On the efficient computation of the minimal coverability set for Petri nets

G. Geeraerts, J.-F. Raskin, and L. Van Begin

Computer Science Department, Université Libre de Bruxelles (U.L.B.)

Abstract. The *minimal coverability set* (MCS) of a Petri net is a finite representation of the downward-closure of its reachable markings. The minimal coverability set allows to decide several important problems like coverability, semi-liveness, place boundedness, etc. The classical algorithm to compute the MCS constructs the Karp&Miller tree [1]. Unfortunately the K&M tree is often huge, even for small nets. An improvement of this K&M algorithm is the Minimal Coverability Tree (MCT) algorithm [2], which has been introduced 15 years ago, and implemented since then in several tools such as Pep [3]. Unfortunately, we show in this paper that the MCT is flawed: it might compute an under-approximation of the reachable markings. We propose a new solution for the efficient computation of the MCS of Petri nets. Our experimental results show that this new algorithm behaves much better in practice than the K&M algorithm.

1 Introduction

Petri nets [4, 5] are a very popular formalism for the modeling and verification of parametric and concurrent systems [6]. The underlying transition graph of a Petri net is potentially infinite. Nevertheless, a large number of interesting verification problems are decidable on Petri nets. Among these decidable problems are the *coverability* problem (to which many safety verification problem can be reduced); the *boundedness* problem (is the number of reachable markings finite ?); the *place boundedness* problem (is the maximal reachable number of tokens bounded for some place p ?); the *semi-liveness* problem (is there a reachable marking in which some transition t is enabled).

In order to decide the aforementioned problems, one can use the *minimal coverability set* (MCS), which is a finite representation of some over-approximation of the reachable markings. The MCS is thus a very useful tool for the analysis of Petri nets, and an efficient algorithm to compute it is highly desirable.

Karp and Miller have shown, in their seminal paper [1], that the minimal coverability set is computable. The main idea of the Karp and Miller (K&M) algorithm is to build a finite tree that summarizes the potentially infinite unfolding of the reachability graph of the Petri net. In particular, this algorithm relies on an acceleration technique, which computes the limit of repeating any number of times some sequences of transitions that strictly increase the number of tokens in certain places. The acceleration technique is sound because Petri nets are *strictly monotonic*, i.e. a sequence of transitions which can be fired from a marking $\mathbf{m}$ can be fired from all markings $\mathbf{m}'$ such that $\mathbf{m} \preccurlyeq \mathbf{m}'$ (where $\preccurlyeq$ is a partial order for the markings). Furthermore sequence of transitions have constant effect, i.e. they add and subtract in each place the same number of tokens no matter from which marking they are fired. At the end of the execution of the K&M algorithm, one obtains a *coverability tree*, from which the MCS can be extracted.

Unfortunately, the K&M algorithm is often useless in practice because the finite tree that it builds is often much larger than the minimal coverability set, and cannot be constructed in reasonable time. As a consequence, a more efficient algorithm is needed. In [2], such an algorithm is proposed. The minimal coverability tree (MCT) builds on the idea of K&M but tries to take advantage more aggressively of the strict monotonicity of Petri nets. The main idea is to construct a tree where all markings that label nodes are incomparable wrt $\preccurlyeq$. To achieve this goal, reduction rules are applied at each step of the algorithm: each time a new marking is computed, it is compared to the other markings. If the new marking is smaller than a existing marking, the construction is not pursued from this new marking. If the new marking is larger than an existing marking, the subtree starting from that smaller marking is removed. The informal justification for this is as follows: the markings that are reachable from removed markings will be covered by markings reachable from the marking that was used for the removal, by the monotonicity property of Petri nets. While this idea is appealing and leads to small trees in practice, we show in this paper that, unfortunately, it is not correct: the MCT algorithm is not complete and can compute a strict under-approximation of the minimal coverability set. The flaw in the algorithm is intricate and we do not see an easy way to get rid of it.

So, instead of trying to fix the MCT algorithm, we consider the problem from scratch and propose a new efficient method to compute the MCS. It is based on novel ideas: first, we do not build a tree but handle sets of pairs of markings. Second, in order to exploit monotonicity property, we define an adequate order on pairs of markings that allows us to maintain sets of maximal pairs only. We give in this paper a detailed proof of correctness for this new method, and explain how to turn it into an efficient algorithm for the computation of the MCS of practically relevant Petri nets. We have implemented our algorithm in a prototype and compared its performance with the K&M algorithm. Our algorithm is orders of magnitude faster than the K&M algorithm.

The rest of the paper is organized as follows. In Section 2, we recall necessary preliminaries. In Section 3, we recall the KM as well as the MCT algorithms. In Section 4, we expose the bug in the MCT algorithm using an example and explain the essence of the flaw. In Section 5, we define the covering sequence, a sequence of sets of pairs of ω-markings that allows to compute the MCS. In Section 6, we show how to turn the concept of Section 5 into a practical algorithm and we report on results obtained with our prototype. Due to the lack of space, we provide most of the proofs in appendix.

2 Preliminaries

Petri nets Let us first recall the model of Petri nets, and fix several notations.

Definition 1. A *Petri net* [4,5] (PN for short) is a tuple $\mathcal{N} = \langle P, T \rangle$, where $P = \{p_1, p_2, \ldots, p_{|P|}\}$ is a finite set of places and $T = \{t_1, t_2, \ldots, t_{|T|}\}$ is a finite set of transitions. Each transition is a tuple $\langle I, O \rangle$, where $I : P \mapsto \mathbb{N}$ and $O : P \mapsto \mathbb{N}$ are respectively the input and output functions of the transition.

An example of Petri nets is to be found in Fig. 1(a). To define the semantics of PN, we first introduce the notion of ω-marking. An *ω-marking* $\mathbf{m}$ is a function $\mathbf{m} : P \mapsto (\mathbb{N} \cup \{\omega\})$ that associates a number of *tokens* to each place (ω meaning 'any natural number'). An ω-marking $\mathbf{m}$ is denoted either as $\langle \mathbf{m}(p_1), \mathbf{m}(p_2), \ldots, \mathbf{m}(p_{|P|}) \rangle$ (vector) , or as $\{\mathbf{m}(p_{i_1})p_{i_1}, \mathbf{m}(p_{i_2})p_{i_2}, \ldots, \mathbf{m}(p_{i_k})p_{i_k}\}$ (multiset), where $p_{i_1}, p_{i_2}, \ldots, p_{i_k}$ are exactly the places that contain at least one token (we omit $\mathbf{m}(p)$ when it is equal to 1). For example, $\langle 0, 1, 0, \omega, 2\rangle$ and $\{p_2, \omega p_4, 2p_5\}$ denote the same ω-marking. An ω-marking $\mathbf{m}$ is a *marking* iff $\forall p \in P : \mathbf{m}(p) \neq \omega$.

Let $\mathcal{N} = \langle P, T\rangle$ be a PN, $\mathbf{m}$ be an ω-marking of $\mathcal{N}$ and $t = \langle I, O\rangle \in T$ be a transition. Then, t is *enabled* in $\mathbf{m}$ iff $\mathbf{m}(p) \geq I(p)$ for any $p \in P$ (we assume that $\omega \geq \omega$ and $\omega > c$ for any $c \in \mathbb{N}$). In that case, t can *fire* and transforms $\mathbf{m}$ into a new ω-marking $\mathbf{m}'$ s.t. for any $p \in P$: $\mathbf{m}'(p) = \mathbf{m}(p) - I(p) + O(p)$ (assuming that $\omega - c = \omega = \omega + c$ for any $c \in \mathbb{N}$). We denote this by $\mathbf{m} \xrightarrow{t} \mathbf{m}'$, and extend the notation to sequences of transitions $\sigma = t_1 t_2 \cdots t_k \in T^*$, i.e., $\mathbf{m} \xrightarrow{\sigma} \mathbf{m}'$ iff either $\sigma = \varepsilon$ (the empty sequence) and $\mathbf{m} = \mathbf{m}'$, or there are $\mathbf{m}_1, \ldots, \mathbf{m}_{k-1}$ s.t. $\mathbf{m} \xrightarrow{t_1} \mathbf{m}_1 \xrightarrow{t_2} \cdots \xrightarrow{t_k} \mathbf{m}'$. Given an ω-marking $\mathbf{m}$ of some PN $\mathcal{N} = \langle P, T\rangle$, we let $\mathsf{Post}(\mathbf{m}) = \{\mathbf{m}' \mid \exists t \in T : \mathbf{m} \xrightarrow{t} \mathbf{m}'\}$ and $\mathsf{Post}^*(\mathbf{m}) = \{\mathbf{m}' \mid \exists \sigma \in T^* : \mathbf{m} \xrightarrow{\sigma} \mathbf{m}'\}$. Given a sequence of transitions $\sigma = t_1 t_2 \cdots t_k$ with $t_i = \langle I_i, O_i\rangle$ for any $1 \leq i \leq k$, we let, for any place p, $\sigma(p) = \sum_{i=1}^{k}(I_i(p) - O_i(p))$, i.e., the effect of σ on p.

In the following, we use the order $\preccurlyeq$ for ω-markings.

Definition 2. Let P be a set of places of some PN. Then, $\preccurlyeq \subseteq (\mathbb{N} \cup \{\omega\})^{|P|} \times (\mathbb{N} \cup \{\omega\})^{|P|}$ is the relation s.t. for any $\mathbf{m}_1, \mathbf{m}_2 \in (\mathbb{N} \cup \{\omega\})^{|P|}$, $\mathbf{m}_1 \preccurlyeq \mathbf{m}_2$ iff for any $p \in P$: $\mathbf{m}_1(p) \leq \mathbf{m}_2(p)$.

We write $\mathbf{m} \prec \mathbf{m}'$ when $\mathbf{m} \preccurlyeq \mathbf{m}'$ but $\mathbf{m} \neq \mathbf{m}'$.

