



Dynamic Recursive Petri Nets

Serge Haddad, Igor Khmelnitsky

► **To cite this version:**

Serge Haddad, Igor Khmelnitsky. Dynamic Recursive Petri Nets. PETRI NETS 2020 - 41st International Conference on Application and Theory of Petri Nets and Concurrency, Jun 2020, Paris, France. hal-02511321

HAL Id: hal-02511321

<https://hal.inria.fr/hal-02511321>

Submitted on 18 Mar 2020

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.

Dynamic Recursive Petri Nets

Serge Haddad¹ and Igor Khmelnitsky¹

LSV, Université Paris-Saclay, ENS Paris-Saclay, CNRS, Inria France

Abstract. In the early two-thousands, Recursive Petri nets (RPN) have been introduced in order to model distributed planning of multi-agent systems for which counters and recursivity were necessary. While having a great expressive power, RPN suffer two limitations: (1) they do not include more general features for transitions like reset arcs, transfer arcs, etc. (2) the initial marking associated the recursive “call” only depends on the calling transition and not on the current marking of the caller. Here we introduce Dynamic Recursive Petri nets (DRPN) which address these issues. We show that the standard extensions of Petri nets for which decidability of the coverability problem is preserved are particular cases of DRPN. Then we establish that w.r.t. coverability languages, DRPN are strictly more expressive than RPN. Finally we prove that the coverability problem is still decidable for DRPN.

Keywords: Recursive Petri nets · Expressiveness · Coverability · Decidability.

1 Introduction

Limitations of Petri nets (PN). When modelling dynamic systems, formalisms that can handle infinite transition systems are required in several contexts among them: the concurrent execution of parallel sequential processes that produce and consume resources and the dynamical creation of processes. While Petri nets are well suited for specifying the first pattern, their static structure forbids the modelling of the second pattern. Furthermore even the management of an unbounded number of resources by PNs suffers some limitations: the reset operation that cleans a buffer or the transfer operation of a set of resources from one buffer to another one cannot be performed in an atomic way.

Recursive Petri nets (RPN). This formalism has been introduced in order to address the issue of modelling dynamic structures with PN [3] (see also [12,13] for similar models). 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, is decidable [8,9] by reducing this property to several reachability problems of PNs. Furthermore, the coverability and termination problems of RPNS have the same complexity as the ones of PNs (EXPSpace-complete, see [4]). In [10], several additional features are proposed while preserving the decidability of the verification problems.

Static extensions of Petri nets. In another direction, PNs have been extended by adding capabilities of transitions while the static structure given by the set of places is unchanged. The reset and transfer arcs allow to perform the corresponding operations by a single transition [2]. The arcs of self-modifying nets are labelled by expressions in such a way that the numbers of tokens consumed or produced by the transitions depend on the current marking [14]. Affine Petri nets unifies the previous extensions with a concise syntax [6]. While reachability becomes undecidable, depending on the model several properties remain decidable including coverability (implying sometimes weak restrictions).

Our contribution. While having a great expressive power, RPN suffer two limitations: (1) they do not include more general features for transitions like reset arcs, transfer arcs, etc. (2) the initial marking associated the recursive “call” only depends on the calling transition and not on the current marking of the caller. So we introduce Dynamic Recursive Petri nets (DRPN) which address these issues. We show that the extensions of Petri nets (discussed above) for which decidability of the coverability problem is preserved are particular cases of DRPN. Then we establish that w.r.t. coverability languages, DRPN are strictly more expressive than RPN. Finally we prove that the coverability problem is still decidable for DRPN.

Outline. In section 2 we introduce DRPN, illustrate their modelling capabilities and define a quasi-order between states of DRPN. In section 3, we study from a theoretical point of view, expressiveness of DRPNs. In section 4, we establish that the coverability problem is decidable. Finally in section 5, we conclude and give some perspectives to this work. All missing proofs can be found in the appendix.

2 Dynamic Recursive Petri Nets

Well quasi-ordered sets. A quasi-ordered set (X, \leq) is *well quasi-ordered* if given any infinite sequence $(x_n)_{n \in \mathbb{N}} \in X^\omega$, there exist $i < j$ such that $x_i \leq x_j$. For instance, \mathbb{N}^P , where P is a finite set, equipped with the component order is well-ordered. From a computability point of view, we assume that the representation of items of X allows to decide whether $x \leq x'$ which is obviously the case for \mathbb{N}^P . Well quasi-ordered sets fulfill properties that we exploit here:

- Let $Y \subseteq X$. Then Y is *upward closed* if for all $x \leq x' \in X$, $x \in Y$ implies $x' \in Y$. Given an arbitrary set Y , the *upward closure* of Y , denoted Y^\uparrow is defined by $Y^\uparrow = \{x' \mid \exists x \in Y \ x \leq x'\}$. The set of minimal elements of an upward closed set Y , denoted $\min(Y)$, is finite and fulfills $Y = \min(Y)^\uparrow$. So whenever we handle upward closed sets, they are implicitly defined by their set of minimal elements.

- Given any infinite sequence of upward closed sets $(Y_n)_{n \in \mathbb{N}} \in (2^X)^\omega$ where for all n , $Y_n \subseteq Y_{n+1}$, there exists n_0 such that $\bigcup_{n \in \mathbb{N}} Y_n = \bigcup_{n \leq n_0} Y_n$.
- Let f be a partial non decreasing function from X to X , f is *effective* if (1) there is an algorithm that takes as input $x \in X$, decides whether x belongs to the domain of f and, in the positive case, computes $f(x)$ and (2) there is an algorithm that takes as input $x \in X$ and computes $\min(f^{-1}(\{x\}^\uparrow))$.
- Let \mathcal{F} be a finite set of effective functions and Y be an upward closed set. Define $Cov(\mathcal{F}, Y)$ as the smallest set C that contains Y and fulfills $\bigcup_{f \in \mathcal{F}} f^{-1}(C) \subseteq C$. Then $Cov(\mathcal{F}, Y)$ can be computed the following *backward exploration*:

$$C \leftarrow Y; \text{ \textbf{repeat} } \text{ old}C \leftarrow C; C \leftarrow C \cup \bigcup_{f \in \mathcal{F}} f^{-1}(C) \text{ \textbf{until} } C = \text{old}C$$

For instance, coverability in Petri nets can be decided using this backward exploration (see for instance [7]) and we will apply it in several contexts.

Notation. Let $X \subseteq Y$ be two sets. The mapping **Id** denotes the identity mapping from X to Y where X and Y should be clear from the context.

Let us introduce dynamic recursive Petri nets (DRPN). Like a Petri net, a DRPN has a set of places P and a set of transitions T partitioned in elementary and abstract transitions (resp. T_{el} and T_{ab}). A state s of an SRPN is a tree whose vertices are labelled by markings (defined by the mapping M) and edges are labelled by transitions (defined by the mapping Λ). A transition t may fire in any vertex u provided that the marking of this vertex $M(u)$ belongs to an upward closed set Grd_t . If t is elementary then $M(u)$ is updated by applying an effective function Upd_t . If t is abstract then (1) a vertex v is created as a child of u with $\Lambda(u, v) = t$, (2) marking $M(v)$ is defined by $Beg_t(M(u))$ where Beg_t is an effective function, and (3) $M(u)$ is updated by applying an effective function $Upd_t^- \leq \mathbf{Id}$. A DRPN is equipped with End , an upward closed set of \mathbb{N}^P . When for some vertex v , $M(v) \in End$ then τ , the *cut transition*, can be fired whose effect consists to (1) delete the subtree rooted v and (2) when $v \neq r$ where r denotes the root of the state to update $M(u)$ where u is the parent of v by applying the effective function $Upd_t^+ \geq \mathbf{Id}$. The state consisting in the empty tree is denoted \perp .

Definition 1 (DRPN). A Dynamic Recursive Petri Net is a 6-tuple $\mathcal{N} = \langle P, T, Grd, Upd, Upd^-, Upd^+, Beg, End \rangle$ where:

- P is a finite set of places;
- $T = T_{el} \uplus T_{ab}$ with $P \cap T = \emptyset$ is a finite set of transitions;
- $Grd = \{Grd_t\}_{t \in T}$ is a family of upward closed sets of \mathbb{N}^P ;
- $Upd = \{Upd_t\}_{t \in T_{el}}$ is a family of effective functions with $Upd_t \in (\mathbb{N}^P)^{Grd_t}$;
- $Upd^- = \{Upd_t^-\}_{t \in T_{ab}}$ is a family of effective functions with $Upd_t^- \in (\mathbb{N}^P)^{Grd_t}$;
- $Upd^+ = \{Upd_t^+\}_{t \in T_{ab}}$ is a family of effective functions with $Upd_t^+ \in (\mathbb{N}^P)^{\mathbb{N}^P}$;
- For all $t \in T_{ab}$, $Upd_t^- \leq \mathbf{Id}$ and $\mathbf{Id} \leq Upd_t^+$;
- $Beg = \{Beg_t\}_{t \in T_{ab}}$ is a family of effective functions with $Beg_t \in (\mathbb{N}^P)^{Grd_t}$;

- *End* is an upward closed set of \mathbb{N}^P .

As discussed above, a state of a DRPN is a labelled tree.

Definition 2 (State). Let \mathcal{N} be a DRPN. Then a state $s = \langle V, M, E, \Lambda \rangle$ of \mathcal{N} is defined by:

- V its finite set of vertices;
- $M : V \rightarrow \mathbb{N}^P$, a function that labels vertices with markings;
- $E \subseteq V \times V$, a set of edges such that (V, E) is a directed tree;
- $\Lambda : E \rightarrow T_{ab}$, a function that labels edges with abstract transitions.

One denotes by $Des_s(v)$ (respectively $Anc_s(v)$) the set of descendants (respectively ancestors) of $v \in V$ in the underlining tree of s (including v itself). If $v \neq r$ then $prd(v)$ is the parent of v in the tree. Given a $U \subseteq V$ we will denote by $Anc_s(U) = \cup_{v \in U} Anc_s(v)$. The *depth* of s is the depth of its tree. Given $\mathbf{m} \in \mathbb{N}^P$, $s_{\mathbf{m}}$ denotes a tree consisting of a single vertex r with marking $M(r) = \mathbf{m}$. Let us formally define the firing of elementary, abstract and cut transitions.

Definition 3. Let \mathcal{N} be a DRPN, s a state of \mathcal{N} , $v \in V$ and $t \in T \cup \{\tau\}$. t is fireable by v from s if either $t \neq \tau$ and $M(v) \in Grd_t$ or $t = \tau$ and $M(v) \in End$. In this case, its firing leads to the state $s' = \langle V', M', E', \Lambda' \rangle$, defined below:

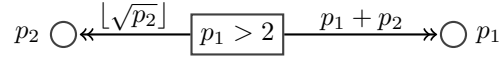
- If $t \in T_{el}$ then $s' = \langle V, M', E, \Lambda \rangle$ where $M'(u) = M(u)$ for all $u \in V \setminus \{v\}$ and $M'(v) = Upd_t(M(v))$;
- If $t \in T_{ab}$ then:
 - $V' = V \cup \{w\}$ where w is a fresh identifier;
 - $M'(u) = M(u)$ for all $u \in V \setminus \{v\}$, $M'(v) = Upd_t^-(M(v))$ and $M'(w) = Beg_t(M(v))$;
 - $E' = E \cup \{(v, w)\}$ and $\Lambda'(e) = \Lambda(e)$ for all $e \in E$ and $\Lambda'((v, w)) = t$.
- If $t = \tau$ and $v = r$ then $s' = \perp$;
- If $t = \tau$ and $v \neq r$ then let $w = prd(v)$:
 - $V' = V \setminus Des_s(v)$;
 - for all $u \neq w$, $M'(u) = M(u)$ and $M'(w) = Upd_{\Lambda(w,v)}^+(M(w))$;
 - $E' = E \cap (V' \times V')$ and Λ' is the restriction of Λ on E' .

The transition firing is denoted $s \xrightarrow{(v,t)} s'$ and when there are several nets, $s \xrightarrow{(v,t)}_{\mathcal{N}} s'$. 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 $\sigma = (v_1, t_1)(v_2, t_2) \dots (v_n, t_n)$. The *length* of σ , denoted $|\sigma|$, is n . The *abstract length* of σ , denoted $|\sigma|_{ab}$, is $|\{i \leq n \mid t_i \in T_{ab}\}|$. The *depth* of σ is the maximal depth of states s_0, \dots, s_n . A *closing sequence* is a firing sequence that reaches \perp . Given a firing sequence that includes the firing of an abstract transition t in vertex v creating vertex w and followed later by the cut transition in w , we say that $(v, t), (w, \tau)$ are *matched* in σ . The *reachability* set $Reach(\mathcal{N}, s_0)$ is defined by: $Reach(\mathcal{N}, s_0) = \{s \mid \exists \sigma s_0 \xrightarrow{\sigma} s\}$.

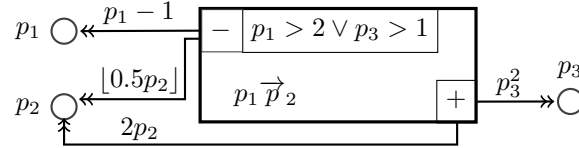
Discussion. The main limitations of the modelling power of DRPN are the requirements that (1) sets like Grd_t must be upward closed and (2) functions like Upd_t must be monotonic. Despite these limitations, DRPNs include many models like:

- *Petri nets* (PN) that can be defined without abstract transitions and such that for all $t \in T$, $Grd_t = \{\mathbf{m} \mid \mathbf{m} \geq \mathbf{Pre}(t)\}$ and $Upd_t = \mathbf{Id} + \mathbf{C}(t)$ where $\mathbf{Pre}(t)$ (resp. $\mathbf{C}(t)$) is the column vector indexed by t of the backward incidence matrix \mathbf{Pre} (resp. incidence matrix \mathbf{C}).
- *Affine Petri nets* ([6]) that can also be defined without abstract transitions and such that for all $t \in T$, there exist a matrix $\mathbf{A}_t \in \mathbb{N}^{P \times P}$ and a vector $\mathbf{B}_t \in \mathbb{Z}^P$ with $Grd_t = \{\mathbf{m} \mid \mathbf{A}_t \mathbf{m} + \mathbf{B}_t \geq 0\}$ and $Upd_t(\mathbf{m}) = \mathbf{A}_t \mathbf{m} + \mathbf{B}_t$.
- *Recursive Petri nets* (RPN) ([9]) such that for all $t \in T$, $Grd_t = \{\mathbf{m} \mid \mathbf{m} \geq \mathbf{Pre}(t)\}$ and when $t \in T_{el}$, $Upd_t = \mathbf{Id} + \mathbf{C}(t)$ and when $t \in T_{ab}$, $Upd_t^- = \mathbf{Id} - \mathbf{Pre}(t)$, $Upd_t^+ = \mathbf{Id} + \mathbf{Post}(t)$ and Beg_t is some constant. Here $\mathbf{Post}(t)$ is the column vector indexed by t of the forward incidence matrix \mathbf{Post} .

