE to an exponential number of instances of the constrained termination problem with similar size:

Theorem 6. *The termination problem of RPN is EXPSPACE-complete.*

Proof. W.l.o.g. we assume that \mathcal{N} has been enlarged to $\widehat{\mathcal{N}}$ since the termination problem remains unchanged by this transformation. Let σ be an infinite firing sequence and U be the subset of initial threads that disappear along σ . Let σ' be the shortest finite prefix of σ such that all threads of U have disappeared. Using Proposition 6, σ' can be assumed to be well-sequenced. Consider σ_U , the prefix of σ' which deletes the threads of U and s_U the state reached by σ_U . State s_U is defined as follows: $V_{s_U} = V_s \setminus U$, E_{s_U} (respectively A_{s_U}) is the restriction of E_s (respectively A_s) to V_{s_U} and $M_{s_U}(v) = M_{s_{ini}}(v) + \sum_{(v,u) \in E_{s_{ini}} \wedge u \in U} W^+(A(v, u))$.

Using the same proof as in Proposition 8 the length of some σ_U (when it exists) is at most doubly exponential.

An infinite firing sequence starting from s exists if and only if there exists an infinite constrained sequence from one of its reachable s_U 's'. For any $U \subseteq V$ one checks whether s_U is reachable from s , and, in the positive case, solves the constrained termination problem for s_U . This can be done in EXPSPACE. Finally, there are only 2^n possible subsets $U \subseteq V_s$, repeating the process described above for every subset U solves the termination problem of RPN in EXPSPACE. \square

6 Conclusion

We have proven that RPN is a strict generalisation of both Petri nets and stack automata without increasing the complexity of coverability and termination problems. It remains several open problems about languages of RPN and decidability/complexity of checking properties. Here is a partial list of open problems: Is the family of covering languages of RPN included in the family of reachability languages of RPN? How to decide whether a word belongs to a coverability or reachability language of a RPN? What is the complexity of the boundedness and finiteness problems? Since the ordering 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?

References

1. Mohamed Faouzi Atig and Pierre Ganty. Approximating Petri net reachability along context-free traces. In *FSTTCS 2011, Mumbai, India*, volume 13 of *LIPICs*, pages 152–163, 2011.
2. Rémi Bonnet. The reachability problem for vector addition system with one zero-test. In *MFCS 2011, Warsaw, Poland*, volume 6907 of *LNCS*, pages 145–157, 2011.
3. Rémi Bonnet, Alain Finkel, Jérôme Leroux, and Marc Zeitoun. Model checking vector addition systems with one zero-test. *Logical Methods in Computer Science*, 8(2:11), 2012.
4. Wojciech Czerwinski, Sławomir Lasota, Ranko Lazic, Jérôme Leroux, and Filip Mazowiecki. The reachability problem for Petri nets is not elementary (extended abstract). *CoRR*, abs/1809.07115, 2018.
5. Jürgen Dassow and Sherzod Turaev. Petri net controlled grammars: the case of special Petri nets. *J. UCS*, 15(14):2808–2835, 2009.
6. Stéphane Demri, Marcin Jurdziński, Oded Lachish, and Ranko Lazic. The covering and boundedness problems for branching vector addition systems. *Journal of Computer and System Sciences*, 79(1):23–38, 2012.
7. Amal El Fallah Seghrouchni and Serge Haddad. A recursive model for distributed planning. In *ICMAS 1996, Kyoto, Japan*, pages 307–314, 1996.
8. Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, 256(1-2):63–92, 2001.