Finally, it is well-known that PN are *strictly monotonic*. That is, if $\mathbf{m}_1$, $\mathbf{m}_2$ and $\mathbf{m}_3$ are three ω-markings and t is a transition of some PN $\mathcal{N}$ s.t. $\mathbf{m}_1 \xrightarrow{t} \mathbf{m}_2$ and $\mathbf{m}_1 \prec \mathbf{m}_3$, then, t is enabled in $\mathbf{m}_3$ and the marking $\mathbf{m}_4$ with $\mathbf{m}_3 \xrightarrow{t} \mathbf{m}_4$ is s.t. $\mathbf{m}_2 \prec \mathbf{m}_4$.

Covering and coverability sets Given a set M of ω-markings, we define the set of maximal elements of M as $\mathsf{Max}^{\preccurlyeq}(M) = \{\mathbf{m} \in M \mid \nexists \mathbf{m}' \in M : \mathbf{m} \prec \mathbf{m}'\}$. Given an ω-marking $\mathbf{m}$ (ranging over set of places P), its *downward-closure* is the set *of markings* $\downarrow^{\preccurlyeq}(\mathbf{m}) = \{\mathbf{m}' \in \mathbb{N}^{|P|} \mid \mathbf{m}' \preccurlyeq \mathbf{m}\}$. Given a set M of ω-markings, we let $\downarrow^{\preccurlyeq}(M) = \cup_{\mathbf{m} \in M} \downarrow^{\preccurlyeq}(\mathbf{m})$. A set D of markings is said to be *downward-closed* whenever $\downarrow^{\preccurlyeq}(D) = D$. Then:

Definition 3. Let $\mathcal{N} = \langle P, T\rangle$ be a PN and let $\mathbf{m}_0$ be the initial ω-marking of $\mathcal{N}$ (for $\mathbf{m}_0$). The *covering set* of $\mathcal{N}$, denoted as $\mathsf{Cover}(\mathcal{N}, \mathbf{m}_0)$ is the set $\downarrow^{\preccurlyeq}(\mathsf{Post}^*(\mathbf{m}_0))$.

Given a PN $\mathcal{N}$ with initial marking $\mathbf{m}_0$, a *coverability set* for $\mathcal{N}$ and $\mathbf{m}_0$ is a finite sub-set $S \subseteq (\mathbb{N} \cup \{\omega\})^{|P|}$ such that $\downarrow^{\preccurlyeq}(S) = \mathsf{Cover}(\mathcal{N}, \mathbf{m}_0)$. Such a set always exists because any downward-closed set of markings can be represented by a finite set of ω-markings:

Lemma 1 ([7]). *For any subset $D \subseteq \mathbb{N}^k$ such that $\downarrow^{\preccurlyeq}(D) = D$ there exists a finite subset $S \subset (\mathbb{N} \cup \{\omega\})^k$ such that $\downarrow^{\preccurlyeq}(S) = D$.*

It is also well-known [2] that there exists one minimal (in terms of $\subseteq$) coverability set (called the *minimal coverability set*).

Labeled trees Finally, let us introduce the notion of *labeled tree*:

Definition 4. Given a set of places P, a labeled tree is a tuple $\mathcal{T} = \langle N, B, root, \Lambda\rangle$, s.t. $\langle N, B, root\rangle$ forms a tree (N is the set of nodes, $B \subseteq N \times N$ is the set of edges and $root \in N$ is the root node) and $\Lambda : N \mapsto (\mathbb{N} \cup \{\omega\})^{|P|}$ is a labeling function of the nodes by ω-markings.

Given two nodes n and n' in N, we write respectively $B(n, n')$, $B^*(n, n')$ $B^+(n, n')$ instead of $(n, n') \in B$, $(n, n') \in B^*$, $(n, n') \in B^+$.

3 The Karp&Miller and the MCT algorithms

The Karp and Miller algorithm The Karp&Miller algorithm [1] is a well-known solution to compute a coverability set of a PN. It consists in building a labeled tree whose root is labeled by $\mathbf{m}_0$. The tree is obtained by unfolding the transition relation of the PN, and by applying *accelerations*, which exploit the strict monotonicity property of PN. That is, let us assume that $\mathbf{m}_1$ and $\mathbf{m}_2$ are two ω-markings s.t. $\mathbf{m}_1 \prec \mathbf{m}_2$ and there exists a sequence of transitions σ with $\mathbf{m}_1 \xrightarrow{\sigma} \mathbf{m}_2$. By (strict) monotonicity, σ is firable from $\mathbf{m}_2$ and produces a ω-marking $\mathbf{m}_3$ s.t. $\mathbf{m}_2 \prec \mathbf{m}_3$. As a consequence, all the places p s.t. $\mathbf{m}_1(p) < \mathbf{m}_2(p)$ are unbounded. Hence, the ω-marking $\mathbf{m}_\omega$ defined as $\mathbf{m}_\omega = \omega$ if $\mathbf{m}_1(p) < \mathbf{m}_2(p)$, and $\mathbf{m}_\omega = \mathbf{m}_1(p)$ otherwise, has the property that $\downarrow^{\preccurlyeq}(\mathbf{m}_\omega) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*(\mathbf{m}_1))$. This can be generalized to the case where we consider an ω-marking $\mathbf{m}$ and a set S of ω-markings s.t. for any $\mathbf{m}' \in S$: $\mathbf{m} \in \mathsf{Post}^*(\mathbf{m}')$. Hence, the following acceleration function:

$$\forall p \in P : \mathsf{Accel}(S, \mathbf{m})(p) = \begin{cases} \omega & \text{if } \exists \mathbf{m}' \in S : \mathbf{m}' \prec \mathbf{m} \text{ and } \mathbf{m}'(p) < \mathbf{m}(p) \\ \mathbf{m}(p) & \text{Otherwise} \end{cases}$$

The Karp&Miller procedure (see Algorithm 1) relies on this function: when developing the successors of a node n, it calls the acceleration function on every $\mathbf{m} \in \mathsf{Post}(\Lambda(n))$, by letting S be the set of all the markings that are met along the branch ending in n. This procedure terminates and computes a coverability set:

Theorem 1 ([1]). *For any* PN $\mathcal{N} = \langle P, T\rangle$ *with initial ω-marking* $\mathbf{m}_0$*, the* KM *procedure produces a finite labeled tree* $\mathcal{T} = \langle N, B, root, \Lambda\rangle$*, s.t.* $\downarrow^{\preccurlyeq}(\{\Lambda(n) \,|\, n \in N\}) = \mathsf{Cover}(\mathcal{N}, \mathbf{m}_0)$.

Properties of the Karp&Miller tree Let $n \neq root$ be a node of some Karp&Miller tree. Hence, $\Lambda(n)$ has been obtained by calling Accelerate with parameters S and $\mathbf{m}$. In this case, we say that n has been obtained *by the acceleration of* $\mathbf{m}$ (with S). For any node $n \neq root$ of any Karp&Miller tree, we assume that the function $\mathsf{M}(n)$ returns the marking $\mathbf{m}$ s.t. $\Lambda(n)$ has been obtained by the acceleration of $\mathbf{m}$. Remark that, for any node $n \neq root$, $\mathsf{M}(n) \in \mathsf{Post}(\Lambda(n'))$ where n' is the father of n. Remark that it might be the case that $\Lambda(n) = \mathsf{M}(n)$.

Let $\mathcal{N} = \langle P, T\rangle$ be a PN with initial marking $\mathbf{m}_0$ and let $\mathcal{T} = \langle N, B, root, \Lambda\rangle$ be its Karp&Miller tree. Then, $\varsigma : N \mapsto T^*$ is a function that associates a sequence of transitions to every node n, as follows. (i) If $n = root$, then $\varsigma(n)$ returns the empty sequence. (ii) If there is no $n' \in N$ s.t. $B^+(n', n)$, $\Lambda(n') \neq \Lambda(n)$ and $\Lambda(n') \preccurlyeq \Lambda(n)$ (hence, n is such that $\Lambda(n) = \mathsf{M}(n)$), then $\varsigma(n)$ returns the empty sequence. (iii)

Data: A PN $\mathcal{N} = \langle P, T \rangle$ and an initial ω-marking $\mathbf{m}_0$.
Result: The minimal coverability set of $\mathcal{N}$ for $\mathbf{m}_0$.
$\mathsf{KM}\,(\mathcal{N}, \mathbf{m}_0)$ **begin**
- $\mathcal{T} \leftarrow \langle N, B, n_0, \Lambda \rangle$ where $N = \{n_0\}$, $B = \emptyset$ and $\Lambda\,(n_0) = \mathbf{m}_0$;
- $to\_treat \leftarrow \{n_0\}$;
- **while** $to\_treat \neq \emptyset$ **do**
  - **let** n be a node of to_treat ;
  - $to\_treat \leftarrow to\_treat \setminus \{n\}$;
  - **if** $\nexists \overline{n} : B^+(\overline{n}, n) \wedge \Lambda\,(\overline{n}) = \Lambda\,(n)$ **then**
    - **foreach** $\mathbf{m} \in \mathsf{Post}\,(\Lambda\,(n))$ **do**
      - $S \leftarrow \{\Lambda\,(n') \mid B^*(n', n)\}$;
      - Let n' be a new node s.t. $\Lambda\,(n') = \mathsf{Accel}\,(S, \mathbf{m})$;
      - $N \leftarrow N \cup \{n'\}$;
      - $B \leftarrow B \cup \{(n, n')\}$;
      - $to\_treat \leftarrow to\_treat \cup \{n'\}$;
- `return` $(\{\Lambda\,(n) \mid n \in N \wedge \nexists n' \in N : \Lambda\,(n') \succ \Lambda\,(n)\})$;

end

Algorithm 1: The KM algorithm.