Graphical representation. For modelling purposes, we equip DRPN with a graphical representation based on net representations. Places (resp. transitions) are depicted by circles (resp. rectangles). However a transition does not have input arcs but only output arcs represented by double-headed arrows and labelled by expressions where a place represents the current value of its marking. The guard of an elementary transition is also represented by a boolean expression inside the rectangle. For instance, the elementary transition t figured below is defined by: $Grd_t = \{\mathbf{m} \mid \mathbf{m}(p_1) > 2\}$ and $Upd_t(\mathbf{m}) = (\mathbf{m}(p_1) + \mathbf{m}(p_2))\vec{p}_1 + \lfloor \sqrt{\mathbf{m}(p_2)} \rfloor \vec{p}_2$ where \vec{p} denotes the vector defined by $\vec{p}[p] = 1$ and for all $p' \neq p$, $\vec{p}[p'] = 0$.



The rectangle of an abstract transition t is divided into several parts: on the top corner left ($-$) starts the edges representing Upd_t^- , on the top center Grd_t is represented, on the bottom corner left Beg_t is represented, and on the bottom corner right ($+$) start the edges representing Upd_t^+ . There are no edges for unchanged place markings. For instance, the abstract transition figured below is defined by: $Grd(t) = \{\mathbf{m} \mid \mathbf{m}(p_1) > 2 \vee \mathbf{m}(p_3) > 1\}$, $Upd_t^-(\mathbf{m}) = (\mathbf{m}(p_1) - 1)\vec{p}_1 + \lfloor 0.5\mathbf{m}(p_2) \rfloor \vec{p}_2 + \mathbf{m}(p_3)\vec{p}_3$, $Upd_t^+(\mathbf{m}) = \mathbf{m}(p_1)\vec{p}_1 + 2\mathbf{m}(p_2)\vec{p}_2 + \mathbf{m}(p_3)^2\vec{p}_3$, and $Beg_t = \mathbf{m}(p_1)\vec{p}_2$. Observe that $Upd_t^- \leq \mathbf{Id}$ and $\mathbf{Id} \leq Upd_t^+$.



Example 1 (Hiring an assassin). In order to illustrate the modelling capabilities of DRPN, we present an example of distributed planning. The DRPN \mathcal{N}_{Jaqen} of Figure 1 represents the possible behaviour of an assassin hired for a job. The transitions filled in black are Petri net transitions and so are presented with input and output arcs as usual.

The assassin is given 3 days (3 tokens in p_{time}), an advance of 20 bitcoins (20 tokens in p_{adv}), and is promised to get a reward of 20 bitcoins after the job is done (20 tokens in p_{reward}). In order to try catching their target he needs to devote one bitcoin and one day of his time. After this day either the assassin is successful (t_{found}) or fails (t_{lost}) and needs to spend another day. When successful, the assassin can collect the reward ($t_{collect}$). However, the assassin has also another strategy which consists of hiring another assassin by giving him a quarter of his advance money and promise him an equal reward, telling him the number of days left (t_{hire} where $f_{pay}(\mathbf{m}) = \mathbf{m}(p_{time})\vec{p}_{time} + [0.5[0.5\mathbf{m}(p_{adv})]]\vec{p}_{adv} + [0.5[0.5\mathbf{m}(p_{adv})]]\vec{p}_{reward}$). If some hired assassin is successful then he can report his success to the hiring guy by firing the cut transition (due to the specification *End*). The state presented on the right of Figure 1 consists of three assassins where the last hired one has killed the target. Observe that as long as a guy has money he can hire several assassins and that even after hiring he can still try to kill the target by himself.

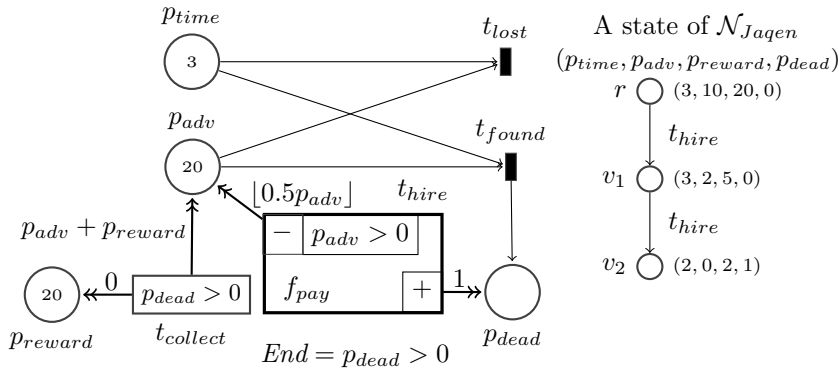


Fig. 1. A DRPN with a state

A firing sequence of \mathcal{N}_{Jagen} is presented in Figure 2 where the vertex who fires the transition is filled in black. The initial assassin first tries to find the target but fails ((r, t_{lost})), losing one bitcoin and one day. Then he hires another assassin by firing the abstract transition (r, t_{hire}), losing half of his advance money and creating a new vertex v , where the hired assassin has two days, an advance of five bitcoins and a promised reward of five bitcoins for completing the job ($M(v) = 2\vec{p}_{time} + 5\vec{p}_{adv} + 5\vec{p}_{reward}$). This assassin kills the target and collects the reward ((v, t_{lost}) followed by $(v, t_{collect})$). Then using the cut transition he reports it to his employer ((v, τ)), which removes v and adds one token to p_{dead} in $M(r)$. Finally the original assassin can collect his money by firing $t_{collect}$.

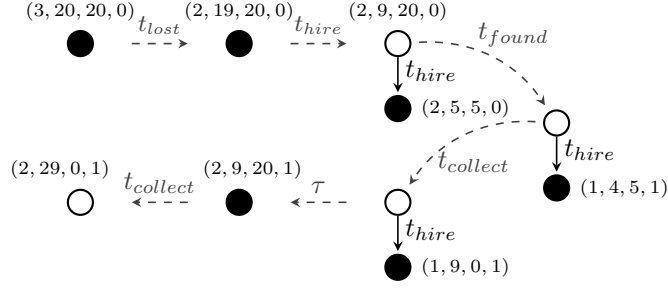


Fig. 2. Firing sequence

Ordering states of a DRPN. We now define a quasi-order on the states of a DRPN. Given two states s, s' of \mathcal{N} we say that s is smaller or equal than s' or equivalently that s' covers s if (without considering labels) there is a subtree in s' isomorphic to s (by some matching) such that (1) given any pair of matched vertices (u, u') , $M(u) \leq M'(u')$ and (2) given any pair of transitions (t, t') labelling matched edges, $Upd_t^+ \leq Upd_{t'}^+$.

Definition 4. Let \mathcal{N} be DRPN and $s = \langle V, M, E, A \rangle$, $s' = \langle V', M', E', A' \rangle$ of a DRPN \mathcal{N} , be two states of \mathcal{N} and φ be an injective mapping from V to V' . We say that s' covers s by φ , denoted $s \preceq_{\varphi} s'$ if:

1. For all $v \in V$, $M(v) \leq M'(\varphi(v))$;
2. For all $(u, v) \in E$, $(\varphi(u), \varphi(v)) \in E'$ and $Upd_{\Lambda(u,v)}^+ \leq Upd_{\Lambda'(\varphi(u), \varphi(v))}^+$.

We say that s' covers s denoted $s \preceq s'$ if there exists φ such that $s \preceq_{\varphi} s'$.

Given states s, s' , deciding whether s' covers s is a necessary condition for designing algorithms related to the coverability relation. So we assume that given a DRPN, for all pairs $t, t' \in T_{ab}$ one can decide whether $Upd_t^+ \leq Upd_{t'}^+$. This hypothesis is satisfied by all “reasonable” effective functions.

The *coverability set* $Cov(\mathcal{N}, s_0)$ is defined as the upward closure of the reachability set: $Cov(\mathcal{N}, s_0) = Reach(\mathcal{N}, s_0)^\uparrow$. As for recursive Petri nets (see [4]) this quasi-order is strongly compatible (and even more): for all $s \preceq_{\varphi} s'$ and $s \xrightarrow{(v,t)} s_1$ there exists $s'_1 \preceq_{\varphi_1} s_1$ such that $s' \xrightarrow{(\varphi(v), t)} s'_1$ and φ and φ_1 coincide on the intersection of their domain. However this quasi-order is not a well quasi-order (see also [4]) and thus in order to solve the coverability problem, one cannot apply the backward exploration.

Notation. Let S_t be a finite set of states. We call a sequence σ such that $s \xrightarrow{\sigma} s' \succeq s'' \in S_t$ a *covering sequence*.

3 Expressiveness

Expressiveness of a formalism may be defined by the family of languages that it can generate. In [4], expressiveness of RPNs was studied using coverability

languages. In order to compare RPN and DRPN we need to define coverability languages of DRPNs and so we equip any transition $t \in T \cup \tau$ with a *label* $\lambda(t) \in \Sigma \cup \{\varepsilon\}$ where Σ is an alphabet and ε is the empty word of Σ^* fulfilling $\lambda(\tau) = \varepsilon$. The labelling is extended to transition sequences in the usual way. Thus given a labelled marked DRPN (\mathcal{N}, s_{init}) and a finite set of states S_t , the coverability language $\mathcal{L}(\mathcal{N}, s_{init}, S_t)$ is defined by:

$$\mathcal{L}(\mathcal{N}, s_{init}, S_t) = \{\lambda(\sigma) \mid \exists s_0 \xrightarrow{\sigma} s \succeq s' \wedge s' \in S_t\}$$

i.e. the set of labellings for sequences covering some state $s \in S_t$ of \mathcal{N} . We say that $\mathcal{L} \subseteq \Sigma^*$ is a *coverability language* if $\mathcal{L} = \mathcal{L}(\mathcal{N}, s_{init}, S_t)$ for some \mathcal{N} , s_{init} and S_t . We also introduce $\mathcal{L}_B(\mathcal{N}, s_{init}, S_t)$ the B -bounded coverability language with $B \in \mathbb{N}$.

$$\mathcal{L}_B(\mathcal{N}, s_{init}, S_t) = \{\lambda(\sigma) \mid \exists s_0 \xrightarrow{\sigma} s \succeq s' \wedge s' \in S_t \wedge |\sigma|_{ab} \leq B\}$$

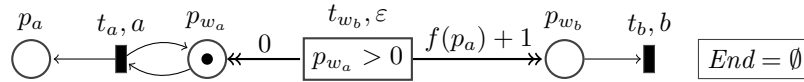
We say that $\mathcal{L} \subseteq \Sigma^*$ is a *B -bounded coverability language* if $\mathcal{L} = \mathcal{L}_B(\mathcal{N}, s_{init}, S_t)$ for some \mathcal{N} , s_{init} and S_t .

The next proposition is an important ingredient for our expressiveness result (see proof in the appendix). The main idea of this proof consists in considering a PN with enough copies of P and T such that each copy mimicks the behaviour of a vertex in a state of the RPN with height at most B and outgoing degree at most B for every vertex. Additional places and transitions allow to express the child relation between vertices and the existence of the vertices in the current mimicked state of the RPN.

Proposition 1. *Let \mathcal{L} be a B -bounded coverability language. Then \mathcal{L} is a PN coverability language.*

We say that a function $f : \mathbb{N} \rightarrow \mathbb{N}$ is *sublinear* if $\lim_{n \rightarrow \infty} \frac{f(n)}{n} = 0$. Let f be sublinear non decreasing with $\lim_{n \rightarrow \infty} f(n) = \infty$, we define on the alphabet $\{a, b\}$ the language $\mathcal{L}_f = \{a^k b^m \mid m \leq f(k)\}$. Examples of such functions are $\lfloor \log(1+n) \rfloor$ or $\lfloor \sqrt{n} \rfloor$. As an immediate consequence of the properties of f , one defines by induction the strictly increasing sequence $(\alpha(n))_{n \in \mathbb{N}}$: $\alpha(0) = 0$ and $\alpha(n+1) = \min(m \mid \alpha(n) < m \wedge f(\alpha(n)) < f(m))$. Note that α depends on f , but since in the sequel we consider a single arbitrary f , for sake of readability we write α instead of α_f .

The next proposition establishes that \mathcal{L}_f is a DRPN coverability language. Indeed the coverability language of the DRPN below (without abstract transitions) such that the initial state consists of a single vertex with one token in p_{w_a} and the final state consists of a single vertex with one token in p_{w_b} is \mathcal{L}_f (see the full proof in appendix). So we will use \mathcal{L}_f for witnessing that DRPN coverability languages strictly include RPN coverability languages.



Proposition 2. *For all f , \mathcal{L}_f is a DRPN coverability language.*

The remainder of this section consists in showing that \mathcal{L}_f is not an RPN coverability language. Let us pick an arbitrary labelled RPN \mathcal{N} with an initial

state s_{init} , a finite set of states S_t such that $\mathcal{L}_f \subseteq \mathcal{L}(\mathcal{N}, s_{init}, S_t)$. Let $\{\sigma_n\}_{n \in \mathbb{N}}$ be a family of sequences covering some state of S_t such that $\lambda(\sigma_n) = a^{\alpha(n)} b^{f(\alpha(n))}$ where among the possible σ_n 's we pick one with the minimal depth and among those with minimal depth one with the minimal length (i.e. $\min |\sigma_n|$). The skeleton of the proof is as follows.

- \mathcal{L}_f is not a PN coverability language (Proposition 3 proved in appendix). Therefore there does not exist B such that \mathcal{L}_f is a B -bounded language.
- If the depth of $\{\sigma_n\}_{n \in \mathbb{N}}$ is bounded then $\mathcal{L}(\mathcal{N}, s_{init}, S_t)$ is a B -bounded language (Proposition 4) which shows that $\mathcal{L}_f \subsetneq \mathcal{L}(\mathcal{N}, s_{init}, S_t)$.
- If the depth of $\{\sigma_n\}_{n \in \mathbb{N}}$ is unbounded $\mathcal{L}(\mathcal{N}, s_{init}, S_t)$ contains words that do not belong to \mathcal{L}_f , (Proposition 5) which shows that $\mathcal{L}_f \subsetneq \mathcal{L}(\mathcal{N}, s_{init}, S_t)$.