9. Serge Haddad and Denis Poitrenaud. Decidability and undecidability results for recursive Petri nets. Technical Report 019, LIP6, Paris VI University, 1999.
10. Serge Haddad and Denis Poitrenaud. Theoretical aspects of recursive Petri nets. In *ICATPN 1999, Williamsburg, Virginia, USA*, volume 1639 of *LNCS*, pages 228–247, 1999.
11. Serge Haddad and Denis Poitrenaud. 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*, pages 449–458, 2000.
12. Serge Haddad and Denis Poitrenaud. Checking linear temporal formulas on sequential recursive Petri nets. In *TIME 2001, Cividale del Friuli, Italy*, pages 198–205. IEEE Computer Society, 2001.
13. Serge Haddad and Denis Poitrenaud. Recursive Petri nets. *Acta Inf.*, 44(7-8):463–508, 2007.
14. Jean-Luc Lambert. A structure to decide reachability in Petri nets. *Theor. Comput. Sci.*, 99(1):79–104, 1992.
15. Ranko Lazic. The reachability problem for vector addition systems with a stack is not elementary. *CoRR*, abs/1310.1767, 2013.
16. Ranko Lazic and Sylvain Schmitz. Non-elementary complexities for branching vass, mell, and extensions. In *CSL-LICS 2014, Vienna, Austria*, pages 61:1–61:10. ACM, 2014.
17. Ranko Lazić and Sylvain Schmitz. The Complexity of Coverability in ν -Petri Nets. In *LICS 2016*, pages 467–476, New York, United States, 2016. ACM Press.
18. Richard J. Lipton. The reachability problem requires exponential space. Technical Report 062, Yale University, Department of Computer Science, January 1976.
19. Gairatzhan Mavlankulov, Mohamed Othman, Sherzod Turaev, Mohd Hasan Selamat, Laula Zhumabayeva, and Tamara Zhukabayeva. Concurrently controlled grammars. *Kybernetika*, 54(4):748–764, 2018.
20. Ernst W. Mayr. An algorithm for the general petri net reachability problem. *SIAM J. Comput.*, 13(3):441–460, 1984.
21. Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223 – 231, 1978.
22. Klaus Reinhardt. Reachability in Petri nets with inhibitor arcs. *Electr. Notes Theor. Comput. Sci.*, 223:239–264, 2008.
23. Philippe Schnoebelen. Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets. In *MFCS 2010, Brno, Czech Republic*, volume 6281 of *LNCS*, pages 616–628, 2010.
24. Georg Zetsche. The emptiness problem for valence automata or: Another decidable extension of Petri nets. In *RP 2015, Warsaw, Poland*, volume 9328 of *LNCS*, pages 166–178, 2015.

7 Appendix

7.1 Proofs of Section 2

Lemma 1. *The relation \preceq is a quasi-order.*

Proof. Let, s, s', s'' be states of an RPN \mathcal{N} with $s = \langle V, M, E, \Lambda \rangle$, $s' = \langle V', M', E', \Lambda' \rangle$ and $s'' = \langle V'', M'', E'', \Lambda'' \rangle$. Let us show that the relation \preceq is a quasi order.

1. Reflexivity: the identify function Id on V insures that $s \preceq s$.
2. Transitivity: Given $s \preceq s' \preceq s''$, there exist two functions $f : V \rightarrow V'$ and $f' : V' \rightarrow V''$. Let $g : V \rightarrow V''$ be the function $g = f' \circ f$. The function g is injective, and it preserves the structure of the tree in s . Any edge $(u, v) \in E$ satisfy:

$$\begin{aligned} W^+(\Lambda''(g(u), g(v))) &= W^+(\Lambda''(f'(f(u)), f'(f(v)))) \\ &\geq W^+(\Lambda'(f(u), f(v))) \\ &\geq W^+(\Lambda(u, v)). \end{aligned}$$

For any $v \in V$, we have $M(v) \leq M'(f(v)) \leq M''(f'(f(v))) = M''(g(v))$. Therefore $s \preceq s''$. □

Lemma 2. *The quasi-order \preceq is strongly compatible.*

Proof. Let us prove that for any two states $s \preceq q$ and an elementary step $s \xrightarrow{(v,t)}$ s' , the elementary step $(f(v), t)$ is also fireable from q and we have $q \xrightarrow{(f(v),t)}$ q' with $s' \preceq q'$. We consider three cases according to the type of the elementary transition t in (v, t) .

1. $t \in T_{el}$. First, $(f(v), t)$ is fireable from q since:

$$M_q(f(v)) \geq M_s(v) \geq W^-(t).$$

Now since the set of threads and the edges didn't change we let $f' = f$, and get the clauses 1 and 2 from the definition of \preceq hold. The only change in s' and q' is the marking of the vertex v and $f(v)$ respectively. Therefore its enough to show that $M_{q'}(f(v)) \geq M_{s'}(v)$. But this is true since:

$$M_{q'}(f'(v)) = M_q(f(v)) + W^+(t) - W^-(t) \geq M_s(v) + W^+(t) - W^-(t) = M_{s'}(v).$$

Therefore clause 3 also holds and hence we get that $s' \preceq q'$.