Otherwise, n has been obtained by the acceleration of $\mathsf{M}(n)$. Let $P_a = \{p \in P \mid \Lambda\,(n)\,(p) = \omega$ and $\mathsf{M}(n)\,(p) \neq \omega\}$ and let $P_\omega = \{p \in P \mid \Lambda\,(n)\,(p) = \mathsf{M}(n)\,(p) = \omega\}$. In that case, $\varsigma\,(n)$ returns one of the finite non-empty sequences such that for any $p \in P_a$: $\varsigma\,(n)\,(p) > 0$; for any $p \in P \setminus (P_a \cup P_\omega)$: $\varsigma\,(n)\,(p) = 0$; and $\varsigma\,(n)$ is firable from $\mathsf{M}(n)$.

The existence of $\varsigma\,(n)$ in the third case is guaranteed by the following lemma, that can be extracted from the main proof of the Algorithm 1, in [1]:

Lemma 2 ([1]). *Let $\mathcal{N} = \langle P, T \rangle$ be a PN with initial ω-marking $\mathbf{m}_0$ and let $\mathcal{T} = \langle N, B, root, \Lambda \rangle$ be its Karp&Miller tree. Let $n \neq root$ be a node of $\mathcal{T}$. Let $P_a = \{p \in P \mid \Lambda\,(n)\,(p) = \omega$ and $\mathsf{M}(n)\,(p) \neq \omega\}$ and $P_\omega = \{p \in P \mid \Lambda\,(n)\,(p) = \mathsf{M}(n)\,(p) = \omega\}$. Then, there exists a sequence of transitions $\sigma \in T^*$ s.t.: (i) for any $p \in P_a$: $\sigma(p) > 0$. (ii) for any $p \in P \setminus (P_a \cup P_\omega)$: $\sigma(p) = 0$. (iii) σ is firable from $\mathsf{M}(n)$.*

The MCT algorithm The *minimal coverability tree algorithm* (MCT for short) has been introduced by Finkel in [2], as an optimization of the Karp&Miller algorithm. It is recalled in Algorithm 2, and relies on two auxiliary functions: given a labeled tree $\mathcal{T}$ and a node n of $\mathcal{T}$, $\mathsf{removeSubtree}(n, \mathcal{T})$ removes the subtree rooted by n from $\mathcal{T}$. The function $\mathsf{removeSubtreeExceptRoot}(n, \mathcal{T})$ is similar to $\mathsf{removeSubtree}(n, \mathcal{T})$ except that the root node n is not removed. The main idea consists in exploiting the monotonicity property of PN in order to avoid developing part of the nodes of the Karp&Miller tree, as well as removing some subtrees during the construction. With respect to the Karp&Miller algorithm, three main differences can be noted. Let n be a node picked from to_treat. First, when there already exists another node $\overline{n}$ with $\Lambda\,(n) = \Lambda\,(\overline{n})$ in the tree, n, is not developed (line (a)). Second, when n is accelerated (line (b)), the result of the acceleration is assigned to the label of its highest ancestor $\overline{n}$ s.t. $\Lambda\,(\overline{n}) \prec \Lambda\,(n)$, and the whole subtree of $\overline{n}$ is removed from the tree. Third, the algorithm avoids adding a node n' to the tree if there is another node $\overline{n}$ s.t. $\Lambda\,(\overline{n}) \succcurlyeq \Lambda\,(n')$ (line (c)).

Data: A PN $\mathcal{N} = \langle P, T \rangle$ and an initial marking $\mathbf{m}_0$
Result: The minimal coverability set of $\mathcal{N}$.

```
MCT (N, m0) begin
    T ← ⟨N, B, n0, Λ⟩ where N = {n0}, B = ∅ and Λ(n0) = m0 ;
    to_treat ← {n0} ;
    while to_treat ≠ ∅ do
        Select some node n in to_treat and remove it;
(a)     if ∄n̄ ∈ N s.t. Λ(n̄) = Λ(n) then
            foreach m ∈ Post(Λ(n)) do
(b)             if ∃n̄ : B*(n̄, n) and Λ(n̄) ≺ m then
                    Let n̄ be the highest node s.t. B*(n̄, n) ∧ Λ(n̄) ≺ m ;
                    Λ(n̄) ← Accel({n' ∈ N | B*(n', n)}, m) ;
                    to_treat ← (to_treat \ {n' | B*(n̄, n')}) ∪ {n̄} ;
                    removeSubtreeExceptRoot(n̄, T) ;
                    break ;
(c)             else if ∄n̄ ∈ N s.t. m ≺ Λ(n̄) then
                    Let n' be a new node s.t. Λ(n') = m ;
                    N ← N ∪ {n'} ; B ← B ∪ (n, n') ;
                    to_treat ← to_treat ∪ {n'};
(d)             while ∃n1, n2 ∈ N : Λ(n1) ≺ Λ(n2) do
                    to_treat ← to_treat \ {n | B*(n1, n)} ;
                    removeSubtree(n1, T) ;
    return({Λ(n) | n ∈ N}) ;
end
```

Algorithm 2: The MCT algorithm [2].

Moreover, the adjunction of n' to the tree (when it happens) triggers the deletion of all the subtrees rooted in some node n'' s.t. $\Lambda(n'') \prec \Lambda(n')$ (line (d)).

Remark that this algorithm is *non-deterministic*, in the sense that no ordering is imposed on the nodes in to_treat. Hence, any strategy that fixes the exploration order (which can possibly improve the efficiency of the algorithm) can be chosen.

4 Counter-example to the MCT algorithm

In this section, we introduce a PN on which the MCT algorithm might compute a *strict under-approximation* of the covering set (see Fig. 1). Fig 1(a) is the PN on which we run the MCT algorithm, and Fig. 1(b) through 1(f) are the key points of the execution.

Let us briefly comment on this execution. First remark that place p_5 of the PN in Fig. 1(a) is unbounded, because marking $\{p_3\}$ is reachable from the initial marking $\mathbf{m}_0 = \{p_1\}$ by firing t_1t_2, and the sequence t_3t_4 can be fired an arbitrary number of times from $\{p_3\}$, and strictly increases the markings of p_5. Then, one possible execution of MCT is as follows (markings in the frontier are underlined):

Fig. 1(b) The three successors of $\mathbf{m}_0$ are computed. Then, the branch rooted in $\{p_2\}$ is unfolded, by firing t_2, t_3 and t_4. At that point, two comparable markings $\{p_3\}$ and $\{p_3, p_5\}$ are met with and an acceleration occurs (line (b) of Algorithm 2) . The result is $\{p_3, \omega p_5\}$, which is put into to_treat.

Fig. 1(c) The subtree rooted in $\{p_6\}$ is unfolded. After the firing of t_6t_4, one obtains $\{p_3, 3p_5\}$, which is smaller than $\{p_3, \omega p_5\}$. Hence, $\{p_3, 3p_5\}$ is not put into to_treat and the branch is stopped (line (c)).

Fig. 1(d) The subtree rooted in $\{p_7\}$ is developed. The unique successor $\{p_2, p_5\}$ of $\{p_7\}$ is larger than $\{p_2\}$. Hence, the subtree rooted in $\{p_2\}$ (including $\{p_3, \omega p_5\}$, still in the frontier) is removed (line (d)).

Fig. 1(e) and 1(f) The tree (actually a single branch) rooted in $\{p_2, p_5\}$ (only node in the frontier) is further developed through the firing of t_2 and t_3. The resulting node $\{p_4, p_5\}$ is strictly smaller than $\{p_4, 2p_5\}$. Hence, that branch is stopped too (line (c)), and the frontier becomes empty. The final result of the algorithm is shown in Fig. 1(f). It is not difficult to see that the set of labels of this tree does not form a coverability set, because it contains no marking $\mathbf{m}$ s.t. $\mathbf{m}(p_5) = \omega$.