The following proposition is obtained by an adaptation of a result related to the (non) weak computability of sublinear functions in PN [11].

Proposition 3. *\mathcal{L}_f is not a PN coverability language.*

Remark 1. In the appendix of [5] Lemma 6 shows that for any RPN language there exists a marked RPN with an initial state consisting of only one vertex. Therefore we will assume in the following that s_{init} consists of a single vertex.

In order to alleviate notations in RPN and to be consistent with the notations of DRPN, for all $t \in T$, we denote $\mathbf{Pre}(t)$, $\mathbf{Post}(t)$ and $\mathbf{C}(t)$ respectively by \mathbf{Pre}_t , \mathbf{Post}_t and \mathbf{C}_t .

Given a labelled RPN \mathcal{N} and an abstract transition t , we introduce the following languages:

$$\mathcal{L}_{\mathcal{N}}(t) = \{\lambda(\sigma) \mid \text{Beg}_t \xrightarrow{\sigma}\}; \quad \mathcal{L}_{\mathcal{N}}^{\perp}(t) = \{\lambda(\sigma) \mid \text{Beg}_t \xrightarrow{\sigma} \perp\}$$

Proposition 4. *Let \mathcal{N} be a labelled RPN, s_{init} be its initial state and S_t be a finite set of states such that $\mathcal{L}_f \subseteq \mathcal{L}(\mathcal{N}, s_{init}, S_t)$. Assume that the depths of $\{\sigma_n\}_{n \in \mathbb{N}}$ are bounded. Then $\mathcal{L}_f \subsetneq \mathcal{L}(\mathcal{N}, s_{init}, S_t)$.*

Proof. Let D denote a bound of the depths of the family of $\{\sigma_n\}_{n=1}^{\infty}$, and let $\mathcal{L}_{\mathcal{N}}$ denote more concisely $\mathcal{L}(\mathcal{N}, s_{init}, S_t)$. We are going to build a net \mathcal{N}'' and some \mathcal{L}'' , a B -bounded language of \mathcal{N}'' such that: $\mathcal{L}_f \subseteq \mathcal{L}''$. Due to Proposition 1 and 3, $\mathcal{L}_f \subsetneq \mathcal{L}''$. We stop the construction earlier if we can conclude that $\mathcal{L}_f \subsetneq \mathcal{L}_{\mathcal{N}}$. Otherwise the relation between \mathcal{L}'' and $\mathcal{L}_{\mathcal{N}}$ will allow us to conclude.

We first build an RPN \mathcal{N}' that fulfills $\mathcal{L}(\mathcal{N}', s_{init}, S_t) = \mathcal{L}_{\mathcal{N}}$ as follows. For all $t \in T_{ab}$, one adds places and transitions according to $\mathcal{L}_{\mathcal{N}}(t)$ and $\mathcal{L}_{\mathcal{N}}^{\perp}(t)$:

- If $a^+ b^+ \cap \mathcal{L}_{\mathcal{N}}(t) = \emptyset$ and $z_t = \max\{m \mid b^m \in \mathcal{L}_{\mathcal{N}}(t)\} < \infty$ then one adds elementary transitions t^- , t^b and a place p_t (see left side of the figure below), where:

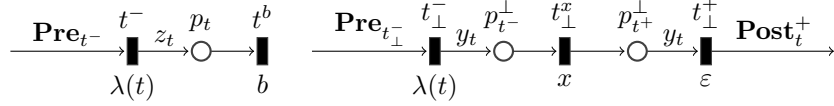
$$\begin{aligned} \mathbf{Pre}_{t^-} &= \mathbf{Pre}_t, & \mathbf{C}_{t^-} &= -\mathbf{Pre}_t + z_t \cdot \vec{p}_t, & \lambda(t^-) &= \lambda(t); \\ \mathbf{Pre}_{t^b} &= \vec{p}_t, & \mathbf{C}_{t^b} &= -\vec{p}_t, & \lambda(t^b) &= b; \end{aligned}$$

- $a^+ b^+ \cap \mathcal{L}_{\mathcal{N}}^{\perp}(t) = \emptyset$ and $\max\{m \mid b^m \in \mathcal{L}_{\mathcal{N}}^{\perp}(t)\} < \infty$ then one adds, elementary transitions $t_{\perp}^-, t_{\perp}^x, t_{\perp}^+$ and places p_{t^-}, p_{t^+} such that where the y_t and x are defined

below:

$$\begin{aligned} \mathbf{Pre}_{t_{\perp}^-} &= \mathbf{Pre}_t, & \mathbf{C}_{t_{\perp}^-} &= y_t \cdot \vec{p}_{t^-}^{\perp} - \mathbf{Pre}_{t_{\perp}^-}, & \lambda(t_{\perp}^-) &= \lambda(t); \\ \mathbf{Pre}_{t_{\perp}^x} &= \vec{p}_{t^-}^{\perp}, & \mathbf{C}_{t_{\perp}^x} &= \vec{p}_{t^+}^{\perp} - \vec{p}_{t^-}^{\perp}, & \lambda(t_{\perp}^x) &= x; \\ \mathbf{Pre}_{t_{\perp}^+} &= y_t \cdot \vec{p}_{t^+}^{\perp}, & \mathbf{C}_{t_{\perp}^+} &= \mathbf{Post}_t - y_t \cdot \vec{p}_{t^+}^{\perp}, & \lambda(t_{\perp}^+) &= \varepsilon \end{aligned}$$

- If $b^m \in \mathcal{L}_t^{\perp}$ for $m > 0$ then $y_t = \max\{m \mid b^m \in \mathcal{L}_t^{\perp}\}$ and $x = b$;
- Else if $a^{\ell} \in \mathcal{L}_t^{\perp}$ for $\ell > 0$ then $y_t = \min\{\ell \mid a^{\ell} \in \mathcal{L}_t^{\perp}\}$ and $x = a$;
- Otherwise, $y_t = 1$ and $x = \varepsilon$.



On the one hand $\mathcal{L}_{\mathcal{N}} \subseteq \mathcal{L}(\mathcal{N}', s_{init}, S_t)$ since any firing sequence in \mathcal{N} can be performed in \mathcal{N}' . On the other hand, the new transitions are built according to $\mathcal{L}_{\mathcal{N}}(t)$ and $\mathcal{L}_{\mathcal{N}}^{\perp}(t)$ in such a way that every firing of a new transition can be replaced by a firing of a sequence of transitions with the same produced label. Hence $\mathcal{L}(\mathcal{N}', s_{init}, S_t) = \mathcal{L}_{\mathcal{N}}$.

We now show that for \mathcal{N}' there exists some B , such for all $n \in \mathbb{N}$ there is a firing sequence σ'_n in \mathcal{N}' with $|\sigma'_n|_{ab} \leq B$ and $\lambda'(\sigma'_n) = \lambda(\sigma_n)$. Denote by $G = \max\{|V_{s_t}| \mid s_t \in S_t\}$. Pick an arbitrary $n \in \mathbb{N}$ and denote more explicitly the covering sequence $s_{init} \xrightarrow{\sigma_n} s_{\varphi} \succeq s_t \in S_t$. Assume there is an occurrence $t \in T_{ab}$ by the vertex u in σ_n creating a vertex v . We transform σ_n according to whether the firing (u, t) has a matching cut transition (v, τ) in σ_n :

The firing (u, t) does not have a matching cut.

- If $a^+b^+ \cap \mathcal{L}_t \neq \emptyset$ then let us suppose that there are more than $2D + G$ occurrences of t without a matching cut. Then there are two vertices v, v' created by t in σ_n which are not: (1) in $\varphi(V_{s_t})$, (2) in the branch leading to it, and (3) both in the same branch. Therefore, one can build a covering sequence σ with $\ell_i, m_i > 0$ such that $\lambda(\sigma) = \dots a^{\ell_1} b^{m_1} \dots a^{\ell_2} b^{m_2} \dots \in \mathcal{L}_{\mathcal{N}}$. So $\mathcal{L}_f \subsetneq \mathcal{L}_{\mathcal{N}}$ and we are done.
- If $\max\{m \mid b^m \in \mathcal{L}_{\mathcal{N}}(t)\} = \infty$ then let us suppose that there are more than $D + G$ occurrences of t without a matching cut. There is a vertex v created by t which is neither in $\varphi(V_{s_t})$ nor in the branch leading to it. Then one can build a covering sequence σ with $k \leq \alpha(n)$ and $m > f(k)$ such that $\lambda(\sigma) = a^k b^m \in \mathcal{L}_{\mathcal{N}}$. So $\mathcal{L}_f \subsetneq \mathcal{L}_{\mathcal{N}}$ and we are done.
- Otherwise (i.e. $a^+b^+ \cap \mathcal{L}_{\mathcal{N}}(t) = \emptyset$ and $z_t = \max\{m \mid b^m \in \mathcal{L}_{\mathcal{N}}(t)\} < \infty$), we replace the firing of (u, t) and all firings from $Des(v)$ by:

$$(u, t^-) \overbrace{(u, t^b) \dots (u, t^b)}^{z_t \text{ times}}$$

which will give us a covering sequence σ'_n such that $\lambda(\sigma'_n) = a^{\ell} b^m$ with $\ell \leq \alpha(n)$ and $m \geq f(\alpha(n))$. If $\ell < \alpha(n)$, and $m > f(\alpha(n))$ then $\mathcal{L}_f \subsetneq \mathcal{L}_{\mathcal{N}}$ and we are done. Otherwise $\lambda'(\sigma'_n) = \lambda(\sigma'_n)$ and $|\sigma'_n|_{ab} < |\sigma_n|_{ab}$. We can repeat this process until either one concludes that $\mathcal{L}_f \subsetneq \mathcal{L}_{\mathcal{N}}$ or there is no more firing of t without a matching cut transition in σ'_n .

The firing (u, t) has a matching cut.

- Assume that $a^+b^+ \cap \mathcal{L}_{\mathcal{N}}^\perp(t) \neq \emptyset$. If there are more than D occurrences of t with a matching cut in σ_n then there are two occurrences (w_1, t) and (w_2, t) where w_2 is neither a descendent nor an ascendent of w_1 . So one could build a covering sequence σ with $\ell_i, m_i > 0$ such that $\lambda(\sigma) = \dots a^{\ell_1} b^{m_1} a^{\ell_2} b^{m_2} \dots \in \mathcal{L}_{\mathcal{N}}$. So $\mathcal{L}_f \subsetneq \mathcal{L}_{\mathcal{N}}$ and we are done.
- If $\max\{m \mid b^m \in \mathcal{L}_{\mathcal{N}}^\perp(t)\} = \infty$. Consider m the occurrences of b in σ_n produced at the subtree rooted in v then there exists $m' > m$ such that one can build a covering sequence σ with $m > f(\alpha(n)) + 1$ such that $\lambda(\sigma) = a^{\alpha(n)} b^{f(\alpha(n)) + m' - m} \in \mathcal{L}_{\mathcal{N}}$. So $\mathcal{L}_f \subsetneq \mathcal{L}_{\mathcal{N}}$ and we are done.
- Otherwise (i.e. $a^+b^+ \cap \mathcal{L}_{\mathcal{N}}^\perp(t) = \emptyset$ and $\max\{m \mid b^m \in \mathcal{L}_{\mathcal{N}}^\perp(t)\} < \infty$), we replace the firing of (u, t) by the sequence below and remove all firings from $Des(v)$,

$$(u, t_-) \overbrace{(u, t_-^x) \dots (u, t_-^x)}^{y_t \text{ times}} (u, t_+)$$

and obtain a covering sequence σ'_n such that $\lambda(\sigma'_n) = a^\ell b^m$ with $\ell \leq \alpha(n)$ and $m \geq f(\alpha(n))$. If $\ell < \alpha(n)$, and $m > f(\alpha(n))$ then $\mathcal{L}_f \subsetneq \mathcal{L}_{\mathcal{N}}$ and we are done. Otherwise $\lambda'(\sigma'_n) = \lambda'(\sigma'_n)$ and $|\sigma'_n|_{ab} < |\sigma_n|_{ab}$. We can repeat this process until either one concludes that $\mathcal{L}_f \subsetneq \mathcal{L}_{\mathcal{N}}$ or there is no more firing of t with a matching cut transition in σ'_n .

If we are not yet done, we have built a sequence σ'_n with $\lambda'(\sigma'_n) = \lambda'(\sigma'_n)$ and such that $|\sigma'_n|_{ab} \leq |T_{ab}|(2D + G)$. So we choose $B = |T_{ab}|(2D + G)$.

In order that for all $a^\ell b^m \in \mathcal{L}_f$ there is a covering sequence σ with $|\sigma|_{ab} \leq B$, we build \mathcal{N}'' from \mathcal{N}' . We observe that the definition of α implies that: $\mathcal{L}_f = \{a^\ell b^m \mid \exists n, \delta^-, \delta^+ \ell = \alpha(n) + \delta^+ \wedge m = f(\alpha(n)) - \delta^-\}$. Due to the observation about RPN languages, the initial state of \mathcal{N}' consists of only one vertex, whose initial marking is denoted \mathbf{m}_{ini} . So one builds the RPN \mathcal{N}'' with initial marking \vec{p}_{ini} from \mathcal{N}' as follows.

- Add elementary transitions t_a, t_{run}, t_{ab} and places p_{ini}, p_{run} such that:

$$\begin{aligned} \mathbf{Pre}_{t_a}'' &= \vec{p}_{ini}, & \mathbf{C}_{t_a}'' &= 0, & \lambda''(t_a) &= a; \\ \mathbf{Pre}_{t_r}'' &= \vec{p}_{ini}, & \mathbf{C}_{t_r}'' &= \mathbf{m}_{ini} - \vec{p}_{ini} + \vec{p}_{run}, & \lambda''(t_r) &= \varepsilon; \\ \mathbf{Pre}_{t_{ab}}'' &= \vec{p}_{run}, & \mathbf{C}_{t_{ab}}'' &= \vec{p}_{run}, & \lambda''(t_{ab}) &= \varepsilon. \end{aligned}$$

- For all $t \in T$, we set:

$$\mathbf{Pre}_t'' = \mathbf{Pre}_t' + \vec{p}_{run}, \mathbf{C}_t'' = \mathbf{C}_t' - \vec{p}_{run}, \mathbf{Beg}_t'' = \mathbf{Beg}_t' + \vec{p}_{run} \text{ when } t \in T_{ab}$$

- For any transition $t \in T$ with $\lambda'(t) = b$, we add the transition t_ε which is a copy of t with $\lambda''(t_\varepsilon) = \varepsilon$.