2. $t \in T_{ab}$. The step $(f(v), t)$ is fireable from q for the same reason as for the previous step. Since it was an abstract transition, there is a new vertex in

$V_{s'}$ and in $V_{q'}$, denote it by $u_{s'}$ and $u_{q'}$ respectively. Let $f' : V_{s'} \rightarrow V_{q'}$ be the function:

$$f'(v) := \begin{cases} f(v) & v \in V_s \\ u_{q'} & v = u_{s'} \end{cases}.$$

The changes in s' and q' are in the marking of the threads v and $f(v)$, the new threads $u_{s'}, u_{q'}$ and their respective edges $(v, u_{s'})$ and $(f(v), u_{q'})$. Hence it is enough to show that f' 'respects' these changes i.e. :

(a) For the marking of the thread v , we have:

$$M_{q'}(f'(v)) = M_q(f(v)) - W^-(t) \geq M_s(v) - W^-(t) = M_{s'}(v).$$

(b) For the marking of the thread $u_{s'}$ we have

$$M_{s'}(u_{s'}) = \Omega(t) = M_{q'}(u_{q'}) = M_{q'}(f(u_{s'})).$$

(c) For the new edge $(v, u_{s'}) \in E'_s$ we have that

$$(f'(v), f'(u_{s'})) = (f(v), u_{q'}) \in E_{q'}.$$

Where its label in q' is the same one as in s' since they both were created by the same abstract transition t .

Therefore clauses 1,2,3 from the definition of \preceq hold and hence we get that $s' \preceq q'$.

3. The elementary step (v, t) is a cutting step, i.e. $t = \tau$. Same as in the first two cases $(f(v), \tau)$ is fireable from q . Since t is a cutting step, the changes in s' and q' are the following: we remove all the threads in $Des_s(v)$ and $Des_q(f(v))$ respectively, and change the marking of the threads $pred_s(v)$ and $pred_q(f(v))$. Denote by $f' : V_{s'} \rightarrow V_{q'}$ the function $f'(v) := f(v)$. We first note that $pred_q(f(v)) = f(pred_s(v))$ because there is only one father for each thread and by clause 1 of the definition of \preceq we have that $(pred_s(v), v) \in E_s$ if and only if $(f(pred_s(v)), f(v)) \in E_q$. Since we only have less edges in s' and since we take a projection of f we get clauses 1 and 2 immediately. The marking on $pred_s(v)$ has:

$$\begin{aligned} M_{q'}(pred(f'(v))) &= M_q(pred(f(v))) + W^+(A_q(f(pred(v)), f(v))) \geq \\ &\geq M_s(pred(v)) + W^+(A_s(pred(v), v)) = M_{s'}(pred(v)). \end{aligned}$$

Therefore clause 3 also holds and hence we get that $s' \preceq q'$. □

7.2 Proofs of Section 3

The next lemma simplifies the proof that coverability languages of RPN are closed by union.

Lemma 6. *Let $\mathcal{L}(\mathcal{N}, S_f)$ be a RPN language. Then there exists a RPN \mathcal{N}' whose initial state consists of a single thread and a finite subset of states S'_f such that $\mathcal{L}(\mathcal{N}', S'_f) = \mathcal{L}(\mathcal{N}, S_f)$.*

Proof. Let $s_0 = \langle V, M, E, \Lambda \rangle$ be the initial state of \mathcal{N} . \mathcal{N}' is built from \mathcal{N} by adding a subset of places $\{p_e \mid e \in E\}$ and a subset of abstract transitions $\{t_e \mid e \in E\}$ all labelled by ε . Let $e = (u, v) \in E$ and denote $t = \Lambda(e)$. Then $W^-(t_e) = p_e$, $W^+(t_e) = W^+(t)$ and $\Omega(t_e) = M(v) + \sum_{e' \mid \exists w \ e'=(v,w)} p_{e'}$. S'_f is obtained from S_f by extending the markings occurring in the states of S_f to be null on the subset $\{p_e \mid e \in E\}$. Finally the single thread r of s'_0 fulfills $M'(r) = M(r) + \sum_{e' \mid \exists w \ e'=(r,w)} p_{e'}$.