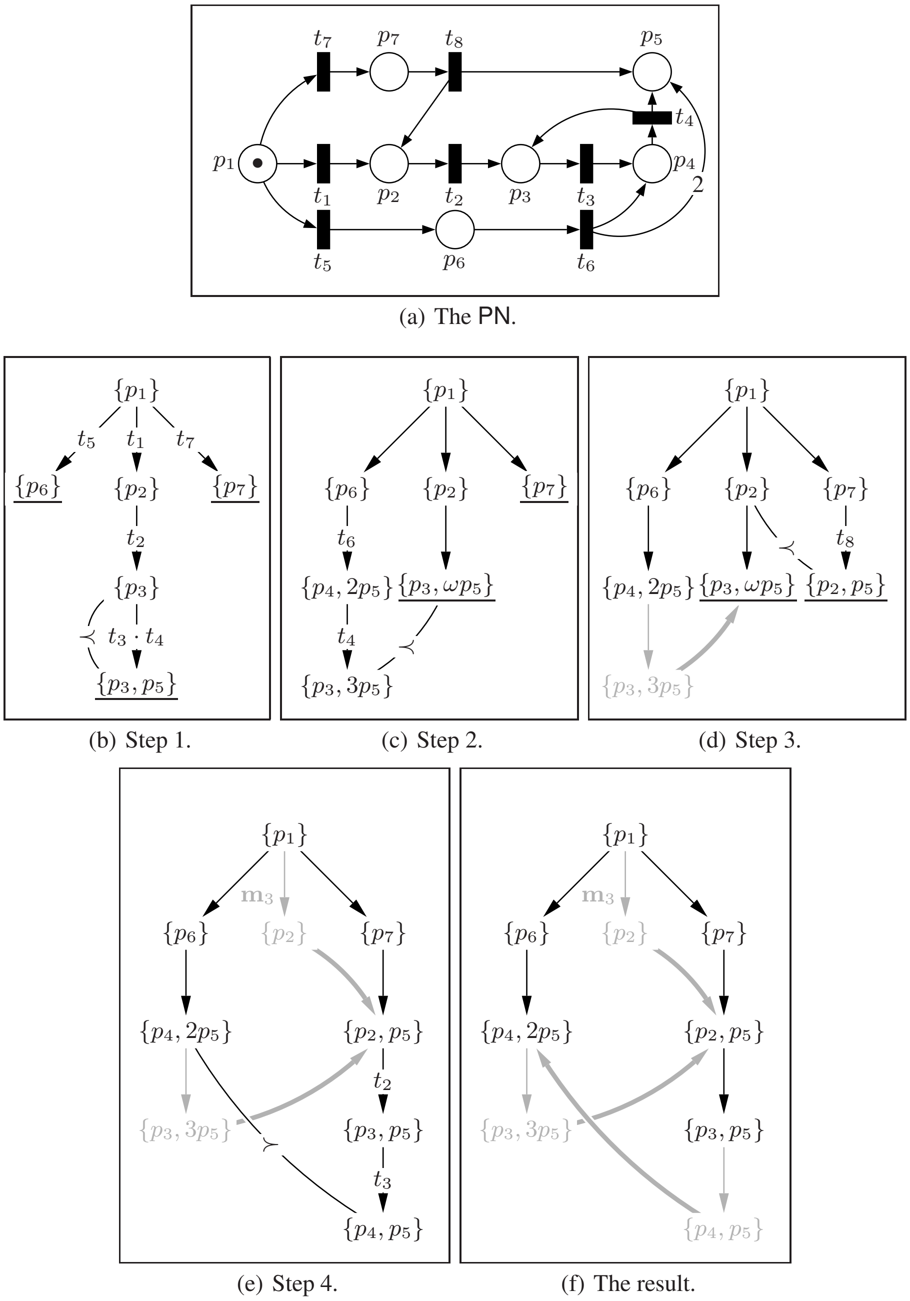
Comment on the counter-example This counter-example allows us to identify a flaw in the logic of the MCT algorithm. The algorithm stops the development of a node n or removes the subtree rooted in n because its has found another node n' s.t. $\Lambda(n')$ is larger than $\Lambda(n)$. In that case, we say that n' is *a proof for* n. The intuition behind this notion is that, by monotonicity, all the successors of n should be covered by some successor of n'. Thus, when n' is a proof for n, the algorithm makes implicitly the hypothesis that either all the successors of n' will be fully developed, or that they will be covered by some other nodes of the tree.

In our counter-example, that reasoning fails because *cycles* appear in 'proofs'. In Fig. 1, we have drawn a thick gray arrow from n to n' when n' is a proof for n. On Fig. 1(d), the node labeled by $\{p_3, \omega p_5\}$, which is a proof for $\{p_3, 3p_5\}$ is deleted, because of $\{p_2, p_5\}$. Hence, $\{p_2, p_5\}$, becomes the proof of $\{p_3, 3p_5\}$ (see Fig. 1(e)). The cycle clearly appears in Fig 1(f): all the successors of $\{p_4, 2p_5\}$ will be eventually covered under the assumption that all the successors of $\{p_2, p_5\}$ are covered. However, this happens under only if all the successors of $\{p_4, 2p_5\}$ are eventually covered.

Implementation of the MCT in the Pep tool Actually, the flaw in the MCT algorithm has already been independently discovered by the team of Prof. Peter Starke. They have implemented in INA (a component of the toolkit Pep [3]) a variation of the MCT which is supposed to correct the aforementioned bug. To the best of our knowledge, this implementation (and the discovery of the bug) has been documented only in a master's thesis in German [8]. Unfortunately, their version of the MCT contains a flaw too, because it offers no guarantee of termination [9, 10], although [8] contains a proof of termination. See [11] for a counter-example to termination. Thus, from our point of view, fixing the bug of the MCT algorithm seems to be a difficult task.

5 The covering sequence

Instead of trying to fix the bug in the MCT algorithm, we propose a different solution based on novel ideas. To introduce our new solution, let us look back at the basics. Remember that we want to compute an effective representation of $\downarrow^{\preccurlyeq}(\mathsf{Post}^*(\mathbf{m}_0))$. It is easy to show that this set is the limit of the following infinite sequence of sets: $X_0 = \downarrow^{\preccurlyeq}(\{\mathbf{m}_0\})$, and for $i \geq 1$, $X_i = \downarrow^{\preccurlyeq}(X_{i-1} \cup \mathsf{Post}(X_{i-1}))$. Note that by strict monotonicity of Petri nets, we can instead consider the following sequence that handles maximal elements only: $Y_0 = \mathsf{Max}^{\preccurlyeq}(\{\mathbf{m}_0\})$, and $Y_i = \mathsf{Max}^{\preccurlyeq}(Y_{i-1} \cup \mathsf{Post}(Y_{i-1}))$



(a) The PN.

(b) Step 1. (c) Step 2. (d) Step 3.

(e) Step 4. (f) The result.

Fig. 1. A counter-example to the MCT algorithm. Underlined markings are in the frontier. A gray arrow from n to n' means that n' is a 'proof' for n.

for any $i \geq 1$. Unfortunately, this is not an effective way to compute the minimal coverability set as we do not know how to compute the limit of this sequence. To compute that limit, we need accelerations. Accelerations are computed from pairs of markings. Our solution constructs a sequence of sets of pairs of markings on which we systematically apply a variant of the Post operator and a variant of the acceleration function. By defining an adequate order $\sqsubseteq$ on pairs of markings, we show that we can concentrate on maximal elements for $\sqsubseteq$.

Preliminaries Let $\mathbf{m}_1$ and $\mathbf{m}_2$ be two ω-markings. Then, $\mathbf{m}_1 \ominus \mathbf{m}_2$ is a function $P \mapsto \mathbb{Z} \cup \{-\omega, \omega\}$ s.t. for any place p: $(\mathbf{m}_1 \ominus \mathbf{m}_2)(p)$ is equal to ω if $\mathbf{m}_1(p) = \omega$; $-\omega$ if $\mathbf{m}_2(p) = \omega$ and $\mathbf{m}_1(p) \neq \omega$; $\mathbf{m}_1(p) - \mathbf{m}_2(p)$ otherwise. Then, given two pairs of ω-markings $(\mathbf{m}_1, \mathbf{m}_2)$ and $(\mathbf{m}'_1, \mathbf{m}'_2)$, we have $(\mathbf{m}_1, \mathbf{m}_2) \sqsubseteq (\mathbf{m}'_1, \mathbf{m}'_2)$ iff $\mathbf{m}_1 \preccurlyeq \mathbf{m}'_1$, $\mathbf{m}_2 \preccurlyeq \mathbf{m}'_2$ and for any place p: $(\mathbf{m}_2 \ominus \mathbf{m}_1)(p) \leq (\mathbf{m}'_2 \ominus \mathbf{m}'_1)(p)$.

For any $(\mathbf{m}_1, \mathbf{m}_2)$, we let $\downarrow^{\sqsubseteq}((\mathbf{m}_1, \mathbf{m}_2)) = \{(\mathbf{m}'_1, \mathbf{m}'_2) \mid (\mathbf{m}'_1, \mathbf{m}'_2) \sqsubseteq (\mathbf{m}_1, \mathbf{m}_2)\}$. We extend this to sets of pairs R as follows: $\downarrow^{\sqsubseteq}(R) = \cup_{(\mathbf{m}_1,\mathbf{m}_2)\in R} \downarrow^{\sqsubseteq}((\mathbf{m}_1, \mathbf{m}_2))$. Given a set R of pairs of markings, we let $\mathsf{Max}^{\sqsubseteq}(R) = \{(\mathbf{m}_1, \mathbf{m}_2) \in R \mid \nexists (\mathbf{m}'_1, \mathbf{m}'_2) \in R : (\mathbf{m}_1, \mathbf{m}_2) \neq (\mathbf{m}'_1, \mathbf{m}'_2) \wedge (\mathbf{m}_1, \mathbf{m}_2) \sqsubseteq (\mathbf{m}'_1, \mathbf{m}'_2)\}$

Our new solution relies on a weaker acceleration function than that of Karp&Miller (because its first argument is restricted to a single marking instead of a set of markings). Given two ω-markings $\mathbf{m}_1$ and $\mathbf{m}_2$ s.t. $\mathbf{m}_1 \preccurlyeq \mathbf{m}_2$, we let $\mathsf{AccelPair}(\mathbf{m}_1, \mathbf{m}_2) = \mathbf{m}_\omega$ s.t. for any place p, $\mathbf{m}_\omega(p) = \mathbf{m}_1(p)$ if $\mathbf{m}_1(p) = \mathbf{m}_2(p)$; $\mathbf{m}_\omega(p) = \omega$ otherwise. According to the following lemma, this acceleration function is sound:

Lemma 3. *Let $\mathcal{N}$ be a* PN *and let $\mathbf{m}_1$ and $\mathbf{m}_2$ be two ω-markings of $\mathcal{N}$ that respect $\mathbf{m}_1 \preccurlyeq \mathbf{m}_2$ and $\downarrow^{\preccurlyeq}(\mathbf{m}_2) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*(\mathbf{m}_1))$. Then, $\downarrow^{\preccurlyeq}(\mathsf{AccelPair}(\mathbf{m}_1, \mathbf{m}_2)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*(\mathbf{m}_2))$.*

Moreover, given a labeled tree $\mathcal{T} = \langle N, B, root, \Lambda \rangle$, we let, for any $n \in N$, $\mathsf{Anc}(\mathcal{T}, n) = \{n' \mid B^*(n', n)\}$ (that is, $\mathsf{Anc}(\mathcal{T}, n)$ is the set of 'ancestors' of n in $\mathcal{T}$, n included). Then, the following lemma draws a link between $\mathsf{AccelPair}$ and the Karp&Miller acceleration. It shows that, although $\mathsf{AccelPair}$ is weaker, it can produce the same results than the Karp&Miller acceleration, when properly applied.