Let $a^\ell b^m \in \mathcal{L}_f$. Then there exist n, δ^-, δ^+ such that $\ell = \alpha(n) + \delta^+$ and $m = f(\alpha(n)) - \delta^-$. Let σ_n be a covering sequence in \mathcal{N}' such that $\lambda'(\sigma_n) = a^{\alpha(n)} b^{f(\alpha(n))}$. We define σ a covering sequence of \mathcal{N}'' as follows.

σ starts by $(r, t_a)^{\delta^+} (r, t_r) (r, t_{ab})^{|\sigma_n|}$. Then σ is completed by $\hat{\sigma}_n$ where $\hat{\sigma}_n$ is obtained from σ_n by:

- changing δ^- occurrences of transitions with label b by their copy;
- whenever σ_n creates a new vertex v , one inserts $(v, t_{ab})^{|\sigma_n|}$ firings.

Observe that $\lambda''(\sigma) = a^\ell b^m$. Let w be a word. Define $w_{\downarrow b}$ as the set of words obtained from w by omitting some occurrences of b . Define $\mathcal{L}'' = \{w \mid \exists w' \in \mathcal{L}_{\mathcal{N}} w \in a^* w'_{\downarrow b}\}$. Therefore $\mathcal{L}_f \subseteq \mathcal{L}_B(\mathcal{N}'', \vec{p}_{ini}, S_t) \subseteq \mathcal{L}''$. Thus $\mathcal{L}_f \subsetneq \mathcal{L}''$. If $\mathcal{L}_{\mathcal{N}} = \mathcal{L}_f$ then $\mathcal{L}'' = \mathcal{L}_f$ which concludes the proof. ■

We are now in position to conclude.

Proposition 5. \mathcal{L}_f is not an RPN coverability language.

Proof. Let \mathcal{N} be an RPN with initial state s_{init} and final states S_t such that $\mathcal{L}_f \subseteq \mathcal{L}(\mathcal{N}, s_{init}, S_t)$. Let us denote more concisely $\mathcal{L}(\mathcal{N}, s_{init}, S_t)$ by $\mathcal{L}_{\mathcal{N}}$. By Proposition 4 if $\{\sigma_n\}_{n \in \mathbb{N}}$ are bounded then $\mathcal{L}_f \subsetneq \mathcal{L}_{\mathcal{N}}$ and we are done.

So assume that the depths of $\{\sigma_n\}_{n \in \mathbb{N}}$ are unbounded. Denote by $G = \max\{|s_t| \mid s_t \in S_t\}$. Let $h = (5|T_{ab}| + 1)G$. There is some $\sigma_n = \sigma'_n \sigma''_n$ such that $s_0 \xrightarrow{\sigma'_n} s'$ where the depth of s' is greater than h . There exists $\{v_i\}_{0 < i \leq 5} \subseteq V_{s'}$ on the same branch which were created by the firing of the same abstract transition with corresponding depths are $\{d_i\}$ and such that $d_{i+1} - d_i > G$. Denote by T_f the subtree of the final state of σ_n which matches the state to be covered and by Br the branch leading to it. Due to the choice of G there exist three consecutive vertices v_i, v_{i+1} and v_{i+2} such that:

- The branch Br_i between v_i and v_{i+2} does not intersect with T_f ;
- Either Br_i does not intersect with Br or Br_i is included in Br .

Each of these vertex either may or may not have a matching cut in σ_n . We pick two of these vertices v, v' ($v' \in Des(v)$) such that either both of them have a matching cut or both of them do not. Denote by w and w' the labellings of the sequences performed in the subtree rooted in v and v' along σ_n , we split the proof in two cases:

- **Case** $w \neq w'$. We denote by $\hat{w} \neq \epsilon$ the trace of the sequence performed in the subtree rooted in v without the one performed in the subtree rooted in v' :
 - $\hat{w} = a^\ell$, for $\ell > 0$. Then one can build another covering sequence by mimicking the behavior of v' from v . But then the trace of the new covering sequence will be $a^{\alpha(n)-\ell} b^{f(\alpha(n))}$ and since $\ell > 0$ we get that $a^{\alpha(n)k-\ell} b^{f(\alpha(n))} \notin \mathcal{L}_f$, from which we conclude that $\mathcal{L}_f \subsetneq \mathcal{L}_{\mathcal{N}}$.
 - $\hat{w} = b^m$, for $m > 0$. Then one can build another covering sequence by mimicking the behavior of v from v' . But then the trace of the new covering sequence will be $a^{\alpha(n)} b^{f(\alpha(n))+m} \notin \mathcal{L}_f$ from which we conclude that $\mathcal{L}_f \subsetneq \mathcal{L}_{\mathcal{N}}$.
 - $\hat{w} = a^\ell b^m$, for $\ell, m > 0$. Then one can build a family of covering sequences $\{\hat{\sigma}_x\}_{x \in \mathbb{N}}$ by mimicking the behavior of v_i from v_{i+1} recursively x times. We would get that $\lambda(\hat{\sigma}_x) = a^{\alpha(n)+x\ell} b^{f(\alpha(n))+xm}$ for any $x \in \mathbb{N}$. But that would give us that:

$$\frac{f(\alpha(n)) + xm}{\alpha(n) + x\ell} \xrightarrow{x \rightarrow \infty} \frac{m}{\ell} > 0$$

Since f is sublinear there exists $x \in \mathbb{N}$ such that $\lambda(\hat{\sigma}_x) \notin \mathcal{L}_f$, from which we conclude that $\mathcal{L}_f \subsetneq \mathcal{L}_{\mathcal{N}}$.

- **Case** $w = w'$. Then one can build another covering sequence with the trace $a^{\alpha(n)} b^{f(\alpha(n))}$ where we mimic the behavior of v' in v . By doing so we get a covering sequence σ'_n not deeper than σ_n but which is shorter than σ_n , i.e. $|\sigma_n| \geq |\sigma'_n|$ which is a contradicts to our assumption about σ_n . ■

4 Decidability of the coverability problem

The *coverability problem* takes as input a DRPN \mathcal{N} , and two states s_0, s and asks whether there exists a sequence $s_0 \xrightarrow{\sigma} s' \succeq s$. Before developing the proof that the coverability problem is decidable let us describe its scheme.

- The algorithm builds a DRPN $\widehat{\mathcal{N}}$ by adding elementary transitions to \mathcal{N} (Definition 6). The DRPN $\widehat{\mathcal{N}}$ is equivalent to \mathcal{N} w.r.t. coverability. Furthermore for any firing sequence σ of \mathcal{N} , there is an equivalent firing sequence $\widehat{\sigma}$ of $\widehat{\mathcal{N}}$ such that in $\widehat{\sigma}$ there is no occurrence of an abstract transition followed later by a matching cut step.
- The definition of $\widehat{\mathcal{N}}$, is based on several upward closed sets of \mathbb{N}^P : (1) $Endable(\mathcal{N})$, the set of markings \mathbf{m} such that from $s_{\mathbf{m}}$ one can reach \perp and (2) for all $t \in T_{ab}$, $Closed_t$ the set of markings from which one can fire t and create a vertex whose marking belongs to $Endable(\mathcal{N})$. So we establish that one can compute these sets (Proposition 6).
- Afterwards we successively define and solve two intermediate particular coverability problems: (1) the *restricted rooted coverability problem* which takes as input a DRPN \mathcal{N} , a marking \mathbf{m}_0 , and a state s and asks whether there exists a sequence $s_{\mathbf{m}_0} \xrightarrow{\sigma} s' \mathbf{Id} \succeq s$ and (2) the *restricted coverability problem* which takes as input a DRPN \mathcal{N} , a marking \mathbf{m}_0 , and a state s and asks whether there exists a sequence $s_{\mathbf{m}_0} \xrightarrow{\sigma} s' \succeq s$. The decidability of the latter problem (Theorem 2) is partially based on the decidability of the former one (Theorem 1).
- Finally we solve the coverability problem (Theorem 3) by a case based analysis of the covering sequence which, depending on the case, is based on either Theorem 1 or Theorem 2.

As announced above, building the following sets is a key ingredient for the decidability of the coverability problem. Due to the properties of \preceq , these sets are upward closed.

Definition 5. Let \mathcal{N} be a DRPN. Then $Endable(\mathcal{N}) \subseteq \mathbb{N}^P$ is defined by:

$$Endable(\mathcal{N}) = \{\mathbf{m} \mid \exists \sigma \ s_{\mathbf{m}} \xrightarrow{\sigma} \perp\}$$

and for all $t \in T_{ab}$, $Closed_t$ is defined by:

$$Closed_t = Beg_t^{-1}(Endable(\mathcal{N}))$$

Example 2. Consider the DRPN of Figure 1. With one token in p_{time} and in p_{adv} one fires t_{found} producing a token in p_{dead} which allows to fire a cut transition. Furthermore firing the abstract transition t_{hire} does not help to reach \perp since in the new vertex, the marking of p_{time} is equal to the marking of p_{time} in its parent vertex and the marking p_{adv} is smaller than the marking of p_{adv} in its parent vertex. Thus $Endable(\mathcal{N}) = \{p_{time} + p_{adv}\}^\uparrow$. The marking of a vertex created by t_{hire} is greater or equal than $p_{time} + p_{adv}$ if in its parent vertex there is at least one token in p_{time} and three tokens in p_{adv} . Thus $Closed_{t_{hire}} = \{p_{time} + 3p_{adv}\}^\uparrow$.

If we are able to compute $Endable$ and thus $\{Closed_t\}_{t \in T_{ab}}$ then we can build a DRPN $\widehat{\mathcal{N}}$ which in some sense is equivalent to \mathcal{N} . The interest of $\widehat{\mathcal{N}}$ is that

for any firing sequence σ of \mathcal{N} , there is an equivalent firing sequence $\widehat{\sigma}$ where the firing of an abstract transition t followed later by a matching cut transition can be replaced by the firing of an elementary t^- followed later by the firing of another elementary transition t^+ . So the set of transitions is extended with $T_r = \{t^-, t^+ \mid t \in T_{ab}\}$. To ensure the sequentiality between these firings, the set of places is extended with $P_r = \{p_t \mid t \in T_{ab}\}$ with one token produced (resp. consumed) in p_t by t^- (resp. t^+). To ensure that the firing of t^- is performed when the corresponding firing of t can be matched later by the firing of the cut transition, the guard of t^- is the guard of t intersected with $Closed_t$. In order to formally define $\widehat{\mathcal{N}}$ and to exhibit relations between states and thus markings of \mathcal{N} and $\widehat{\mathcal{N}}$, we introduce the projection Proj from $\mathbb{N}^{\widehat{P}}$ to \mathbb{N}^P where $\widehat{P} = P \cup P_r$. In addition $\widehat{\mathcal{N}}_{el}$, obtained from $\widehat{\mathcal{N}}$ by deleting T_{ab} , allows to track the evolution of the marking of a vertex in $\widehat{\mathcal{N}}$ when no firing of abstract transitions occurs in this vertex.

Definition 6. *Let \mathcal{N} be a DRPN.*

Then $\widehat{\mathcal{N}} = \langle \widehat{P}, \widehat{T}, \widehat{Grd}, \widehat{Upd}, \widehat{Upd}^-, \widehat{Upd}^+, \widehat{Beg}, \widehat{End} \rangle$ is a DPRN defined by:

- $\widehat{P} = P \cup P_r$ and $\widehat{T} = T \cup T_r$;
- for all $t \in T$, $\widehat{Grd}_t = \text{Proj}^{-1}(Grd_t)$
for all $t^- \in T_r$, $\widehat{Grd}_{t^-} = \text{Proj}^{-1}(Grd_t \cap Closed_t)$
for all $t^+ \in T_r$, $\widehat{Grd}_{t^+} = p_t > 0$;
- for all $t \in T_{el}$, all $p \in P$ and all $p_{t'} \in P_r$,
 $\widehat{Upd}_t(p) = Upd_t(p) \circ \text{Proj}$ and $\widehat{Upd}_t(p_{t'}) = p_{t'}$;
- for all $t^- \in T_r$, all $p \in P$ and all $p_{t'} \in P_r$,
 $\widehat{Upd}_{t^-}(p) = Upd_{t^-}(p) \circ \text{Proj}$ and $\widehat{Upd}_{t^-}(p_{t'}) = p_{t'} + \mathbf{1}_{t=t'}$;
- for all $t^+ \in T_r$, all $p \in P$ and all $p_{t'} \in P_r$,
 $\widehat{Upd}_{t^+}(p) = Upd_{t^+}(p) \circ \text{Proj}$ and $\widehat{Upd}_{t^+}(p_{t'}) = p_{t'} - \mathbf{1}_{t=t'}$;
- for all $t \in T_{ab}$, all $p \in P$ and all $p_{t'} \in P_r$,
 1. $\widehat{Upd}_t^-(p) = Upd_t^-(p) \circ \text{Proj}$, $\widehat{Upd}_t^+(p) = Upd_t^+(p) \circ \text{Proj}$,
 $\widehat{Beg}_t(p) = Beg_t(p) \circ \text{Proj}$;
 2. $\widehat{Upd}_t^-(p_{t'}) = p_{t'}$, $\widehat{Upd}_t^+(p_{t'}) = p_{t'}$, $\widehat{Beg}_t(p_{t'}) = 0$;
- $\widehat{End} = \text{Proj}^{-1}(End)$;

$\widehat{\mathcal{N}}_{el}$ is obtained from $\widehat{\mathcal{N}}$ by deleting T_{ab} .

The following lemma states the correspondence between \mathcal{N} , $\widehat{\mathcal{N}}$ and $\widehat{\mathcal{N}}_{el}$. We extend Proj to states of $\widehat{\mathcal{N}}$ by applying it to the marking of vertices and furthermore to sets of states by the standard set extension. In the reverse direction, given a marking $\mathbf{m} \in \mathbb{N}^P$ we define $\widehat{\mathbf{m}} \in \mathbb{N}^{\widehat{P}}$ the extended marking with no token in P_r . Similarly, given s a state of \mathcal{N} , we define \widehat{s} a state of $\widehat{\mathcal{N}}$ by extending the marking of vertices of s with no token in P_r .