Let us prove that $\mathcal{L}(\mathcal{N}, S_f) \subseteq \mathcal{L}(\mathcal{N}', S'_f)$. Consider a covering sequence σ for (\mathcal{N}, S_f) . One starts to fire all transitions of $\{t_e \mid e \in E\}$ in a top-down way to get a state s'_1 such that (when omitting places $\{p_e\}$) both $s_0 \preceq s'_1$ and $s'_1 \preceq s_0$ so that we identify the corresponding threads. Let us denote σ_0 this sequence. Then $\sigma_0\sigma$ is a covering sequence for (\mathcal{N}', S'_f) with $\lambda(\sigma_0\sigma) = \lambda(\sigma)$.

Let us prove that $\mathcal{L}(\mathcal{N}', S'_f) \subseteq \mathcal{L}(\mathcal{N}, S_f)$. Consider a covering sequence σ for (\mathcal{N}', S'_f) . By construction of \mathcal{N}' , all occurrences of transitions of $\{t_e \mid e \in E\}$ can be pushed at the beginning of the sequence leading to a covering sequence $\sigma' = \sigma_1\sigma_2$ where σ_1 consist of such occurrences. Then σ_1 can be completed by the missing occurrences still in a top-down way producing σ'_1 and yielding a covering sequence $\sigma'' = \sigma'_1\sigma_2$. Moreover the state s_1 (when omitting places $\{p_e\}$) reached from σ'_1 can be identified to s_0 . Thus σ_2 is a covering sequence for (\mathcal{N}, S_f) with $\lambda(\sigma_2) = \lambda(\sigma)$. \square

Proposition 1. *Coverability languages of RPNs are closed by union.*

Proof. Consider two RPNs with final states (\mathcal{N}, S_f) and (\mathcal{N}', S'_f) such that (w.l.o.g. due to Lemma 6) their initial state consists of a single thread. One denotes m_0 and m'_0 the marking of these threads. Let us define $\widehat{\mathcal{N}}$ 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 $t \in T$, $\widehat{W}^-(t) = W^-(t) + 1 \cdot p$ and $\widehat{W}^+(t) = W^+(t)$.
- For all $t \in T'$, $\widehat{W}^-(t) = W'^-(t) + 1 \cdot p'$ and $\widehat{W}^+(t) = W'^+(t)$.
- For all $t \in T_{ab}$, $\widehat{\Omega}(t) = \Omega(t) + 1 \cdot p$.
- For all $t \in T'_{ab}$, $\widehat{\Omega}(t) = \Omega'(t) + 1 \cdot p'$.
- $\widehat{W}^-(t_b) = \widehat{W}^-(t_b) = 1 \cdot p_0$, $\widehat{W}^+(t_b) = m + 1 \cdot p$, $\widehat{W}^+(t_b) = m' + 1 \cdot p'$
- $\widehat{W}^-(t_c) = 1 \cdot p$, $\widehat{W}^+(t_c) = 2 \cdot p$, $\widehat{W}^-(t'_c) = 1 \cdot p'$, $\widehat{W}^+(t'_c) = 2 \cdot p'$
- $\widehat{\mathcal{F}} = \{m + 1 \cdot p \mid m \in \mathcal{F}\} \cup \{m + 1 \cdot p \mid m \in \mathcal{F}'\}$
- \widehat{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 $t \in T$, $\widehat{\lambda}(t) = \lambda(t)$, for all $t \in T'$, $\widehat{\lambda}(t) = \lambda'(t)$.
- For all $t \in \{t_b, t_c, t'_b, t'_c\}$, $\widehat{\lambda}(t) = \varepsilon$.
- The initial state of $\widehat{\mathcal{N}}$ consist of a single thread whose marking is $1 \cdot p_0$.