Lemma 4. *Let $\mathcal{N} = \langle P, T \rangle$ be a Petri net with initial marking $\mathbf{m}_0$ and let $\mathcal{T} = \langle N, B, root, \Lambda \rangle$ be its Karp&Miller tree. Let $n \neq root$ be a node of $\mathcal{T}$. Let $\mathbf{m}'$ be s.t. $\mathsf{M}(n) \xrightarrow{\varsigma(n)} \mathbf{m}'$. Then, $\Lambda(n) \preccurlyeq \mathsf{AccelPair}(\mathsf{M}(n), \mathbf{m}')$.*

Proof. Let $P_a = \{p \in P \mid \Lambda(n)(p) = \omega \wedge \mathsf{M}(n)(p) \neq \omega\}$. By construction, for any place p, $\Lambda(n)(p) = \omega$ if $p \in P_a$; $\Lambda(n)(p) = \mathsf{M}(n)$ otherwise. Moreover, by definition of $\mathsf{AccelPair}$, for any place p, $\mathsf{AccelPair}(\mathsf{M}(n), \mathbf{m}')(p) = \omega$ if $\varsigma(n)(p) > 0$; and $\mathsf{AccelPair}(\mathsf{M}(n), \mathbf{m}')(p) = \mathsf{M}(n)(p)$ otherwise. By def. of $\varsigma(n)$, $p \in P_a$ implies $\varsigma(n)(p) > 0$, and $p \notin P_a$ implies $\Lambda(n)(p) = \mathsf{M}(n)(p)$. Hence the lemma. □

Finally, we introduce several operators that work directly on pairs of markings. Given a set R of pairs of ω-markings, we let $\mathsf{Flatten}(R) = \{\mathbf{m} \mid \exists \mathbf{m}' : (\mathbf{m}', \mathbf{m}) \in R\}$. Given a pair of markings $(\mathbf{m}_1, \mathbf{m}_2)$, we let $\overline{\mathsf{Post}}((\mathbf{m}_1, \mathbf{m}_2)) = \{(\mathbf{m}_1, \mathbf{m}'), (\mathbf{m}_2, \mathbf{m}') \mid \mathbf{m}' \in \mathsf{Post}(\mathbf{m}_2)\}$ and $\overline{\mathsf{Accel}}((\mathbf{m}_1, \mathbf{m}_2)) = \{(\mathbf{m}_2, \mathsf{AccelPair}(\mathbf{m}_1, \mathbf{m}_2))\}$ if $\mathbf{m}_1 \prec \mathbf{m}_2$; and $\overline{\mathsf{Accel}}((\mathbf{m}_1, \mathbf{m}_2))$ is undefined otherwise. We extend these two functions to sets R of pairs in the following way: $\overline{\mathsf{Post}}(R) = \cup_{(\mathbf{m}_1,\mathbf{m}_2)\in R} \overline{\mathsf{Post}}((\mathbf{m}_1, \mathbf{m}_2))$ and $\overline{\mathsf{Accel}}(R) = \cup_{(\mathbf{m}_1,\mathbf{m}_2)\in R, \mathbf{m}_1 \prec \mathbf{m}_2} \{\overline{\mathsf{Accel}}(\mathbf{m}_1, \mathbf{m}_2)\}$.

Definition of the sequence We are now ready to introduce the covering sequence. We will define the sequence in a way that allows for optimizations. To incorporate those optimizations elegantly, we allow our construction to be helped by an oracle which is a procedure that produces pairs of markings. This oracle potentially allows for the early convergence of the covering sequence. However, we will prove that our sequence converges even if the oracle is trivial (returns always an empty set of pairs of markings). In the next section, we will show that the oracle can be implemented by a recursive call to the covering sequence, by considering ω-markings where the number of ω is increasing as initial ω-markings in the recursive call. This will lead to an efficient procedure as we will see in the next section.

In the following, given a Petri net $\mathcal{N}$ and an initial marking $\mathbf{m}_0$, we call an *oracle* any function $\mathsf{Oracle} : \mathbb{N} \mapsto (\mathbb{N} \cup \{\omega\})^{|P|} \times (\mathbb{N} \cup \{\omega\})^{|P|}$ that returns, for any $i \geq 0$, a set of pairs of ω-markings s.t.

$$\downarrow^{\preccurlyeq}(\mathsf{Post}\,(\mathsf{Flatten}\,(\mathsf{Oracle}\,(i)))) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Flatten}\,(\mathsf{Oracle}\,(i))) \tag{1}$$

and

$$\downarrow^{\preccurlyeq}(\mathsf{Flatten}\,(\mathsf{Oracle}\,(i))) \subseteq \mathsf{Cover}\,(\mathcal{N})\,. \tag{2}$$

Let $\mathcal{N} = \langle P, T \rangle$ be a PN, $\mathbf{m}_0$ be an initial marking, and Oracle be an oracle. Then, the covering sequence of $\mathcal{N}$, noted $\mathsf{CovSeq}\,(\mathcal{N}, \mathbf{m}_0, \mathsf{Oracle})$ is the infinite sequence $(V_i, F_i, O_i)_{i \geq 0}$, defined as follows:

- $V_0 = \emptyset$, $O_0 = \emptyset$ and $F_0 = \{(\mathbf{m}_0, \mathbf{m}_0)\}$;
- For any $i \geq 1$: $O_i = \mathsf{Max}^{\sqsubseteq}\,(O_{i-1} \cup \mathsf{Oracle}\,(i))$;
- For any $i \geq 1$: $V_i = \mathsf{Max}^{\sqsubseteq}\,(V_{i-1} \cup F_{i-1}) \setminus \downarrow^{\sqsubseteq}(O_i)$;
- For any $i \geq 1$: $F_i = \mathsf{Max}^{\sqsubseteq}\left(\overline{\mathsf{Post}}\,(F_{i-1}) \cup \overline{\mathsf{Accel}}\,(F_{i-1})\right) \setminus \downarrow^{\sqsubseteq}(V_i \cup O_i)$.

It is not difficult to see that this sequence enjoys the following three properties:

Lemma 5. *Let $\mathcal{N}$ be a* PN*, $\mathbf{m}_0$ be its initial marking,* Oracle *be an oracle, and let* $\mathsf{CovSeq}\,(\mathcal{N}, \mathbf{m}_0, \mathsf{Oracle}) = (V_i, F_i, O_i)_{i \geq 0}$. *Then, for all $i \geq 0$:*

1. $\overline{\mathsf{Post}}\,(V_i) \cup \overline{\mathsf{Accel}}\,(V_i) \subseteq \downarrow^{\sqsubseteq}(V_i \cup F_i \cup O_i)$*;*
2. $\downarrow^{\preccurlyeq}(\mathsf{Flatten}\,(V_i \cup O_i)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Flatten}\,(V_{i+1} \cup O_{i+1}))$*;*
3. *for all* $(\mathbf{m}_1, \mathbf{m}_2) \in F_i \cup V_i$*:* $\downarrow^{\preccurlyeq}(\mathbf{m}_2) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\,(\mathbf{m}_1))$.

Completeness of the sequence For all marking $\mathbf{m}$ computed by the Karp&Miller algorithm, we show that there exists a finite value k s.t. $\mathbf{m} \in \downarrow^{\preccurlyeq}(\mathsf{Flatten}\,(V_k))$, where k depends on the depth of the node labeled by $\mathbf{m}$ in the Karp&Miller tree:

Lemma 6. *Let $\mathcal{N}$ be a* PN*, $\mathbf{m}_0$ be its initial marking,* Oracle *be an oracle, $\mathcal{T} = \langle N, B, root, \Lambda \rangle$ be the K&M tree of $\mathcal{N}$ and* $\mathsf{CovSeq}\,(\mathcal{N}, \mathbf{m}_0, \mathsf{Oracle}) = (V_i, F_i, O_i)_{i \geq 0}$. *Then $\forall n \in N$: $\forall k \geq \sum_{n' \in \mathsf{Anc}(\mathcal{T}, n)}(|\varsigma\,(n')\,| + 3)$: $\downarrow^{\preccurlyeq}(\Lambda\,(n)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Flatten}\,(V_k \cup O_k))$.*

Proof. Sketch.[1] We show by induction on the depth ℓ of nodes in the Karp&Miller tree that the lemma holds for all $n \in N$. For a node n at depth ℓ, we prove that there exists i and a pair $(\mathbf{m}_1, \mathbf{m}_2) \in V_i \cup O_i$ s.t. $\downarrow^{\preccurlyeq}(\Lambda\,(n)) \subseteq \downarrow^{\preccurlyeq}(\mathbf{m}_2)$, as follows. By induction

[1] The complete proof can be found in appendix.

hypothesis, there is j s.t. $\downarrow^{\preccurlyeq}(\Lambda(n')) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Flatten}(O_j \cup V_j))$, where n' is the father of n. The value i s.t. $\Lambda(n) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Flatten}(O_i \cup V_i))$ depends on j and the length of $\varsigma(n)$. Hence, a second induction on the size of $\varsigma(n)$ is used. That induction is applied in the case where $|\varsigma(n)| > 0$ and allows to prove that there is a pair $(\mathbf{m}_3, \mathbf{m}_4) \in V_{i-1} \cup O_{i-1}$ such that $(\mathsf{M}(n), \mathbf{m}) \sqsubseteq (\mathbf{m}_3, \mathbf{m}_4)$ where $\mathbf{m}$ is such that $\mathsf{M}(n) \xrightarrow{\varsigma(n)} \mathbf{m}$. Once that result is obtained, we have by Lemma 4 that $\Lambda(n) \preccurlyeq \overline{\mathsf{Accel}}((\mathsf{M}(n), \mathbf{m}))$, since $\mathsf{M}(n) \prec \mathbf{m}$, and that $\overline{\mathsf{Accel}}((\mathsf{M}(n), \mathbf{m})) \preccurlyeq \overline{\mathsf{Accel}}((\mathbf{m}_3, \mathbf{m}_4)) = \mathbf{m}'$ by definition of $\overline{\mathsf{Accel}}$ and $\sqsubseteq$. Hence $(\mathbf{m}_4, \mathbf{m}') \in \downarrow^{\sqsubseteq}(O_{i-1} \cup F_{i-1} \cup V_{i-1})$ by Lemma 5.1. Finally, by construction of O_i and V_i, we conclude that $(\mathbf{m}_4, \mathbf{m}') \in \downarrow^{\sqsubseteq}(O_i \cup V_i)$, with $\Lambda(n) \preccurlyeq \mathbf{m}'$, hence $\Lambda(n) \in \downarrow^{\preccurlyeq}(\mathsf{Flatten}(O_i \cup V_i))$. □

As a consequence, the covering sequence is complete:

Corollary 1. *Let $\mathcal{N}$ be a* PN*, $\mathbf{m}_0$ be its initial marking,* Oracle *be an oracle, and* $\mathsf{CovSeq}(\mathcal{N}, \mathbf{m}_0, \mathsf{Oracle}) = (V_i, F_i, O_i)_{i \geq 0}$. *There exists $k \geq 0$ such that for all $k' \geq k$ we have* $\mathsf{Cover}(\mathcal{N}, \mathbf{m}_0) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_{k'} \cup O_{k'}))$.