Lemma 1. *Let \mathcal{N} be a DRPN and s_0 be a state. Then:*

1. For all $\widehat{s}_0 \xrightarrow{\sigma} \widehat{\mathcal{N}} s$, there exists $s_0 \xrightarrow{\sigma'} \mathcal{N} s'$ with $\text{Proj}(s) \preceq_{\text{Id}} s'$
and for all $s_0 \xrightarrow{\sigma'} \mathcal{N} s$, $\widehat{s}_0 \xrightarrow{\sigma} \widehat{\mathcal{N}} \widehat{s}$;

2. $Cov(\mathcal{N}, s_0) = \text{Proj}(Cov(\widehat{\mathcal{N}}, \widehat{s}_0))$;
3. For all $s \in \text{Reach}(\widehat{\mathcal{N}}, \widehat{s}_0)$, there exists a sequence $\widehat{s}_0 \xrightarrow{\sigma}_{\widehat{\mathcal{N}}} s$ such that no firing of abstract transition is matched by a cut transition.
4. $\text{Proj}(\text{Endable}(\widehat{\mathcal{N}}) \cap \bigcap_{t \in T_{ab}} p_t = 0) = \text{Proj}(\text{Endable}(\widehat{\mathcal{N}}_{el}) \cap \bigcap_{t \in T_{ab}} p_t = 0)$
 $= \text{Endable}(\mathcal{N})$.

Proof.

• Let $\widehat{s}_0 \xrightarrow{\sigma}_{\widehat{\mathcal{N}}} s$. Consider successively all $t \in T_{ab}$. Observe that due to the presence of place p_t every occurrence of (v, t^+) in σ' can be matched with an occurrence of (v, t^-) . The unmatched occurrences of (v, t^-) can be omitted since they only produce useless tokens in p_t and do not increase the marking of any other place. So we get a new firing sequence $\widehat{s}_0 \xrightarrow{\sigma^*}_{\widehat{\mathcal{N}}} s^*$ with $\text{Proj}(s) \preceq_{\text{Id}} \text{Proj}(s^*)$.

We transform this sequence in a sequence $s_0 \xrightarrow{\sigma'}_{\mathcal{N}} \text{Proj}(s^*)$ as follows. For every pair of matching firings $(v, t^-), (v, t^+)$, we substitute to (v, t^-) the firing (v, t) creating the vertex w with an initial $\mathbf{m} \in \text{Endable}(\mathcal{N})$ due to the guard of t . Then we substitute to (v, t^+) a sequence $s_{\mathbf{m}} \xrightarrow{\sigma_{\mathbf{m}}} \perp$ applying it to w . The “reverse” direction is immediate since for all $t \in T \cup \{\tau\}$ and all states s and s' , $s \xrightarrow{(v,t)}_{\mathcal{N}} s'$ implies $\widehat{s} \xrightarrow{(v,t)}_{\widehat{\mathcal{N}}} \widehat{s}'$.

• Assertion 2 is an immediate consequence of Assertion 1 of the lemma.

• Let $\widehat{s}_0 \xrightarrow{\sigma'} s$ be a firing sequence of $\widehat{\mathcal{N}}$. Consider successively all $t \in T_{ab}$ and all matching pairs $(v, t), (w, \tau)$ occurring in σ' where w is the vertex created by the firing of (v, t) . Let σ^* be the subsequence of σ of firings in the subtree rooted at w ended by (w, τ) . Then one substitutes to (v, t) , the firing (v, t^-) which is fireable as witnessed by σ^* and one deletes σ^* substituting (w, τ) by (v, t^+) . Iterating this process, one gets the sequence we are looking for.

• Let $\mathbf{m} \in \text{Endable}(\widehat{\mathcal{N}}) \cap \bigcap_{t \in T_{ab}} p_t = 0$. Consider a sequence $s_{\mathbf{m}} \xrightarrow{\sigma}_{\widehat{\mathcal{N}}} \perp$. Using Assertion 3, one can assume that in σ , no firing of an abstract transition is matched with a cut transition. Consider (r, t) the firing of an abstract transition in the root occurring in σ . Since it is not matched by a cut transition one can delete it and all the firings in the subtree rooted at the created vertex and still reaches \perp . Such a sequence is thus a firing sequence of $\widehat{\mathcal{N}}_{el}$. The other direction is immediate since the set of transitions of $\widehat{\mathcal{N}}_{el}$ is included in the one of $\widehat{\mathcal{N}}$.

The inclusion $\text{Endable}(\mathcal{N}) \subseteq \text{Proj}(\text{Endable}(\widehat{\mathcal{N}}) \cap \bigcap_{t \in T_{ab}} p_t = 0)$ is immediate since for all $t \in T \cup \{\tau\}$, $s \xrightarrow{(v,t)}_{\mathcal{N}} s'$ implies $\widehat{s} \xrightarrow{(v,t)}_{\widehat{\mathcal{N}}} \widehat{s}'$.

Let $\mathbf{m} \in \text{Proj}(\text{Endable}(\widehat{\mathcal{N}}) \cap \bigcap_{t \in T_{ab}} p_t = 0)$. One applies Assertion 1 of the lemma with $s_0 = s_{\mathbf{m}}$. Let $\widehat{s}_{\mathbf{m}} \xrightarrow{\sigma}_{\widehat{\mathcal{N}}} s \xrightarrow{(r,\tau)}_{\widehat{\mathcal{N}}} \perp$. Then there exists $s_0 \xrightarrow{\sigma}_{\mathcal{N}} s'$ with $\text{Proj}(s) \preceq_{\text{Id}} s'$. Thus $s' \xrightarrow{(r,\tau)}_{\mathcal{N}} \perp$ which establishes that $\mathbf{m} \in \text{Endable}(\mathcal{N})$. ■

Let us describe how Algorithm 1 computes *Endable* and $\{\text{Closed}_t\}_{t \in T_{ab}}$. During lines 2-14, it builds a version of $\widehat{\mathcal{N}}_{el}$ where for every t , Closed_t is replaced by $\text{Closed}[t]$. Since $\text{Closed}[t]$ will be updated during the loop of lines 15-23, the definition of Grd_t^- is performed at line 16. Still in this loop, using a standard backward exploration, during lines 17-20, it computes in variable X , *Endable* for this version of $\widehat{\mathcal{N}}_{el}$. Afterwards still in this loop, it updates Y by restricting X

Algorithm 1: Computing the closure of abstract transitions

```

Closing( $\mathcal{N}$ )
Input:  $\mathcal{N}$  a DRPN
/*  $\widehat{P} = P \cup \{p_t \mid t \in T_{ab}\}$ ,  $\widehat{T}_{el} = T_{el} \cup \{t^-, t^+ \mid t \in T_{ab}\}$  */
/* Proj is the projection from  $\mathbb{N}^{\widehat{P}}$  to  $\mathbb{N}^P$  */
Data:  $X, oldX$  subsets of  $\mathbb{N}^{\widehat{P}}$ ;  $oldY, Y$  subset of  $\mathbb{N}^P$ ;  $t$  a transition
Output: Closed an array indexed by  $T_{ab}$  of upward closed sets of  $\mathbb{N}^P$ 
1  $Y \leftarrow \emptyset$ ; for  $t \in T_{ab}$  do  $Closed[t] \leftarrow \emptyset$ 
2 for  $t \in T_{el}$  do
3    $Grd_t \leftarrow Proj^{-1}(Grd_t)$ 
4   for  $p \in P$  do  $Upd_{t-}(p) \leftarrow Upd_t(p) \circ Proj$ 
5   for  $p_{t'} \in \widehat{P}$  do  $Upd_t(p_{t'}) \leftarrow p_{t'}$ 
6 end
7 for  $t \in T_{ab}$  do
8   for  $p \in P$  do  $Upd_{t-}(p) \leftarrow Upd_t^-(p) \circ Proj$ 
9   for  $p_{t'} \in \widehat{P}$  with  $t' \neq t$  do  $Upd_{t-}(p_{t'}) \leftarrow p_{t'}$ 
10   $Upd_{t-}(p_t) \leftarrow p_t + 1$ ;  $Grd_{t+} \leftarrow p_t > 0$ 
11  for  $p \in P$  do  $Upd_{t+}(p) \leftarrow Upd_t^+(p) \circ Proj$ 
12  for  $p_{t'} \in \widehat{P}$  with  $t' \neq t$  do  $Upd_{t+}(p_{t'}) \leftarrow p_{t'}$ 
13   $Upd_{t+}(p_t) \leftarrow p_t - 1$ 
14 end
15 repeat
16   for  $t \in T_{ab}$  do  $Grd_{t-} \leftarrow Proj^{-1}(Grd_t \cap Closed[t])$ 
17    $oldY \leftarrow Y$ ;  $X \leftarrow Proj^{-1}(End)$ ;  $oldX \leftarrow \emptyset$ 
18   while  $X \neq oldX$  do
19      $oldX \leftarrow X$ ; for  $t \in \widehat{T}_{el}$  do  $X \leftarrow X \cup (Upd_t^{-1}(X) \cap Grd_t)$ 
20   end
21    $Y \leftarrow Proj(X \cap \bigcap_{t \in T_{ab}} p_t = 0)$ 
22   for  $t \in T_{ab}$  do  $Closed[t] \leftarrow Beg_t^{-1}(Y)$ 
23 until  $Y \neq oldY$ 
24 return  $Y, Closed$ 

```

to the markings with no token in P_r and then projecting it to P . Then using Y , updates for every t , $Closed[t]$. The algorithm terminates when Y is no more enlarged.

Proposition 6. *Algorithm 1 terminates and upon termination $Y = Endable$ and for all $t \in T_{ab}$, $Closed[t] = Closed_t$.*

Proof. In the sequel $Closed[t]$ denotes the value of this variable at some execution point. Let us denote \mathcal{N}' the version of \widehat{N}_{el} built by the algorithm and updated at every iteration of loop of lines 15-23.

• **Termination.** We prove by induction that the sequence of sets Y_n and for all t , $Closed_n[t]$ at the beginning of iteration n of the **repeat** loop is an increasing sequence of upward closed sets of \mathbb{N}^P and $\mathbb{N}^{\widehat{P}}$, respectively. So it must stabilize after a finite number of iterations. Since T is finite this will establish termination of the algorithm. First, for all t , $\emptyset = Closed_1[t] \subseteq Closed_2[t]$. Assume that for some $1 < n$, for all t , $Closed_{n-1}[t] \subseteq Closed_n[t]$. Then the n^{th} iteration corresponds to the $n - 1^{th}$ iteration with $Closed_n[t]$ substituted to $Closed_{n-1}[t]$. Since the operations involving $Closed[t]$ are intersection, union and projection, this immediately entails that $Closed_n[t] \subseteq Closed_{n+1}[t]$.

It remains to prove that for all n , Y_n and (for all t) $Closed_n[t]$ are upward closed and that the **while** loop terminates. We also prove it by induction on n . Consider the n^{th} iteration of the **repeat** loop and let us prove the sequence of sets X_k at the beginning of iteration k of the **while** loop is an increasing sequence of upward closed sets. This will establish the termination of this loop. Observe that $X_1 = \text{Proj}^{-1}(End)$ is an upward closed set and that at every iteration X is increased because it is updated by union of some set with itself. Furthermore, X remains an upward closed set since (1) upward closed sets are closed by union, intersection, and inverse of non decreasing mappings. Finally while $X \cap \bigcap_{t \in T_{ab}} p_t = 0$ is not upward closed, this is the case for $Y_{n+1} = \text{Proj}(X \cap \bigcap_{t \in T_{ab}} p_t = 0)$. Thus for every t , $Closed_{n+1}[t]$ is upward closed.

• **Consistency.** We establish by induction on the iterations of the **repeat** loop that $Y \subseteq Endable(\mathcal{N})$ and for all $\mathbf{m} \in Closed[t]$, there is a sequence $s_{Beg(t)(\mathbf{m})} \xrightarrow{\sigma_{\mathbf{m}}} \perp$, implying $Closed[t] \subseteq Closed_t$. Consider an arbitrary iteration of the **repeat** loop. Thus the **while** loop computes X the set $Endable(\mathcal{N}')$. Since by induction, $Closed[t] \subseteq Closed_t$ one deduces that $X \subseteq Endable(\widehat{N}_{el})$. Applying Assertion 4 of Lemma 1, one deduces that $Y \subseteq Endable(\mathcal{N})$ and so that at the end of the iteration $Closed[t] \subseteq Closed_t$.

• **Completeness.** Let $\mathbf{m} \in Closed_t$. Consider a sequence $s_{Beg(t)(\mathbf{m})} \xrightarrow{\sigma} \mathcal{N} \perp$. Observe that if in σ , there exists a firing of an abstract transition creating some vertex v not later followed by a cut transition in v , then one can omit this firing and all firings in the subtree rooted at v and still reaches \perp . Thus we assume that every vertex v created by the firing of abstract transition is later deleted by a matching cut transition in v .

We establish the completeness of the algorithm by recurrence on the depth of σ . If the depth is null, it means that σ only includes firing of elementary transitions in r ended by the cut transition. So $\widehat{s}_{Beg(t)(\mathbf{m})} \xrightarrow{\sigma} \mathcal{N}_{el} \perp$. Furthermore since

$\sigma \in (\{r\} \times T_{el})^*(r, \tau)$, $\widehat{s}_{Beg(t)(\mathbf{m})} \xrightarrow{\sigma}_{\mathcal{N}'} \perp$ for \mathcal{N}' built at the beginning of the first iteration of the **repeat** loop. During every iteration of the **repeat** loop, the **while** loop computes in X the set of markings from which a sequence of transitions of \mathcal{N}' leads to some marking in $\text{Proj}^{-1}(\text{End})$. So $\widehat{Beg}_t(\mathbf{m}) \in X$ at the end of the iteration and after the **for** loop at line 22, $\mathbf{m} \in \text{Closed}[t]$.

Assume that σ has depth $h > 0$. So every \mathbf{m}' in the root from which there is a firing of an abstract transition t' belongs to $\text{Closed}[t']$ since the subsequence in the created vertex up to the cut transition has depth strictly less than h . Consider the last iteration of the **repeat** loop for which such t' is added to $\text{Closed}[t']$. Then either at this iteration \mathbf{m} already belongs to $\text{Closed}[t]$ or it will be added at the next iteration (which exists since Y is enlarged) due to execution of the **while** loop. Indeed consider a closing subsequence of σ for a child of the root created by some transition t' and substitute the firing of t' by t'^- , delete the closing subsequence and substitute the cut step by the firing of t^+ in r . Doing this transformation (and omitting the cut step in r) one obtains a closing firing sequence in \mathcal{N}' as described above from $\widehat{s}_{Beg(t)(\mathbf{m})}$.

Thus at the beginning of last iteration of the **while** loop, $\mathcal{N}' = \widehat{\mathcal{N}}_{el}$. Using Assertion 4 of Lemma 1, one gets that at this end of this iteration, $Y = \text{Endable}(\mathcal{N})$. \blacksquare

Theorem 1. *The restricted rooted coverability problem is decidable.*

Proof. Due to Assertion 1 of Lemma 1, we consider $\widehat{\mathcal{N}}$. Applying Assertion 3 of Lemma 1, the sequence we are looking for, does not create other vertices than the vertices of s since (1) the firing of abstract transitions matched by cut transitions are non necessary in $\widehat{\mathcal{N}}$ and (2) those that are not matched decrease the marking of a vertex and create a subtree useless for covering $s = (V, E, M, A)$.