Let us prove that $\mathcal{L}(\mathcal{N}, S_f) \cup \mathcal{L}(\mathcal{N}', S'_f) \subseteq \mathcal{L}(\widehat{\mathcal{N}}, \widehat{S}_f)$. Let σ be a covering sequence (\mathcal{N}, S_f) . The corresponding covering sequence $\widehat{\sigma}$ of $L(\widehat{\mathcal{N}}, \widehat{S}_f)$ is built as follows. Initially, one fires $(r, t_b)(r, t_c)^{\ell_r}$ where ℓ_r is the number of 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 firings occurring in σ triggered by r . It is routine to check that $\widehat{\sigma}$ is covering sequence. The proof for $\mathcal{L}(\mathcal{N}', S'_f)$ is similar.

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

A *leftmost derivation* always applies rules on the leftmost non terminal symbol of a word of $(V \cup \Sigma)^*$. Using only leftmost derivations let unchanged the language.

Proposition 2. *Context-free languages are included in coverability languages of RPNs.*

Proof. Let $G = (V, \Sigma, R, S)$ be a context-free grammar.

We define a labelled marked net \mathcal{N}_G as follows.

- The set of places is defined by:

$$P = \{b_v, e_v \mid v \in V\} \cup \{p_{i,j} \mid 1 \leq i \leq n \wedge 0 \leq j < n_i\}$$

- $\mathcal{F} = \{m_v\}_{v \in V \setminus \{S\}}$ where $m_v(e_v) = 1$ and for all $p \neq e_v$, $m_v(p) = 0$.
- The set of transitions is defined by $T = \bigcup_{i \leq n} T_i$ where all abstract transitions are labelled by ε and:
 - If $n_i = 0$ then $T_i = \{t_{i,0}\}$ with $W^-(t_{i,0}) = b_S$ and $W^+(t_{i,0}) = e_S$, $t_{i,0} \in T_{el}$ and $\lambda(t_{i,0}) = \varepsilon$;
 - Otherwise $T_i = \{t_{i,0}, t_{i,1}, \dots, t_{i,n_i}\}$ with
 1. $W^-(t_{i,0}) = b_{v_i}$ and $W^+(t_{i,0}) = p_{i,0}$; $t_{i,0} \in T_{el}$ and $\lambda(t_{i,0}) = \varepsilon$;
 2. for all $1 < j < n_i$ $W^-(t_{i,j}) = p_{i,j}$ and $W^+(t_{i,0}) = p_{i,j+1}$;
 3. $W^-(t_{i,n_i}) = p_{i,n_i-1}$ and $W^+(t_{i,n_i}) = e_{v_i}$;
 4. for all $1 < j \leq n_i$ if $u_{i,j} \in V$ then $t_{i,j} \in T_{ab}$ and $\Omega(t_{i,j}) = b_{u_i[j]}$ else $t_{i,j} \in T_{el}$, $\lambda(t_{i,j}) = u_i[j]$.

- The initial marking is a single vertex whose marking consists of a token in place b_S . We want to show that $\mathcal{L}(G) = \mathcal{L}(\mathcal{N}_G, \{s_f\})$ where s_f consists in a single thread whose marking consists in a token in place e_S .

The result of a (possibly partial) leftmost derivation, not leading to empty word, is represented by:

$$(w, u_{i_k}[j_k, n_k], u_{i_{k-1}}[j_{k-1}, n_{k-1}], \dots, u_{i_1}[j_1, n_1])$$

with (1) for all $\ell \leq k$ $0 < n_\ell$, (2) $1 \leq j_k \leq n_k + 1$, and (3) for all $\ell < k$ $1 < j_\ell \leq n_{\ell+1}$ where:

- $w \in \Sigma^*$ is the subword already generated;

- $(i_1, j_1) \cdots (i_k, j_k)$ are the current nested rules with the index of the next symbol to be processed (none when $j_\ell = n_\ell + 1$).