Soundness of the sequence In order to show that the covering sequence is correct, it remains to show that any marking $\mathbf{m}$ produced by the sequence is s.t. $\downarrow^{\preccurlyeq}(\mathbf{m}) \subseteq \mathsf{Cover}(\mathcal{N}, \mathbf{m}_0)$. For that purpose, we need the two following lemmata:

Lemma 7. *Let $\mathcal{N}$ be a* PN *let A and B be two sets of ω-markings of $\mathcal{N}$. Then,* $\downarrow^{\preccurlyeq}(A) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*(B))$ *implies that* $\downarrow^{\preccurlyeq}(\mathsf{Post}^*(A)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*(B))$.

Lemma 8. *Let $\mathcal{N}$ be a* PN*, $\mathbf{m}_0$ be its initial marking,* Oracle *be an oracle, and let* $\mathsf{CovSeq}(\mathcal{N}, \mathbf{m}_0, \mathsf{Oracle}) = (V_i, F_i, O_i)_{i \geq 0}$. *Then, $\forall i \geq 1$: $\forall \mathbf{m} \in \mathsf{Flatten}(T_i \cup F_i \cup O_i)$,* $\downarrow^{\preccurlyeq}(\mathbf{m}) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*(\mathbf{m}_0))$.

As a consequence, we directly obtain our soundness result:

Corollary 2. *Let $\mathcal{N}$ be a* PN*, $\mathbf{m}_0$ be its initial marking,* Oracle *be an oracle, and* $\mathsf{CovSeq}(\mathcal{N}, \mathbf{m}_0, \mathsf{Oracle}) = (V_i, F_i, O_i)_{i \geq 0}$. *Then, $\forall i \geq 1$,* $\downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_i \cup O_i)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*(\mathbf{m}_0))$.

Corollary 1 and 2 allow us to obtain the next Theorem.

Theorem 2. *Let $\mathcal{N}$ be a* PN*, $\mathbf{m}_0$ be its initial marking,* Oracle *be an oracle, and* $\mathsf{CovSeq}(\mathcal{N}, \mathbf{m}_0, \mathsf{Oracle}) = (V_i, F_i, O_i)_{i \geq 0}$. *Then, there exists $k \geq 0$ such that*

1. *for all $1 \leq i < k$:* $\downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_i \cup O_i)) \subset \downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_{i-1} \cup O_{i-1}))$*;*
2. *for all $i \geq k$:* $\downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_i \cup O_i)) = \mathsf{Cover}(\mathcal{N}, \mathbf{m}_0)$.

Proof. By Corollary 1 and 2, we conclude that there exists at least one $k \in \mathbb{N}$ such that $\downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_k \cup O_k)) = \downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_{k+1} \cup O_{k+1}))$. Let us consider the smallest $k \in \mathbb{N}$ that satisfies that condition and let us prove that $\downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_k \cup O_k)) = \mathsf{Cover}(\mathcal{N})$. Note that by Lemma 5.2 we have for all $0 \leq i < k$: $\downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_i \cup O_i)) \subset \downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_{i+1} \cup O_{i+1}))$.

First, we prove that $\downarrow^{\preccurlyeq}(\mathsf{Flatten}(F_k)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_k \cup O_k))$. By construction, $\downarrow^{\sqsubseteq}(F_k) \subseteq \downarrow^{\sqsubseteq}(V_{k+1} \cup O_{k+1})$. Hence, $\downarrow^{\preccurlyeq}(\mathsf{Flatten}(F_k)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_{k+1} \cup O_{k+1}))$, by definition of $\sqsubseteq$. However, $\downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_{k+1} \cup O_{k+1})) = \downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_k \cup O_k))$, by definition of k.

By Lemma 5.1, $\overline{\mathsf{Post}}\,(V_k) \cup \overline{\mathsf{Accel}}\,(V_k) \subseteq {\downarrow^{\sqsubseteq}}(V_k \cup F_k \cup O_k)$, which implies that ${\downarrow^{\preccurlyeq}}\big(\mathsf{Flatten}\,\big(\overline{\mathsf{Post}}\,(V_k) \cup \overline{\mathsf{Accel}}\,(V_k)\big)\big) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_k \cup F_k \cup O_k))$. In particular, ${\downarrow^{\preccurlyeq}}\big(\mathsf{Flatten}\,\big(\overline{\mathsf{Post}}\,(V_k)\big)\big) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_k \cup F_k \cup O_k))$. Since ${\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(F_k)) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_k \cup O_k))$, we have ${\downarrow^{\preccurlyeq}}\big(\mathsf{Flatten}\,\big(\overline{\mathsf{Post}}\,(V_k)\big)\big) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_k \cup O_k))$. This means that ${\downarrow^{\preccurlyeq}}(\mathsf{Post}\,(\mathsf{Flatten}\,(V_k))) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_k \cup O_k))$. Furthermore, by (1) and definition of O_k, ${\downarrow^{\preccurlyeq}}(\mathsf{Post}\,(\mathsf{Flatten}\,(O_k))) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(O_k))$. We conclude that ${\downarrow^{\preccurlyeq}}(\mathsf{Post}\,(\mathsf{Flatten}\,(V_k \cup O_k))) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_k \cup O_k))$. Then, by Lemma 5.2, and since $\mathbf{m}_0 \in {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_1 \cup O_1))$, we have $\mathbf{m}_0 \in {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_k \cup O_k))$. Hence, ${\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_k \cup O_k))$ is a Post fixpoint that covers $\mathbf{m}_0$. Thus ${\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_k \cup O_k)) \supseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_0))$. Since, by Corollary 2, ${\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(O_k \cup V_k)) \subseteq \mathsf{Cover}\,(\mathcal{N}, \mathbf{m}_0)$, we conclude that ${\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_k \cup O_k)) = \mathsf{Cover}\,(\mathcal{N}, \mathbf{m}_0)$. Finally, by Lemma 5.2 and Corollary 2, ${\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_i \cup O_i)) = {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_k \cup O_k)) = \mathsf{Cover}\,(\mathcal{N})$, for all $i > k$. Hence, the lemma. □

6 Practical implementation

To implement the method in practice, we have to instantiate the oracle. First, note that the *empty oracle*, i.e. $\mathsf{Oracle}\,(i) = \emptyset$ for all $i \geq 1$, is a correct oracle. Indeed, ${\downarrow^{\preccurlyeq}}(\mathsf{Post}\,(\emptyset)) = \emptyset \subseteq {\downarrow^{\preccurlyeq}}(\emptyset)$ and ${\downarrow^{\preccurlyeq}}(\emptyset) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_0))$ for all $\mathbf{m}_0$. Thus, the oracle can be regarded as an optional optimization of our algorithm. Yet, this optimization can be very powerful, as we show now. When using the empty oracle our method performs a breadth first search. In particular, if several accelerations can be applied from an ω-marking $\mathbf{m}$, each of them putting ω's in different places (for instance a first acceleration puts one ω in place p_1 and a second one puts an ω is p_2), then all the possible orders for their application will be investiguated (i.e. first put the ω in place p_1 in $\mathbf{m}$ and then an ω in p_2; and vice-versa). However, all the possible orders lead to the same ω-marking (with an ω in p_1 and p_2) that covers the intermediate ones (where there is one ω either in p_1 or p_2). Thus, in order to improve our method, only one possible order should be explored. To achieve that goal, we present in the next paragraph the $\mathsf{CovProc}$ procedure where the oracle is implemented as a recursive call on ω-markings resulting from an acceleration. As a consequence, the initial breadth first search is mixed with a depth first search that allows to develop first the ω-markings resulting from an acceleration.

The $\mathsf{CovProc}$ procedure The $\mathsf{CovProc}$ procedure is shown in Algorithm 3. It closely follows the definition of the covering sequence. At each step i, the oracle is implemented as a finite number of recursive calls to $\mathsf{CovProc}$, where the initial ω-markings are the results of the accelerations occurring at this step. Note that a recursive call is not applied on all the accelerated ω-markings but a non-deterministically chosen subset S. Indeed, in practice, if we have two accelerated markings $\mathbf{m}_1$ and $\mathbf{m}_2$ with ${\downarrow^{\preccurlyeq}}(\mathbf{m}_2) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_1))$ then it is not necessary to apply $\mathsf{CovProc}$ on $\mathbf{m}_2$ to explore the markings that are reachable from $\mathbf{m}_2$.