Then for any vertex $v \in V$, one guesses an order of creation of its children along the sequence σ and for any transition t labelling an edge of E , one guesses a transition t' with $\text{Upd}_{t'}^+ \geq \text{Upd}_t^+$. Observe that there are only a finite number of such guesses and so the algorithm enumerates them.

Afterwards the algorithm proceeds bottom-up from the leaves of s to the root. Let v be a leaf. Then the algorithm computes by backward exploration the upward closed set $\text{Cov}(v) = \{\mathbf{m} \in \mathbb{N}^{\widehat{P}} \mid \exists \mathbf{m}' \geq M(v) \exists \sigma \in \widehat{T}_{el}^* \mathbf{m} \xrightarrow{\sigma} \mathbf{m}'\}$.

Let v be an internal vertex with v_1, \dots, v_n its children enumerated in the guessed order and t_1, \dots, t_n the associated guessed transitions. Then the algorithm computes by backward exploration the upward closed sets $\text{Cov}_n(v), \dots, \text{Cov}_0(v) = \text{Cov}(v)$ as follows: $\text{Cov}_n(v) = \{\mathbf{m} \in \mathbb{N}^{\widehat{P}} \mid \exists \mathbf{m}' \geq M(v) \exists \sigma \in \widehat{T}_{el}^* \mathbf{m} \xrightarrow{\sigma} \mathbf{m}'\}$ and for $i < n$,

$$\begin{aligned} \text{Cov}_i(v) = \{ \mathbf{m} \in \mathbb{N}^{\widehat{P}} \mid \\ \exists \mathbf{m}' \in (\widehat{\text{Upd}}_{t_{i+1}}^-)^{-1} \left(\widehat{\text{Beg}}_{t_{i+1}}^{-1}(\text{Cov}(v_{i+1})) \cap \text{Cov}_{i+1}(v) \right) \cap \widehat{\text{Grd}}_{t_{i+1}} \\ \exists \sigma \in \widehat{T}_{el}^* \mathbf{m} \xrightarrow{\sigma} \mathbf{m}' \} \end{aligned}$$

By a straightforward induction, one establishes that $\mathbf{m} \in Cov_i(v)$ if and only from vertex v marked by \mathbf{m} there is a firing sequence $\sigma = \sigma_i t_{i+1} \sigma_{i+1} \dots t_n \sigma_n$ with for all j , $\sigma_j \in \widehat{T}_{el}^*$ and such that (1) the marking of v reached by σ is greater or equal than $M(v)$, and (2) for all $j > i$, from the initial marking of v_j one can fire a sequence that builds a tree covering the subtree of s rooted at v_i using the guessed transitions and orders of creation. Finally the algorithm returns true if and only if for some guess $\widehat{\mathbf{m}}_0 \in Cov(r)$ where r is the root of s . ■

Theorem 2. *The restricted coverability problem is decidable.*

Proof. Let us fix some net \mathcal{N} and some state s . As above we substitute $\widehat{\mathcal{N}}$ to \mathcal{N} but for sake of readability we omit the occurrences of ‘ $\widehat{\cdot}$ ’. Thus one observes that the state s' that should cover s can be chosen as a single branch, say Br , leading to a tree s^* isomorphic to s . Indeed the firings that would create other branches are useless since they only decrease the markings in Br or in s' .

Observe that in the previous proof, instead of answering the decision problem, one can compute the set $RRC(\mathcal{N}, s) = \{\mathbf{m} \mid \exists s_{\mathbf{m}} \xrightarrow{\sigma} s^* \mathbf{id} \succeq s\}$.

Let us define $RC(\mathcal{N}, s, k) = \{\mathbf{m} \mid \exists s_{\mathbf{m}} \xrightarrow{\sigma} s'_{\varphi} \succeq s \wedge |r \rightarrow_{s'} \varphi(r)| \leq k\}$ where $|x \rightarrow_{s'} y|$ denotes the length of the elementary path from x to y in s' . One immediately observes that $RC(\mathcal{N}, s, 0) = RRC(\mathcal{N}, s)$ that for all k , $RC(\mathcal{N}, s, k)$ is upward closed and $RC(\mathcal{N}, s, k) \subseteq RC(\mathcal{N}, s, k+1)$. Furthermore the answer to the restricted coverability problem is positive if and only if $\mathbf{m}_0 \in \bigcup_{k \in \mathbb{N}} RC(\mathcal{N}, s, k)$. So it only remains to show how to compute $RC(\mathcal{N}, s, k+1)$ when one knows $RC(\mathcal{N}, s, k)$. Observe that:

$$\begin{aligned} RC(\mathcal{N}, s, k+1) = & RC(\mathcal{N}, s, k) \cup \{\mathbf{m} \mid \\ & \exists \sigma \in \widehat{T}_{el}^* \exists \mathbf{m}' \in \bigcup_{t \in T_{ab}} Grd_t \cap Beg_t^{-1}(RC(\mathcal{N}, s, k)) \\ & \mathbf{m} \xrightarrow{\sigma} \mathbf{m}'\} \end{aligned}$$

where the second term of this union can be computed by a backward exploration. ■

Using the previous theorems, we are now in position to decide the coverability problem.

Theorem 3. *The coverability problem of DRPN is decidable.*

Proof. Let us fix some net \mathcal{N} and states s_0 and s . As above we substitute $\widehat{\mathcal{N}}$ to \mathcal{N} and omit the occurrences of ‘ $\widehat{\cdot}$ ’. In order to decide the existence of a sequence $s_0 \xrightarrow{\sigma} s'_{\varphi} \succeq s$, we consider two cases depending on $\varphi(r)$

- $|\varphi(V_s) \cap V_{s_0}| \leq 1$. So one guesses a vertex $w \in V_{s_0}$ that is the deepest on the branch leading from r to $\varphi(r)$ in s' . Then we transform the net (whose current version is in the sequel denoted \mathcal{N}') as follows. We examine bottom-up all the (proper) descendants of w in s_0 as follows. Let v be such a vertex, u its parent and t the abstract transition labelling the edge (u, v) . If $M(v) \in Endable(\mathcal{N}')$ then one adds a place $p_{u,v}$ with a token in u and no token elsewhere and an elementary

transition $t_{u,v}$ such that $Grd_{t_{u,v}} = p_{u,v} > 0$ and $Upd_{t_{u,v}} = Upd_t^+ - p_{u,v}$. By construction, there exists a sequence $s_0 \xrightarrow{\sigma}_{\mathcal{N}} s'_{\varphi} \succeq s$, if and only if there exists a sequence $s_{M(w)} \xrightarrow{\sigma}_{\mathcal{N}'} s'_{\varphi} \succeq s$. So this case is decidable by Theorem 2.

- $|\varphi(V_s) \cap V_{s_0}| > 1$. Then one builds a state s'_0 with root $\varphi(r)$ and simultaneously transform the net (whose current version is also denoted \mathcal{N}') as follows. We eliminate all the vertices of s_0 which are not descendants of $\varphi(r)$ since there are irrelevant due to the choice of φ . Then we eliminate bottom-up all the descendants of $\varphi(r)$ in s_0 which do not belong to $\varphi(V_s)$ as follows. Let v be such a vertex, u its parent and t the abstract transition labelling the edge (u, v) . When one examines v , it has become a leaf. If $M(v) \in Endable(\mathcal{N}')$ then one adds a place $p_{u,v}$ with a token in u and no token elsewhere and an elementary transition $t_{u,v}$ such that $Grd_{t_{u,v}} = p_{u,v} > 0$ and $Upd_{t_{u,v}} = Upd_t^+ - p_{u,v}$. Afterwards one deletes v and (u, v) . By construction, there exists a sequence $s_0 \xrightarrow{\sigma}_{\mathcal{N}} s'_{\varphi} \succeq s$, if and only if there exists a sequence $s'_0 \xrightarrow{\sigma}_{\mathcal{N}'} s'_{\varphi} \succeq s$.

For all vertices u of s'_0 denote s_u the subtree of s rooted at $\varphi(u)$ consisting of vertices whose deepest ancestor in s is $\varphi(u)$. Then there exists a sequence $s'_0 \xrightarrow{\sigma}_{\mathcal{N}'} s'_{\varphi} \succeq s$ if and only if the following conditions hold:

- For all $(u, v) \in E_{s'_0}$, $Upd^+(\Lambda(u, v)) \geq Upd^+(\Lambda(\varphi^{-1}(u), \varphi^{-1}(v)))$;
- For all $u \in V_{s'_0}$ there exists a sequence $s_{M(u)} \xrightarrow{\sigma_u}_{\mathcal{N}'} s'_{\mathbf{Id}} \succeq s_u$.

The first item is decidable by effectiveness of \mathcal{N} . The second item is decidable by Theorem 1.

Since there are only a finite number of possible φ (more precisely their restriction over s_0) the coverability problem is decidable. ■

5 Conclusion

We have introduced dynamic recursive Petri nets that extend recursive Petri nets in several directions. We have shown that w.r.t. coverability languages, this extension is strict and we have established that the coverability problem is still decidable.

On the one hand, we plan to define a restriction of DRPN for which reachability would be still decidable as in RPN. On the other hand since our algorithm is based on backward explorations and our team has already developed an efficient tool for coverability in PN based on such explorations [1], we want to adapt it for DRPN.

References

1. Michael Blondin, Alain Finkel, Christoph Haase, and Serge Haddad. The logical view on continuous Petri nets. *ACM Trans. Comput. Log.*, 18(3):24:1–24:28, 2017.
2. Catherine Dufourd, Alain Finkel, and Philippe Schnoebelen. Reset nets between decidability and undecidability. In *ICALP'98, Aalborg, Denmark*, volume 1443 of *Lecture Notes in Computer Science*, pages 103–115. Springer, 1998.

3. Amal El Fallah Seghrouchni and Serge Haddad. A recursive model for distributed planning. In *ICMAS 1996, Kyoto, Japan*, pages 307–314, 1996.
4. Alain Finkel, Serge Haddad, and Igor Khmelnitsky. Coverability and termination in recursive Petri nets. In *Proceedings of Petri Nets*, volume 11522 of *LNCS*, pages 429–448, 2019.
5. Alain Finkel, Serge Haddad, and Igor Khmelnitsky. Coverability and termination in recursive Petri nets. In <https://hal.inria.fr/hal-02081019>, 2019.
6. Alain Finkel, Pierre McKenzie, and Claudine Picaronny. A well-structured framework for analysing Petri net extensions. *Information and Computation*, 195:1 – 29, 2004.
7. Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, 256(1-2):63–92, 2001.
8. 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.
9. Serge Haddad and Denis Poitrenaud. Modelling and Analyzing Systems with Recursive Petri Nets. In *WODES 2000, Ghent, Belgium*, pages 449–458. Kluwer Academic Publishers, 2000.
10. Serge Haddad and Denis Poitrenaud. Recursive Petri nets. *Acta Informatica*, 44(7-8):463–508, 2007.
11. Jérôme Leroux and Philippe Schnoebelen. On functions weakly computable by Petri nets and vector addition systems. In *RP 2014, Oxford, UK*, volume 8762 of *LNCS*, pages 190–202, 2014.
12. Irina A. Lomazova. Nested petri nets: Multi-level and recursive systems. *Fundam. Inform.*, 47(3-4):283–293, 2001.
13. Richard Mayr. Process rewrite systems. *Inf. Comput.*, 156(1-2):264–286, 2000.
14. Rüdiger Valk. Self-modifying nets, a natural extension of Petri nets. In *ICALP'78, Udine, Italy*, volume 62 of *LNCS*, pages 464–476. Springer, 1978.

6 Appendix

6.1 Proofs of Section 3

We show how to build a Petri net with same coverability language as a B -bounded coverability language of an RPN. Let \mathcal{N} be an RPN with the abstract transitions T_{ab} . For any $t, t' \in T_{ab}$ we say that $t \leq t'$ if $\mathbf{Post}_t \leq \mathbf{Post}_{t'}$. This is a quasi order on the abstract transitions. For any $t \in T_{ab}$ denote by $O_t \in \{0, 1\}^{|T_{ab}|}$ where:

$$O_t(t') = \begin{cases} 1 & t \geq t' \\ 0 & \text{else} \end{cases}$$

$t \geq t'$ if and only if $O_t \geq O_{t'}$.

In the following, in order to simplify the notation, we use for Petri nets the notation $\mathbf{Post}(t)$ instead of \mathbf{Post}_t (similarly for \mathbf{C} and \mathbf{Pre}).