The leftmost derivations are simulated by the net as follows. There is a firing sequence with trace w leading to a reachable state defined by:

- if $k = 0$ then a single node whose marking is a token in place e_S ;
- if $k \geq 1$ and $j_k = n_k + 1$ then a single branch of length $k - 1$ labelled by $t_{i_1, j_1 - 1}, \dots, t_{i_{k-1}, j_{k-1} - 1}$ with all markings empty except the last one consisting of a token in e_{v_k} ;
- if $k \geq 1$, $j_k \leq n_k$ and $u_{i_k}[j_k] \in \Sigma$ then a single branch of length $k - 1$ labelled by $t_{i_1, j_1 - 1}, \dots, t_{i_{k-1}, j_{k-1} - 1}$ with all markings empty except the last one consisting of a token in p_{i_k, j_k} ;
- if $k \geq 1$, $j_k \leq n_k$ and $u_{i_k}[j_k] \in V$ then a single branch of length k labelled by $t_{i_1, j_1 - 1}, \dots, t_{i_k, j_k - 1}$ with all markings empty except the last one consisting of a token in $b_{u_{i_k}[j_k]}$;

Let us describe the simulation.

- If $j_k \leq n_k$ and $u_{i_k}[j_k] \in V$ then the derivation consists in choosing some rule $r_{i_{k+1}} = u_{i_k}[j_k] \rightarrow u_{i_{k+1}}$ leading to $(w, u_{i_{k+1}}[1, n_{k+1}], u_{i_k}[j_k + 1, n_k], \dots, u_{i_1}[j_1, n_1])$. This is simulated by the firing of elementary transition $t_{i_{k+1}, 0}$ possibly followed by the firing of abstract transition $t_{i_{k+1}, 1}$ when $u_{i_{k+1}}[1] \in V$;
- If $j_k \leq n_k$ and $u_{i_k}[j_k] \in \Sigma$ then the derivation consists in concatenating the letter $u_{i_k}[j_k]$ to the word w leading to $(wu_{i_k}[j_k], u_{i_k}[j_k + 1, n_k], u_{i_{k-1}}[j_{k-1}, n_{k-1}], \dots, u_{i_1}[j_1, n_1])$. This is simulated by the firing of elementary transition $t_{i_k, j_k + 1}$ possibly followed by the firing of abstract transition $t_{i_k, j_k + 1}$ when $j_k + 1 \leq n_k$ and $u_{i_k}[j_k + 1] \in V$;
- If $j_k = n_k + 1$ then the derivation consists in deleting the empty word $u_{i_k}[j_k, n_k]$ leading to $(w, u_{i_{k-1}}[j_{k-1}, n_{k-1}], \dots, u_{i_1}[j_1, n_1])$. This is simulated by the cut transition in the leaf of the branch producing the token specified by the postcondition of $t_{i_{k-1}, j_{k-1} - 1}$ possibly followed by the firing of abstract transition $t_{i_{k-1}, j_{k-1}}$ when $j_{k-1} \leq n_{k-1}$ and $u_{i_{k-1}}[j_{k-1}] \in V$;

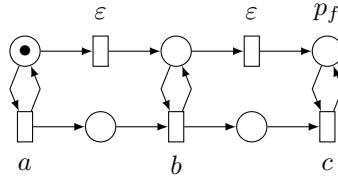
The case of the empty word is straightforward.

Conversely the simulation from traces of firing sequences to words generated by (possibly partial) leftmost derivations are done in a similar way. \square

Proposition 3. *Let $\Sigma = \{a, b, c\}$ and $\mathcal{L}_1 = \{a^m b^n c^p \mid m \geq n \geq p\}$. Then \mathcal{L}_1 is the coverability language of some Petri net and is not a context-free language.*

Proof. Let us recall (a weak version of) Ogden lemma. 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 $n \geq 0$, $w_1 w_2^n w_3 w_4^n w_5 \in \mathcal{L}$.

Assume that \mathcal{L} is a context-free language and consider the word $w = a^N b^N c^N$ with all c positions marked. So $w = w_1 w_2 w_3 w_4 w_5$ with the decomposition fulfilling the requirements of Ogden lemma. Since $w' = w_1 w_2^2 w_3 w_4^2 w_5 \in \mathcal{L}$, 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} . 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} .



Consider the net above and the coverability language associated with the empty marking. It is routine to check that this language is \mathcal{L} . \square

7.3 Proofs of Section 4

Proposition 6. *Let \mathcal{N} be an RPN and $s \xrightarrow{\sigma} s'$ be a firing sequence. There exists a well-sequenced firing sequence $s \xrightarrow{\hat{\sigma}} s'$ in $\hat{\mathcal{N}}$, with $|\hat{\sigma}| \leq |\sigma|$.*

Proof. By construction of $\hat{\mathcal{N}}$, σ is fireable in $\hat{\mathcal{N}}$.