This strategy allows to mix the breadth-first exploration of the covering sequence and the depth-first exploration due to the recursive calls which favor ω-markings with more ω. It turns out to be very efficient in practice (see hereunder). Since, for any pair $(\mathbf{m}_1, \mathbf{m}_2)$, $\mathsf{AccelPair}\,(\mathbf{m}_1, \mathbf{m}_2)$ contains strictly more ω's than $\mathbf{m}_1$ and $\mathbf{m}_2$, and since the number of places of the PN is bounded, the depth of recursion is bounded too, which ensures termination.

Data: A PN $\mathcal{N} = \langle P, T \rangle$, an initial ω-marking $\mathbf{m}_0$
Result: A set of pairs of markings.
$\mathsf{CovProc}\,(\mathcal{N}, \mathbf{m}_0)$
begin
  $i := 0$; $\overline{O}_0 := \emptyset$; $\overline{V}_0 := \emptyset$; $\overline{F}_0 := \{(\mathbf{m}_0, \mathbf{m}_0)\}$;
  repeat
    $i := i + 1$;
    $R_i := \cup_{\mathbf{m} \in S} \mathsf{CovProc}\,(\mathcal{N}, \mathbf{m})$ where $S \subseteq \mathsf{Flatten}\left(\overline{\mathsf{Accel}}\,(F_{i-1})\right)$;
    $\overline{O}_i := \mathsf{Max}^{\sqsubseteq}\left(\overline{O}_{i-1} \cup R_i\right)$;
    $\overline{V}_i := \mathsf{Max}^{\sqsubseteq}\left(\overline{V}_{i-1} \cup \overline{F}_{i-1}\right) \setminus {\downarrow^{\sqsubseteq}}(\overline{O}_i)$;
    $\overline{F}_i := \mathsf{Max}^{\sqsubseteq}\left(\overline{\mathsf{Post}}\left(\overline{F}_{i-1}\right) \cup \overline{\mathsf{Accel}}\left(\overline{F}_{i-1}\right)\right) \setminus {\downarrow^{\sqsubseteq}}(\overline{O}_i \cup \overline{V}_i)$;
  until ${\downarrow^{\preccurlyeq}}\left(\mathsf{Flatten}\left(\overline{O}_i \cup \overline{V}_i\right)\right) \subseteq {\downarrow^{\preccurlyeq}}\left(\mathsf{Flatten}\left(\overline{O}_{i-1} \cup \overline{V}_{i-1}\right)\right)$;
  `return` $(\overline{O}_i \cup \overline{V}_i)$;
end

Algorithm 3: The $\mathsf{CovProc}$ algorithm.

Let us show that this solution is correct and terminates. For any marking $\mathbf{m}$, we let $\mathsf{Nb}\omega\,(\mathbf{m}) = |\{p \mid \mathbf{m}(p) = \omega\}|$, i.e. the number of unbounded places in $\mathbf{m}$. We first state the following technical lemma:

Lemma 9. *Let $\mathcal{N}$ be a* PN*, $\mathbf{m}_0$ be a ω-marking, and let $\overline{F}_i$ be the sets computed by* $\mathsf{CovProc}\,(\mathcal{N}, \mathbf{m}_0)$*. Then, for any $i \geq 0$: for any $\mathbf{m} \in \mathsf{Flatten}\left(\overline{F}_i\right)$:* $\mathsf{Nb}\omega\,(\mathbf{m}) \geq \mathsf{Nb}\omega\,(\mathbf{m}_0)$.

Then, the proof of total correctness of $\mathsf{CovProc}$ is as follows:

Theorem 3. *For any* PN *$\mathcal{N}$ and any ω-marking $\mathbf{m}_0$:* $\mathsf{CovProc}\,(\mathcal{N}, \mathbf{m}_0)$ *terminates and* ${\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(\mathsf{CovProc}\,(\mathcal{N}, \mathbf{m}_0))) = \mathsf{Cover}\,(\mathcal{N}, \mathbf{m}_0)$.

Proof. The proof works by induction on $\mathsf{Nb}\omega\,(\mathbf{m}_0)$.
Base case ($\mathsf{Nb}\omega\,(\mathbf{m}_0) = |P|$) In that case, $\mathsf{CovProc}\,(\mathcal{N}, \mathbf{m}_0)$ finishes after two iterations and returns $\overline{O}_2 \cup \overline{V}_2 = \{(\mathbf{m}_0, \mathbf{m}_0)\}$. Remark that no recursive call is performed because $R_1 = \overline{\mathsf{Accel}}\left(\overline{F}_0\right) = \emptyset$ and $R_2 = \overline{\mathsf{Accel}}\,(\{(\mathbf{m}_0, \mathbf{m}_0)\}) = \emptyset$. Moreover, ${\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(\mathsf{CovProc}\,(\mathcal{N}, \mathbf{m}_0))) = {\downarrow^{\preccurlyeq}}(\mathbf{m}_0) = \mathsf{Cover}\,(\mathcal{N}, \mathbf{m}_0)$.
Inductive case ($\mathsf{Nb}\omega\,(\mathbf{m}_0) = k < |P|$) We consider two cases. First, assume that the algorithm terminates after ℓ iterations, i.e., assume that ${\downarrow^{\preccurlyeq}}\left(\mathsf{Flatten}\left(\overline{O}_\ell \cup \overline{V}_\ell\right)\right) = {\downarrow^{\preccurlyeq}}\left(\mathsf{Flatten}\left(\overline{O}_{\ell-1} \cup \overline{V}_{\ell-1}\right)\right)$, but for any $1 \leq j \leq \ell - 1$: ${\downarrow^{\preccurlyeq}}\left(\mathsf{Flatten}\left(\overline{O}_j \cup \overline{V}_j\right)\right) \neq {\downarrow^{\preccurlyeq}}\left(\mathsf{Flatten}\left(\overline{O}_{j-1} \cup \overline{V}_{j-1}\right)\right)$. By Lemma 9, for any $1 \leq j \leq \ell$, for any $(\mathbf{m}_1, \mathbf{m}_2) \in \overline{F}_j$: $\mathsf{Nb}\omega\,(\mathbf{m}_2) \geq k$. Hence, for any $1 \leq j \leq \ell$, for any $\mathbf{m} \in \mathsf{Flatten}\left(\overline{\mathsf{Accel}}\left(\overline{F}_j\right)\right)$: $\mathsf{Nb}\omega\,(\mathbf{m}) \geq k + 1$. Thus, by induction hypothesis, for any $1 \leq j \leq \ell$, for any $\mathbf{m} \in \mathsf{Flatten}\left(\overline{\mathsf{Accel}}\left(\overline{F}_j\right)\right)$, $\mathsf{CovProc}\,(\mathcal{N}, \mathbf{m})$ terminates and returns a set of pairs such that ${\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(\mathsf{CovProc}\,(\mathcal{N}, \mathbf{m}))) = \mathsf{Cover}\,(\mathcal{N}, \mathbf{m})$. As a consequence, and since $\mathsf{Flatten}\left(\overline{\mathsf{Accel}}\left(\overline{F}_j\right)\right)$ is a finite set for any $1 \leq j \leq \ell$, we conclude that R_j is computed in a finite amount of time and that ${\downarrow^{\preccurlyeq}}(\mathsf{Post}\,(\mathsf{Flatten}\,(R_j))) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(R_j)) \subseteq \mathsf{Cover}\,(\mathcal{N}, \mathbf{m}_0)$, for any $1 \leq j \leq \ell$.

Let Ω denote the function s.t., for any $1 \leq j \leq \ell$: $\Omega(j) = R_j$, and, for any $j > \ell$, $\Omega(j) = \emptyset$. Thus, Ω is an oracle. Let us assume that $\mathsf{CovSeq}\,(\mathcal{N}, \mathbf{m}_0, \Omega) =$

$(V_i, F_i, O_i)_{i\geq 1}$. Clearly, for any $0 \leq j \leq \ell$, $\overline{V}_j = V_j$ and $\overline{O}_j = O_j$. Thus, by Theorem 2, there exists k s.t. $\downarrow^{\preccurlyeq}\big(\mathsf{Flatten}\left(\overline{V}_{k-1} \cup \overline{O}_{k-1}\right)\big) = \downarrow^{\preccurlyeq}\big(\mathsf{Flatten}\left(\overline{V}_k \cup \overline{O}_k\right)\big) = \mathsf{Cover}\left(\mathcal{N}, \mathbf{m}_0\right)$, and s.t. for every $1 \leq j \leq k-1$: $\downarrow^{\preccurlyeq}\big(\mathsf{Flatten}\left(\overline{V}_{j-1} \cup \overline{O}_{j-1}\right)\big) \subset \downarrow^{\preccurlyeq}\big(\mathsf{Flatten}\left(\overline{V}_j \cup \overline{O}_j\right)\big)$. Hence, $k = \ell$, and we conclude that $\mathsf{CovProc}\left(\mathcal{N}, \mathbf{m}_0\right)$ terminates and returns $\downarrow^{\preccurlyeq}\big(\mathsf{Flatten}\left(\overline{V}_\ell \cup \overline{O}_\ell\right)\big) = \mathsf{Cover}\left(\mathcal{N}, \mathbf{m}_0\right)$.