Let $\mathcal{L} = \mathcal{L}_B(\mathcal{N}, s_{init}, S_t)$ a B -bounded coverability language and define a Petri net $\mathcal{N}^B = (P^B, T^B, \mathbf{Pre}^B, \mathbf{C}^B)$, with an initial marking \mathbf{m}_B , a set of target markings S_t^B and labeling function for its transitions λ^B . This Petri net will be constructed by copying the structure of the RPN in a tree like structure. Let $T^B = (E^B, V^B)$ be a rooted and directed tree with a root r_B , of depth B , for which every internal node has B children. For each node $v \in V^B$ define a set of places P^v and a set of transitions T^v . The set P^v includes four types of places: (1) p^v for every place p in the RPN \mathcal{N} , (2) p_t^v for every abstract transition t , (3) p_{sleep}^v (when marked meaning that this node “was not created yet”), (4) p_{run}^v (when marked meaning that the node is “alive”):

$$P^v = \{p^v \mid p \in P\} \sqcup \{p_t^v \mid t \in T_{ab}\} \sqcup \{p_{sleep}^v, p_{run}^v\}$$

For any node $v \in V^B$, the set T^v includes three types of transitions: (1) t^v for every elementary transition of the RPN, (2) $t^{v,w}$ for every abstract transition of the RPN and w child of v in T^B , (3) $\tau_{\mathbf{m}_e, t}^v$ for every abstract transition $t \in T_{ab}$ and every marking $\mathbf{m}_e \in \mathit{End}$ (this is our “cut transitions”).

$$T^v = \{t^v \mid t \in T_{el}\} \sqcup \{t^{v,w} \mid t \in T_{ab}, \mathit{prd}(w) = u\} \sqcup \{\tau_{\mathbf{m}_e, t}^v \mid \mathbf{m}_e \in \mathit{End}, t \in T_{ab}\}$$

note that for any leaf there are no transitions of the second type i.e.

$$T^v = \{t^v \mid t \in T_{el}\} \sqcup \{\tau_{\mathbf{m}_e, t}^v \mid \mathbf{m}_e \in \mathit{End}, t \in T_{ab}\}$$

So $P^B = \bigcup_{v \in V^B} P^v$, $T^B = \bigcup_{v \in V^B} T^v$. Given $v \in V^B$ and a marking $\mathbf{m} = \sum_{p \in P} a_p \cdot \vec{p}$, denote by $[\mathbf{m}]^v = \sum_{p \in P} a_p \cdot \vec{p}^v$. Denote by $\mathbf{m}_{\text{Anc}}(v) = \sum_{u \in \text{Anc}(v)} \vec{p}_{run}^u$ i.e. v and all its ancestor are ‘alive’, and for any $t \in T_{ab}$ by $\mathbf{Ma}_t(v) = \sum_{t' \in T_{ab}} O_t(t') \vec{p}_{t'}^v$. It remains to define \mathbf{Pre}^B and \mathbf{C}^B :

- The transition t^v only updates the marking of P^v :

$$\mathbf{Pre}^B(t^v) = \mathbf{m}_{\text{Anc}}(v) + [\mathbf{Pre}_t]^v ; \mathbf{C}^B(t^v) = [\mathbf{C}_t]^v$$

- The transition $t^{v,w}$ is fireable if t is ‘fireable’ in v and if the node w ‘was not created yet’ ($p_{sleep}^w > 0$). It puts tokens in P^w according Beg_t and in p_{run}^w making it ‘alive’, removes a token from p_{sleep}^w , adds a token in $\mathbf{Ma}_t(w)$ related to t , and updates the marking on the node v according to \mathbf{Pre}_t :

$$\begin{aligned}\mathbf{Pre}^B(t^{v,w}) &= \mathbf{m}_{\text{Anc}}(u) + [\mathbf{Pre}_t]^v + \vec{p}_{sleep}^w ; \\ \mathbf{C}^B(t^{v,w}) &= \vec{p}_{run}^w + \mathbf{Ma}_t(w) - \vec{p}_{sleep}^w + [Beg_t]^w - [\mathbf{Pre}_t]^v\end{aligned}$$

- The transition $\tau_{\mathbf{m}_e,t}^v$ represents the cut steps of v . It is fireable if v was created by a transition $t' \geq t$ and has marking greater or equal than $\mathbf{m}_e \in \text{End}$. Its firing removes the token from p_{run}^v (‘killing’ v) and puts tokens in the places of its parent according to \mathbf{Post}_t :

$$\mathbf{Pre}^B(\tau_{\mathbf{m}_e,t}^v) = \mathbf{m}_{\text{Anc}}(v) + [\mathbf{m}_e]^v + \mathbf{Ma}_t(v) ; \quad \mathbf{C}^B(\tau_{\mathbf{m}_e,t}^v) = [\mathbf{Post}_t]^{prd(v)} - \vec{p}_{run}^v$$

Given a state s of the RPN \mathcal{N} and an isomorphism ϕ from its underlining tree structure to a subtree of T^B , we denote by:

$$M(\phi, s) = \sum_{u \in V_s} [M_s(u)]^{\phi(u)} + \sum_{u \in \text{Anc}(\phi(V_s))} \vec{p}_{run}^u + \sum_{(u,v) \in E_s} \mathbf{Ma}_{A(u,v)}(\phi(v))$$

the ‘translation’ of the state to a marking of \mathcal{N}^B . Recall that s_{init} is the initial state of the marked RPN \mathcal{N} consisting of a single vertex r , and pick the tree isomorphism ϕ such that $\phi(r) = r^B$. The marking $\mathbf{m}_B = M(\phi, s_{init}) + \sum_{u \neq r^B} p_{sleep}^u$ is the initial marking of \mathcal{N}^B .

We also translate the target states. Given a state $s \in S_t$, and $T_s = (V_s, E_s)$ is its underlining tree structure. Denote by I_s the set of tree isomorphisms of T_s to a subtree in T^B . The set of target states is defined by:

$$S_t^B = \bigcup_{s \in S_t} \left(\bigcup_{\varphi \in I_s} M(\varphi, s) \right)$$

Finally, we set $\lambda^B(t^v) = \lambda(t)$, $\lambda^B(t^{v,w}) = \lambda(t)$, $\lambda^B(\tau_{\mathbf{m}_e,t'}^v) = \varepsilon$.

For any marking \mathbf{m} of \mathcal{N}^B , we define the projection of \mathbf{m} to a specific node $u \in V_B$, denoted by $[\mathbf{m}]_u$, as follows. Let:

$$\mathbf{m} = \sum_{u \in V_B} \left(a_{run}^u \cdot \vec{p}_{run}^u + a_{sleep}^u \cdot \vec{p}_{sleep}^u + \sum_{p \in P} a_p^u \cdot \vec{p}^u + \sum_{t \in T_{ab}} a_t^u \cdot \vec{p}_t^u \right)$$

Then:

$$[\mathbf{m}]_u = a_{run}^u \cdot \vec{p}_{run}^u + a_{sleep}^u \cdot \vec{p}_{sleep}^u + \sum_{p \in P} a_p^u \cdot \vec{p}^u + \sum_{t \in T_{ab}} a_t^u \cdot \vec{p}_t^u$$

Proposition 1. *Let \mathcal{L} be a B -bounded coverability language. Then \mathcal{L} is a PN coverability language.*

Proof. To conclude the proposition it would be enough to show that for any $B \in \mathbb{N}$ we have $\mathcal{L}_B(\mathcal{N}, s_{init}, S_t) = \mathcal{L}(\mathcal{N}^B, \mathbf{m}_B, S_t^B)$

On the one hand if $\omega \in \mathcal{L}_B(\mathcal{N}, s_{init}, S_t)$ there exists a covering sequence $s_{init} \xrightarrow{\sigma} s \succeq s_t \in S_t$ where:

$$s_{init} = s_0 \xrightarrow{(v_1, t_1)} s_1 \xrightarrow{(v_2, t_2)} \dots \xrightarrow{(v_\ell, t_\ell)} s_\ell = s$$

$\lambda(\sigma) = \omega$, and $|\sigma| \leq B$. If cutting steps would instead of removing vertices would only forbid firing in them and their decedents, then after firing the sequence σ we would get a state with an underlining rooted tree T . T is isomorphic to a sub tree of T_B rooted in r_B , since each inner node can not have more than B children and its depth is smaller than B . Denote this isomorphism by ϕ .

We now show by induction that there exists a firing sequence

$$\mathbf{m}_B = \mathbf{m}_0 \xrightarrow{t'_1} \mathbf{m}_1 \xrightarrow{t'_2} \dots \xrightarrow{t'_\ell} \mathbf{m}_\ell$$

in \mathcal{N}^B such that $\mathbf{m}_i \geq M(\phi, s_i)$ and $\lambda_B(t'_i) = \lambda(t_i)$. For $i = 0$, $\mathbf{m}_0 = \mathbf{m}_B$ and this holds from the definition of \mathbf{m}_B . Assume we have shown it for $i < n$, and show it for $i = n$. We split the proof into 3 cases according to t_i :

- $t_i \in T_{el}$. We show that the transition $t'_i = t_i^{\phi(v_i)}$ is fireable from \mathbf{m}_{i-1} to an appropriate \mathbf{m}_i . From the induction hypothesis we know that:

$$\mathbf{m}_{i-1} \geq \mathbf{m}_{\text{Anc}}(\phi(v_i)) + [M(v_i)]^{\phi(v_i)} \geq \mathbf{Pre}^B(t^{\phi(v_i)})$$

hence $t^{\phi(v_i)}$ is fireable from \mathbf{m}_{i-1} . Since $\mathbf{C}^B(t_i^{\phi(v_i)})$ only changes the places of the type $p^{\phi(v_i)}$ for $p \in P$ and t_i only changes the marking of the vertex v_i we get that:

$$M(\phi, s_i) - M(\phi, s_{i-1}) = [\mathbf{C}_{t_i}]^{\phi(v_i)} = \mathbf{C}^B(t^{\phi(v_i)})$$

hence $\mathbf{m}_i = \mathbf{m}_{i-1} + \mathbf{C}^B(t^{\phi(v_i)}) \geq M(\phi, s_i)$, and $\lambda_B(t'_i) = \lambda_B(t^{\phi(v_i)}) = \lambda(t_i)$ by definition.

- $t_i \in T_{ab}$, which creates a new vertex w in s_i . We show that the transition $t'_i = t_i^{\phi(v_i), \phi(w)}$ is fireable from \mathbf{m}_{i-1} to an appropriate \mathbf{m}_i . By the same reasoning as in the elementary transitions and the fact that the vertex w is fresh (i.e. $p_{sleep}^{\phi(w)} = 1$) we have that $\mathbf{m}_{i-1} \geq \mathbf{Pre}^B(t_i^{\phi(v_i), \phi(w)})$. Note that the changes in s_i compared to s_{i-1} are a new edge to a new vertex w marked by Be_{t_i} , and that the marking of v_i changes according to \mathbf{Pre}_{t_i} . We get that:

$$\begin{aligned} M(\phi, s_i) - M(\phi, s_{i-1}) &= \vec{p}_{run}^{\phi(w)} + \mathbf{Ma}_{t_i}(\phi(w)) - \vec{p}_{sleep}^{\phi(w)} + [Be_{t_i}]^{\phi(w)} + \\ &\quad - [\mathbf{Pre}_{t_i}]^{\phi(v_i)} = \mathbf{C}^B(t^{\phi(v_i), \phi(w)}) \end{aligned}$$

hence $\mathbf{m}_i = \mathbf{m}_{i-1} + \mathbf{C}^B(t^{\phi(v_i), \phi(w)}) \geq M(\phi, s_i)$ and $\lambda_B(t'_i) = \lambda_B(t^{\phi(v_i), \phi(w)}) = \lambda(t_i)$ by definition.

- $t_i = \tau$, where $\Lambda(\text{prd}(v_i), v_i) = \hat{t}$, and $M_{s_{i-1}}(v_i) \geq \mathbf{m}_e \in \text{End}$. We show that the transition $\tau_{\mathbf{m}_e, \hat{t}}^{\phi(v_i)}$ is fireable from \mathbf{m}_{i-1} to an appropriate \mathbf{m}_i . By the same reasoning as for the elementary transitions $\mathbf{m}_{i-1} \geq \mathbf{Pre}^B(\tau_{\mathbf{m}_e, \hat{t}}^{\phi(v_i)})$. Note that the changes in s_i compared to s_{i-1} are that the vertex v_i is removed and the marking of $\text{prd}(v_i)$ changes according to $\mathbf{Post}_{\hat{t}}$. We get that:

$$M(\phi, s_i) - M(\phi, s_{i-1}) \leq [\mathbf{Post}_{\hat{t}}]^{\phi(\text{prd}(v_i))} - \vec{p}_{run}^{\phi(v_i)} = \mathbf{C}^B(t_{\mathbf{m}_e, \hat{t}}^{\phi(v_i)})$$

hence $\mathbf{m}_i = \mathbf{m}_{i-1} + \mathbf{C}^B(t_{\mathbf{m}_e, \hat{t}}^{\phi(v_i)}) \geq M(\phi, s_i)$ and $\lambda_B(t'_i) = \lambda_B(\tau_{\mathbf{m}_e, \hat{t}}^{\phi(v_i)}) = \lambda(\tau)$ by definition.

Recall that $s_\ell \rho \succeq s_t \in S_t$ where $\rho: V_{s_t} \rightarrow V_s$ is an injective morphism associated with the order. We get that by firing the sequence $\sigma' = t'_1 t'_2 \dots t'_\ell$:

$$\mathbf{m}_B \xrightarrow{\sigma'} \mathbf{m}_\ell \geq M(\phi, s_\ell) \geq M(\phi \circ \rho, s_t) \in S_t^B$$

Therefore $\omega = \lambda(\sigma') \in \mathcal{L}(\mathcal{N}^B, \mathbf{m}_B, S_t^B)$.

On the other hand if $\omega \in \mathcal{L}(\mathcal{N}^B, \mathbf{m}_B, S_t^B)$, there exists a covering sequence σ :

$$\mathbf{m}_B = \mathbf{m}_0 \xrightarrow{t_1} \mathbf{m}_1 \xrightarrow{t_2} \dots \xrightarrow{t_\ell} \mathbf{m}_\ell > \mathbf{m}_t$$

in \mathcal{N}^B such that $\lambda_B(\sigma) = \omega$ and $\mathbf{m}_t \in S_t^B$. We will show that there exists a firing sequence σ' :

$$s_{ini} = s_0 \xrightarrow{(v_1, t'_1)} s_1 \xrightarrow{(v_2, t'_2)} \dots \xrightarrow{(v_\ell, t'_\ell)} s_\ell \succeq s \in S_t$$

in \mathcal{N} such that for any $0 \leq i \leq \ell$ we have:

1. There is an isomorphism ϕ_i from a subtree of T^B to the underlying tree of the state s_i
2. For $U_i = \{v \in V_B \mid \mathbf{m}_i \geq \mathbf{m}_{\text{Anc}}(v)\}$, we have $\phi_i(U_i) = V_{s_i}$
3. For all vertices $v \in V_{s_i}$, and $u = \phi_i^{-1}(v)$:

$$0 \leq [\mathbf{m}_i]_u - \mathbf{Ma}_{\Lambda(\text{prd}(v), v)}(u) - \vec{p}_{run}^u \leq [M_{s_i}(v)]^u$$

where if $v = r$ then we denote $\text{prd}(r) = \Lambda(\text{prd}(r), r) = \emptyset$ and $\mathbf{Ma}_\emptyset(r) = 0$

4. $\lambda_B(t_i) = \lambda(t'_i)$.

We do this by induction on $0 \leq i \leq \ell$. For $i = 0$ since s_{init} has only one vertex we let $\phi_0(r_B) = r$ and from definition we have $\mathbf{m}_0 = \mathbf{m}_B$ which satisfies the assertions 1-4. Assume we have shown this for every $n < i$ and show it for $n = i$. We split the proof into 3 cases according to t_i :

- $t_i = t^u$ for $u \in V_B$ and $t \in T_{el}$. First note that since t^u was fired from \mathbf{m}_{i-1} we have $\mathbf{m}_{i-1} > p_{run}^u$. Let $v_i = \phi_{i-1}(u) \in V_{s_{i-1}}$ which exists by assertion 1.