First 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 $(v, \tau) \in \sigma$. One builds σ' by (1) deleting from σ the step (u, t) , (2) deleting all the firings from $Des_{\sigma}(v)$ in σ and (3) replacing the step (v, τ) by (u, t') . We claim that $s \xrightarrow{\sigma'} s'$. Indeed in u the step (u, t') has the same incidence in u as the step (u, t) followed by (v, τ) (thus ‘anticipating’ (v, 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. Denote σ^* the sequence obtained by iterating the process for all such cases.

Let us establish that we can transform σ^* into $\hat{\sigma}$ fulfilling in the requirements for being well-sequenced, by induction on $|V_s \cup V_{s'}|$, i.e. the number of extremal states.

If $|V_s \cup V_{s'}| = 0$, $\sigma^* = \varepsilon$, hence well-sequenced. Assume that we have shown the proposition for any σ^* with $|V_s \cup V_{s'}| < n$, and that now we have σ^* which fulfills $|V_s \cup V_{s'}| = n$. We consider three cases according to the newly added leaf thread u (in s or s') applying the induction for u with maximal depth:

1. $u \in V_{s'} \cap V_s$. Denote by $\sigma_u^* \subset \sigma^*$ the subsequence consisting of the firings performed by u in σ^* and σ_r^* the subsequence of σ^* obtained by removing σ_u^* . Due to the first transformation, σ_u^* consists of elementary transitions. Since u is both an initial and final thread in σ^* , σ_u^* can be fired at the end (or indifferently at the beginning) of σ^* . Observe that σ_r^* is a firing sequence

from s without u to s' without u . So the induction applies: let $\widehat{\sigma}_r$ be the well-sequenced sequence corresponding to σ_r^* . So $\widehat{\sigma}_r\sigma_u^*$ is the sequence we are looking for.

2. $u \in V_{s'} \setminus V_s$. Let (v, t) be the step which creates the thread u . Denote by $\sigma_u^* \subset \sigma^*$ the subsequence consisting of the firings performed by u in σ^* and σ_r^* the subsequence of σ^* obtained by removing (v, t) and σ_u^* . σ_u^* consists of elementary transitions. Since u is a final thread in σ^* , σ_u^* can be fired at the end of σ^* . Observe that σ_r^* is a firing sequence from s to s' without u where the marking of v is increased by $W^-(t)$. So the induction applies: let σ_r^+ be the well-sequenced sequence corresponding to σ_r^* and $\widehat{\sigma}_r$ be σ_r^+ with the firing (v, t) inserted at the end of the subsequence of firings performed by v in σ_r^+ . Then $\widehat{\sigma}_r\sigma_u^*$ is the sequence we are looking for.
3. $u \in V_s \setminus V_{s'}$. Denote by $\sigma_u^* \subset \sigma^*$ the subsequence consisting of the firings performed by u in σ^* and σ_r^* the subsequence of σ^* obtained by removing σ_u^* .
 - If σ_u^* does not end by a cut transition then σ_r^* is a firing sequence from s without u to s' . Let $\widehat{\sigma}_r$ be the well-sequenced sequence corresponding to σ_r^* . But $\widehat{\sigma}_r$ is also a sequence from s to s' .
 - If σ_u^* ends by a cut transition and u is the root then σ_u^* is the sequence we are looking for.
 - If σ_u^* ends by a cut step, v is the parent of u and $\Lambda(v, u) = t$ then σ_r^* is a firing sequence from s without u where the marking of v is increased by $W^+(t)$ to s' . So the induction applies: let $\widehat{\sigma}_r$ be the well-sequenced sequence corresponding to σ_r^* . So $\sigma_u^*\widehat{\sigma}_r$ is the sequence we are looking for.