We show that that letting $t'_i = t$ we would get that (v_i, t'_i) is fireable from s_{i-1} to an appropriate s_i . First notice that by induction:

$$[M_{s_{i-1}}(v_i)]^u \geq [\mathbf{m}_{i-1}]_u - \vec{p}_{run}^u - \mathbf{Ma}_{\Lambda(\text{prd}(v_i), v_i)}(u) \geq \mathbf{Pre}^B(t^u) - \mathbf{m}_{\text{Anc}}(u) = [\mathbf{Pre}_t]^u$$

hence $M_{s_{i-1}}(v_i) \geq \mathbf{Pre}_t$ and (v_i, t'_i) is fireable from s_{i-1} . Since t does not create new vertices and for any $w \in V_B$ and the transition t^u dose not change the tokens in the places p_{run}^w for $w \in V^B$, letting $\phi_i = \phi_{i-1}$ satisfies assertions 1 and 2. Note that, $\mathbf{C}^B(t^u)$ only changes the tokens in the places p^u for $p \in P$ hence $0 \leq [m_i]_u - \mathbf{Ma}_{\Lambda(\text{prd}(v_i), v_i)}(u) - \vec{p}_{run}^u$. Since (v_i, t'_i) only change the marking of v_i , we get that for any vertex $w \neq u$ assertion 3 holds. For u we get that :

$$[m_i]_u = [m_{i-1}]_u + \mathbf{C}^B(t^u) \leq [M_{s_{i-1}}(v_i) + \mathbf{C}_{t'_i}]^u + \mathbf{Ma}_{\Lambda(\text{prd}(v_i), v_i)}(u) + \vec{p}_{run}^u$$

and $M_{s_i}(v_i) = M_{s_{i-1}}(v_i) + \mathbf{C}_{t'_i}$ which satisfies assertion 3 for the node u . Finally by definition $\lambda(t'_i) = \lambda(t) = \lambda_B(t^u) = \lambda_B(t_i)$.

- $t_i = t^{u, u'}$ for $u, u' \in V_B$, $u = \text{prd}(u')$ in T_B and $t \in T_{ab}$. By the same reasoning as in previous case $v_i = \phi_{i-1}(u) \in V_{s_{i-1}}$ and letting $t'_i = t$ we get that (v_i, t'_i) is fireable from s_{i-1} . Since $t'_i \in T_{ab}$, firing (v_i, t'_i) creates for v_i a new child v'_i . Denote by:

$$\phi_i(w) = \begin{cases} \phi_{i-1}(w) & w \neq u' \\ v'_i & w = u' \end{cases}$$

ϕ_i is an isomorphism from a sub-tree of T_B to the underlining tree of s_i , hence assertion 1 holds. Moreover, since the transition $t_i = t^{u, u'}$ adds a token to $p_{run}^{u'}$ and for any node $w \neq u'$ does not decrees the tokens in p_{run}^w we get that $\phi_i(U_i) = \phi_i(U_{i-1} \cup u') = V_{s_{i-1}} \cup \{v'\} = V_{s_i}$ which satisfies assertion 2. For any $w \in V_{s_i}$ such that $w \neq v_i, v'_i$ $M_{s_i}(w) = M_{s_{i-1}}(w)$ since firing (v_i, t'_i) does not change their marking. For any $w \in V_B$ such that $w \neq u, u'$ $[m_i]_w = [m_{i-1}]_w$ since firing $t^{u, u'}$ does not change their marking. Therefore to show assertion 3 we only need to show it for the nodes u, u' , and we get:

$$\begin{aligned} [\mathbf{m}_i]_u &= [\mathbf{m}_{i-1}]_u + [\mathbf{C}^B(t^{u, u'})]_u \leq \\ &= [M_{s_{i-1}}(v_i) - \mathbf{Pre}_{t'_i}]^u + \vec{p}_{run}^u + \mathbf{Ma}_{\Lambda(\text{prd}(v_i), v_i)}(u) \\ [\mathbf{m}_i]_{u'} &= [\mathbf{m}_{i-1}]_{u'} + [\mathbf{Beg}_t]^{u'} + \vec{p}_{run}^{u'} + \mathbf{Ma}_t(u') - \vec{p}_{sleep}^{u'} \\ &= [M_{s_i}(v'_i)]^{u'} + \vec{p}_{run}^{u'} + \mathbf{Ma}_{\Lambda(v_i, v'_i)}(u') \end{aligned}$$

While in \mathcal{N} we have $M_{s_i}(v) = M_{s_{i-1}}(v_i) - \mathbf{Pre}_{t'_i}$ and $M_{s_i}(v'_i) = \mathbf{Beg}_{t'_i}$ which satisfies assertion 3 for the nodes u and u' . Finally by definition $\lambda(t'_i) = \lambda(t) = \lambda_B(t^{u, u'}) = \lambda_B(t_i)$.

- $t_i = \tau_{\mathbf{m}_e, t}^u$ for $t \in T_{ab}$ and $\mathbf{m}_e \in \text{End}$. Denote by u' the parent of u in T_B . By the same reasoning as in previous case $v_i = \phi_{i-1}(u) \in V_{s_{i-1}}$ and letting $t'_i = \tau$ we get that (v_i, t'_i) is fireable from s_{i-1} .

Firing (v_i, τ) removes v_i and its decedents, hence $V_{s_i} = V_{s_{i-1}} \setminus (Des_{s_i}(v_i))$. Denote by $\phi_i = \phi_{i-1}|_{\phi^{-1}(V_{s_i})}$ this is an isomorphism from a sub-tree of T_B to the underlining tree of s_i , hence assertion 1 holds. Moreover, since the transition $t_i = \tau_{\mathbf{m}_e, t}^u$ does not change the tokens in p_{run}^w for any $w \neq u$ and decrees the tokens in p_{run}^u to 0, we have $U_i = U_{i-1} \setminus (Des_{T_B}(u))$ and assertion 2 holds.

Let v'_i be the parent of v_i in s_{i-1} . For any $w \in V_{s_i}$ such that $w \neq v'_i$ $M_{s_i}(w) = M_{s_{i-1}}(w)$ since firing (v_i, τ) does not change their marking. For any $w \in U_i$ such that $w \neq u'$ we have $[\mathbf{m}_i]_w = [\mathbf{m}_{i-1}]_w$ since firing $\tau_{\mathbf{m}_e, t'}^u$ does not change their marking. Therefore to show assertion 3 we only need to show it for the node u' . Let $t' = \Lambda(v'_i, v_i)$ the labeling of the edge between v'_i to v_i in s_{i-1} , since $\tau_{\mathbf{m}_e, t}^u$ is fireable from m_{i-1} we know that $O_t \leq O_{t'}$ from which we conclude that $\mathbf{Post}_t \leq \mathbf{Post}_{t'}$, and we get:

$$\begin{aligned} [\mathbf{m}_i]_u &= [\mathbf{m}_{i-1}]_u + [\mathbf{C}^B(\tau_{\mathbf{m}_e, t}^u)]_u \leq [\mathbf{m}_{i-1}]_u + [\mathbf{C}^B(\tau_{\mathbf{m}_e, t'}^u)]_u = \\ &= [M_{s_{i-1}}(v') + \mathbf{Post}_{t'}]^u + \vec{p}_{run}^u + \text{Ma}_{\Lambda(v'', v')}(u) \end{aligned}$$

and $M_{s_i}(v') = M_{s_{i-1}}(v') + \mathbf{Post}_{t'}$ which satisfies assertion 3 for the node u' . Finally by definition $\lambda(t'_i) = \lambda(\tau) = \lambda_B(\tau_{\mathbf{m}_e, t'}^u) = \lambda_B(t_i)$.

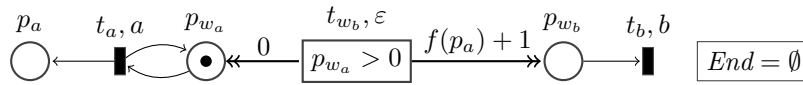
Recall that $\mathbf{m}_B = \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}_\ell > \mathbf{m}_t \in S_t^B$ hence there exists a φ such that $\mathbf{m}_t = s_t^\varphi$ for some $s_t \in S_t$. Moreover note that $\varphi(V_{s_t}) \subseteq U_i$. Hence $\psi = \phi_\ell \circ \varphi$ is an isomorphism from the underlining tree of s_t to the underlining tree of s_ℓ . For any $v \in V_{s_t}$ $M_{s_t}(v) \leq M_{s_\ell}(\psi(v))$ and for any $u, v \in E_{s_t}$ $\Lambda_{s_t}(u, v) \leq \Lambda_{s_\ell}(\psi(u), \psi(v))$ hence $s_t \preceq_\psi s_\ell$. Finally, by assertion 4 we have that $\omega = \lambda_B(\sigma) = \lambda(\sigma') \in \mathcal{L}(\mathcal{N}, s_{init}, S_t)$. ■

Proposition 2. *For all f , \mathcal{L}_f is a DRPN coverability language.*

Proof. Consider the DRPN \mathcal{N} in the figure below. Let the initial state, s_{init} , and final state, s_{tar} , both consist of a single vertex, where s_{init} is marked with \vec{p}_{w_a} and s_{tar} is marked with \vec{p}_{w_b} . Let λ be defined by $\lambda(t_a) = a$, $\lambda(t_b) = b$ and $\lambda(t_{w_a}) = \varepsilon$, the coverability language $\mathcal{L}(\mathcal{N}, s_{init}, \{s_{tar}\}) = \mathcal{L}_f$.

Since there is no abstract transition, all firings occur in the root and so we omit the vertex in the firing sequences. Every firing sequence can be either:

- t_a^n with trace a^n , a token in p_{w_a} , no token in p_{w_b} and n tokens in p_a ;
- $t_a^n t_{w_b} t_b^k$ with trace $a^n b^k$ such that $k \leq f(n) + 1$, no token p_{w_a} , $f(n) + 1 - k$ tokens in p_{w_b} and n tokens in p_a . So this sequence is a covering sequence iff $k \leq f(n)$ which concludes the proof.



■

To show that \mathcal{L}_f is not a Petri net coverability language, we first need to recall what is a *weakly computable* function.

Definition 7. A function $f : \mathbb{N} \rightarrow \mathbb{N}$ is called *weakly computable by a Petri net* if there exists a Petri net \mathcal{N} which has four designated places p_{beg} , p_{in} , p_{out} and p_{fin} and fulfills the following two properties:

1. For any $n \in \mathbb{N}$ there exists a sequence σ_n such that $p_{beg} + n \cdot p_{in} \xrightarrow{\sigma_n} \mathbf{m} \geq p_{end}$ and the number of tokens in p_{out} is exactly $f(n)$
2. For any $n \in \mathbb{N}$ and a sequence σ such that $p_{beg} + n \cdot p_{in} \xrightarrow{\sigma} \mathbf{m} \geq p_{end} + x \cdot p_{out}$ we have that $x \leq f(n)$.

In [11] the authors show the following result concerning weakly computable functions:

Theorem 4 ([11]). *Unbounded sublinear function cannot be weakly computed by a Petri net.*

Which brings us to the following conclusion:

Proposition 3. \mathcal{L}_f is not a PN coverability language.

Proof. Assume for contradiction that there is a Petri net $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{C} \rangle$ with an initial marking \mathbf{m}_{init} , a set of final markings M_t , and a labeling function $\lambda : T \rightarrow \{a, b, \varepsilon\}$ for which $\mathcal{L}(\mathcal{N}, \mathbf{m}_{init}, M_t) = \mathcal{L}_f$. We will now show that we can define a Petri net $\mathcal{N}' = \langle P', T', \mathbf{Pre}', \mathbf{C}' \rangle$ that will weakly computes f . Let $P' = P \cup \{p_{beg}, p_{run}, p_{fin}, p_{in}, p_{out}\}$ and $T' = T \cup \{t_{\mathbf{m}} \mid \mathbf{m} \in M_t\} \cup \{t_{run}\}$. For any $t \in T$ we set:

$$\mathbf{Pre}'(t) = \begin{cases} \mathbf{Pre}(t) + \vec{p}_{in} + \vec{p}_{run} & \lambda(t) = a \\ \mathbf{Pre}(t) + \vec{p}_{run} & \text{else} \end{cases}; \mathbf{C}'(t) = \begin{cases} \mathbf{C}(t) - \vec{p}_{in} & \lambda(t) = a \\ \mathbf{C}(t) + \vec{p}_{out} & \lambda(t) = b \\ \mathbf{C}(t) & \text{else} \end{cases}$$

For every $\mathbf{m} \in M_t$ we set $\mathbf{Pre}(t_{\mathbf{m}}) = \mathbf{m} + \vec{p}_{run}$ and $\mathbf{C}(t_{\mathbf{m}}) = \vec{p}_{fin} - \vec{p}_{run}$. Last we set $\mathbf{Pre}(t_{run}) = \vec{p}_{beg}$ and $\mathbf{C}(t_{run}) = \vec{p}_{run} + \mathbf{m}_{init}$.

For all $n \in \mathbb{N}$ there is a sequence σ_n in the original net, such that $\lambda(\sigma_n) = a^n b^{f(n)}$ and for some $\mathbf{m} \in S_t$ we have that $\mathbf{m}_{ini} \xrightarrow{\sigma_n} \mathbf{m}' \geq \mathbf{m}$. Therefore the sequence $t_{run} \sigma_n t_{\mathbf{m}}$ is fireable in the net \mathcal{N}' , from the marking $\vec{p}_{beg} + n \vec{p}_{in}$ reaching a marking greater or equal than \vec{p}_{end} with exactly $f(n)$ tokens in p_{out} .

Let $\mathbf{m}_n = \vec{p}_{beg} + n \vec{p}_{in}$ be an initial marking and σ be a sequence such that $\mathbf{m}_n \xrightarrow{\sigma} \mathbf{m} > x \vec{p}_{out} + \vec{p}_{end}$ for some x . The first transition in this sequence has to be t_{run} since its the only one fireable from \mathbf{m}_n . The final marking is greater or equal than p_{end} , hence there has to be a firing of $t_{\mathbf{m}}$, for some $\mathbf{m} \in M_t$, in the sequence but after firing $t_{\mathbf{m}}$ no other transition can be fired. Combining these two facts, we get that $\sigma = t_{run} \sigma' t_{\mathbf{m}}$ where $\sigma' \in T^*$ and it is fireable in the original net, where $\mathbf{m}_{init} \xrightarrow{\sigma'} \mathbf{m}' \geq \mathbf{m} \in M_t$. Therefore $\lambda(\sigma') = a^m b^\ell \in \mathcal{L}_f$, where $m \leq n$ since the transitions labeled with a can be fired at most n times (the initial marking of \mathcal{N}' had n tokens in p_{in}). Since $m \leq n$ we get that $\ell \leq f(m) \leq f(n)$. Therefore looking back to \mathcal{N}' we have that after firing σ we have $x \leq f(n)$. ■