Since we did not add any new step, $|\widehat{\sigma}| \leq |\sigma|$. □

7.4 Proofs of Section 5

Lemma 3. *Let \mathcal{N} be an RPN and s_{ini} be an initial state. Then there is a deep constrained infinite sequence starting from s_{ini} if and only if there is a path from $u \in V_{s_{ini}}$ to a cycle in $G_{\mathcal{N}, s_{ini}}$.*

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 $|\gamma| > |T_{ab}| + |V_{s_{ini}}|$. Let us denote it by $\gamma := (v_i)_{i=1}^m$. This path can be decomposed as $\gamma = (v_i)_{i=1}^{k-1}(v_i)_{i=k}^m$ with for all $i < k$, $v_i \in V_{s_{ini}}$, for all $i \geq k$, $v_i \notin V_{s_{ini}}$, and $m - k > |T_{ab}|$. For all $i \geq k$ denote by t_i the abstract transition that creates v_i . Using γ , one builds a path $\gamma_{at} = v_{k-1}v_{t_k} \dots v_{t_m}$ in G_{at} as follows. First $v_{k-1} \in V_s$ and $M_{s_{ini}}(v_{k-1}) = M_{at}(v_{k-1})$. Since along σ the thread v_{k-1} fires the t_k to create v_k , there is an edge between v_{k-1} to v_{t_k} in G_{at} . For any $k \leq i \leq m$ the thread v_i is created with the marking $\Omega(t_i) = M_{at}(v_{t_i})$. Since v_{i+1} is a child of v_i , somewhere on the sequence σ the thread v_i fires the abstract transition t_{i+1} . Therefore there is an edge between v_{t_i} to $v_{t_{i+1}}$ in G_{at} .

The path γ_{at} starts from $v_{k-1} \in V_s$ and has length strictly greater than $|T_{ab}| + 1$. Since every vertex on the path γ_{at} except v_{k-1} is some v_t with $t \in T_{ab}$ there at

least a vertex v_t occurring twice in γ_{at} . Therefore there is a path from $v_{k-1} \in V_s$ to a cycle in G_{at} .

Conversely assume that there is a path from $v \in V_s$ to a cycle in G_{at} . Then there is an infinite path $\gamma_{at} = \{v_i\}_{i=0}^\infty$ in $G_{at}(\mathcal{N}, s)$, 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_{ini} . Note that $v_0 \in V_{s_{ini}}$ and that $M_{s_{ini}}(v_0) = M_{at}(v_0)$. By definition of E_{at} 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 $M_{s_1}(v_1) = M_{at}(v_1)$, therefore on can continue translating the path γ_{at} in the same way as the first edge. Since for any (v_i, v_{i+1}) on γ_{at} we create a new thread from v_i one gets an deep sequence. Since the built sequence does not delete no initial thread, it is also constrained. \square

Given two vertices in a graph, the *distance* between them is the length of the shortest path going from one to the other.

Lemma 4. *Given \mathcal{N} an RPN with an initial state s , and σ a shallow sequence. Then there is a thread v that fires infinitely many times in σ .*

Proof. Assume for contradiction that there is no thread v which fires infinitely many times in σ . Let σ be:

$$s_0 \xrightarrow{(u_1, t_1)} s_1 \xrightarrow{(u_2, t_2)} s_2 \xrightarrow{(u_3, t_3)} \dots$$

Denote by $V_0 := V_{s_0}$. Since V_0 is finite and no $v \in V_0$ fires infinitely many times along σ there exists an index m_0 such that for any $i > m_0$, $u_i \notin V_0$. Let us show how to build inductively a set V_k and an index m_k , with the following properties:

- For all $i > m_k$, $u_i \notin \bigcup_{i=0}^k V_k$;
- For all $v \in V_k$ the distance between v and the root is at least k ;
- V_k is finite.

Let $V_k = V_{s_{m_{k-1}}} \setminus \bigcup_{i=0}^{k-1} V_i$. By construction, V_k is finite. Moreover, any thread in V_k has been created by a thread $u_i \in Des_\sigma(V_{k-1})$. Hence it is at least of distance k from the root. Finally, since no thread fires infinitely many times on σ there is an index m_k such that for all $i > m_k$ $u_i \notin V_k$.

Let $n = |T_{ab}| + |V_s| + 1$. The distance between s_{m_n} and the root is greater than $|T_{ab}| + |V_s|$ yielding a contradiction. \square