In the latter case, we assume that the algorithm does not terminate and derive a contradiction. This can happen for two reasons: either because the test of the **repeat** loop is never fulfilled, or because some step j of the loop takes an infinite time to complete. By re-using the arguments of the first part of this proof, we can show that the latter is not possible. Indeed, R_j is computed in a finite amount of time and the functions $\mathsf{Max}^{\sqsubseteq}$, $\mathsf{Flatten}$, $\overline{\mathsf{Post}}$, $\overline{\mathsf{Accel}}$, and the test that guards the "until" are computable, i.e. the computation of the sets $\overline{O}_j$, $\overline{V}_j$ and $\overline{F}_j$ always takes a finite amount of time. Thus, if the algorithm does not terminates, it computes an infinite sequence of sets $(\overline{V}_i, \overline{F}_i, \overline{O}_i)_{i\geq 0}$. Symmetrically to the first part of this proof, we build an oracle Ω s.t. $\Omega(j) = R_j$ for any $j \geq 1$. Let us assume that $\mathsf{CovSeq}\left(\mathcal{N}, \mathbf{m}_0, \Omega\right) = (V_i, F_i, O_i)_{i\geq 0}$. Clearly, for any $j \geq 0$, we have $V_j = \overline{V}_j$ and $O_j = \overline{O}_j$. Hence, by Theorem 2, we conclude that there exists $k \geq 1$ s.t. $\downarrow^{\preccurlyeq}\big(\mathsf{Flatten}\left(\overline{V}_k \cup \overline{O}_k\right)\big) = \downarrow^{\preccurlyeq}\big(\mathsf{Flatten}\left(\overline{V}_{k-1} \cup \overline{O}_{k-1}\right)\big)$, which compels the algorithm to terminate at step k. Contradiction. $\square$

Empirical evaluation We have implemented a prototype that computes the coverability set of a PN, thanks to the covering sequence method and the Karp&Miller algorithm. We have selected five bounded PN and eight unbounded PN. Those examples describes (mutual exclusion) protocols (bounded PN), parameterized systems and communication protocols (unbounded PN). The prototype has been written in the PYTHON programming language in a very straightforward way. As a consequence the running times of the prototype are given for the sake of comparison only. Nevertheless, as can be seen in Table 1, this prototype performs very well on our set of examples[2].

More precisely, we have compared two implementations of the covering sequence to the KM algorithm. The former (column **Cov. Seq. w/o oracle**) is the covering sequence where we let $\mathsf{Oracle}\,(i) = \emptyset$ for any $i \geq 0$ (that is, the oracle-based optimization is disabled). In that case the sets of pairs built by our algorithm are small (see column Max P.) compared to the size of the Karp& Miller tree (column Nodes), although the number of pairs created by the algorithm is not dramatically small compared to the size of the K&M tree (see column Tot. P.). This shows the efficiency of our approach based on *pairs* of markings (and on the $\sqsubseteq$ order), with respect to the classical approach.

The latter implementation is the CovProc procedure (Algorithm 3) where we have implemented the oracle as follows: we consider the accelerated ω-markings one by one. If an accelerated ω-marking is already covered by the Flatten of the pairs computed by previous recursive calls, it is forgotten; otherwise a recursive call is applied on it. In the case of *bounded* PN, CovProc performs as the covering sequence with trivial oracle, which is not surprising since no acceleration occur on these examples. In the case of *unbounded* PN, the optimization based on the oracle turns to be useful. Indeed, the sets of pairs built by CovProc are much smaller than the respective K&M trees, and negligible with respect to the sets built with the trivial oracle. Hence, the CovProc procedure terminates within 20 minutes with reasonable execution time (and memory

[2] See `http://www.ulb.ac.be/di/ssd/ggeeraer/eec` for a complete description.

Table 1. Empirical evaluation of the covering sequence. Experiments on an INTEL XEON 3GHZ. Times in seconds ($\times$ = no result within 20 minutes). P = number of places; T = number of transitions; MCS = size of the minimal coverability set ; Tp = Bounded or Unbounded PN; Max P. = $\max\{|V_i \cup O_i \cup F_i|, i \geq 1\}$; Tot. P. = total number of pairs created along the whole execution

Example					KM		**Cov. Seq. w/o Oracle**			CovProc		
Name	P	T	MCS	Tp	Nodes	Time	Max P.	Tot. P.	Time	Max P.	Tot. P.	Time
RTP	9	12	9	B	16	0.18	47	47	0.10	47	47	0.13
lamport	11	9	14	B	83	0.18	115	115	0.17	115	115	0.17
peterson	14	12	20	B	609	2.19	170	170	0.21	170	170	0.25
dekker	16	14	40	B	7,936	258.95	765	765	1.13	765	765	1.03
readwrite	13	9	41	B	11,139	529.91	1,103	1,103	1.43	1,103	1,103	1.75
manuf.	13	6	1	U	32	0.19	9	101	0.18	2	47	0.14
kanban	16	16	1	U	9,839	1221.96	593	9,855	95.05	4	110	0.19
basicME	5	4	3	U	5	0.10	5	5	0.12	5	5	0.12
CSM	14	13	16	U	$>2.40 \cdot 10^6$	$\times$	371	3,324	14.38	178	248	0.34
FMS	22	20	24	U	$>6.26 \cdot 10^5$	$\times$	$>4{,}460$	$\times$	$\times$	477	866	2.10
PNCSA	31	36	80	U	$>1.02 \cdot 10^6$	$\times$	$>5{,}896$	$\times$	$\times$	2,617	13,408	113.79
multipoll	18	21	220	U	$>1.16 \cdot 10^6$	$\times$	$>7{,}396$	$\times$	$\times$	14,034	14,113	365.90
mesh2x2	32	32	256	U	$>8.03 \cdot 10^5$	$\times$	$>6{,}369$	$\times$	$\times$	10,483	12,735	330.95

consumption) and outperforms the covering sequence with trivial oracle. Finally, the execution times of CovProc are several order of magnitudes smaller than those of the KM procedure, showing the interest of our new algorithm.

References

1. Karp, R.M., Miller, R.E.: Parallel Program Schemata. JCSS **3** (1969) 147–195
2. Finkel, A.: The minimal coverability graph for petri nets. In: ATPN (1991) 210–243
3. Grahlmann, B.: The pep tool. In Grumberg, O., ed.: CAV. Volume 1254 of LNCS, Springer (1997) 440–443
4. Petri, C.A.: Kommunikation mit Automaten. PhD thesis, Tech. University Darmstadt (1962)
5. Reisig, W.: Petri Nets. An introduction. Springer (1986)
6. German, S.M., Sistla, A.P.: Reasoning about Systems with Many Processes. Journal of ACM **39**(3) (1992) 675–735
7. Van Begin, L.: Efficient Verification of Counting Abstractions for Parametric systems. PhD thesis, Université Libre de Bruxelles, Belgium (2003)
8. Luttge, K.: Zustandsgraphen von Petri-Netzen. Master's thesis, Humboldt-Universität zu Berlin (1995)
9. Starke, P.: Personnal communication
10. Geeraerts, G.: Coverability and Expressiveness Properties of Well-structured Transition Systems. PhD thesis, Université Libre de Bruxelles, Belgium (2007)
11. Finkel, A., Geeraerts, G., Raskin, J.F., Van Begin, L.: A counter-example the the minimal coverability tree algorithm. Technical Report 535, Université Libre de Bruxelles (2005)
12. Geeraerts, G., Raskin, J.F., Van Begin, L.: Well-structured languages. Submitted.

A Proof of Lemma 3

Let P' be the set of places $\{p \mid \mathbf{m}_1(p) < \mathbf{m}_2(p)\}$. Remark that, since $\mathbf{m}_1 \preccurlyeq \mathbf{m}_2$, $\mathbf{m}_1(p) = \mathbf{m}_2(p)$ for any $p \notin P'$. Since $\downarrow^{\preccurlyeq}(\mathbf{m}_2) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*(\mathbf{m}_1))$, and $\mathbf{m}_1 \preccurlyeq \mathbf{m}_2$ there must exist a sequence of transitions σ that is firable from $\mathbf{m}_1$ and allows to increase the number of tokens in the places of P'. That is, there exists σ and a marking $\overline{\boldsymbol{m}}$ s.t. (i) $\mathbf{m}_1 \xrightarrow{\sigma} \overline{\boldsymbol{m}}$, (ii) $\mathbf{m}_1 \preccurlyeq \overline{\boldsymbol{m}}$ and (iii) for any place $p \in P'$, $\overline{\boldsymbol{m}}(p) > \mathbf{m}_1(p)$. Indeed, let $\mathbf{m}'$ be defined as follows:

$$\forall p \in P : \mathbf{m}'(p) = \begin{cases} 0 & \text{if } p \notin P' \text{ and } \mathbf{m}_1(p) = \omega \\ \mathbf{m}_1(p) & \text{if } p \notin P' \text{ and } \mathbf{m}_1(p) \neq \omega \\ \mathbf{m}_1(p) + 1 & \text{if } p \in P' \end{cases}$$

By Definition, we have $\mathbf{m}' \in \downarrow^{\preccurlyeq}(\mathbf{m}_2)$ but $\mathbf{m}' \notin \downarrow^{\preccurlyeq}(\mathbf{m}_1)$ (remark in particular that $p \in P'$ implies that $\mathbf{m}_1(p) \neq \omega$ and $\mathbf{m}_2(p) \geq \mathbf{m}_1(p) + 1$). Since $\mathbf{m}' \in \downarrow^{\preccurlyeq}(\mathbf{m}_2) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*(\mathbf{m}_1))$, there exists a marking $\overline{\boldsymbol{m}}$ and a sequence of transitions σ s.t. $\mathbf{m}_1 \xrightarrow{\sigma} \overline{\boldsymbol{m}}$ and $\mathbf{m}' \preccurlyeq \overline{\boldsymbol{m}}$. Hence, $\mathbf{m}'(p) \leq \overline{\boldsymbol{m}}(p)$ for every place p. We consider three cases. (i) when $\mathbf{m}_1(p) = \omega$, we have necessarily $\overline{\boldsymbol{m}}(p) = \omega$. Hence, $\mathbf{m}_1(p) \leq \overline{\boldsymbol{m}}(p)$ for every place p s.t. $\mathbf{m}_1(p) = \omega$. (ii) when $\mathbf{m}_1(p) \neq \omega$ and $p \notin P'$, we have $\mathbf{m}'(p) = \mathbf{m}_1(p)$, by definition of $\mathbf{m}'$. Hence $\mathbf{m}_1(p) \leq \overline{\boldsymbol{m}}(p)$. (iii) when $p \in P'$ (hence $\mathbf{m}_1(p) \neq \omega$), we have $\mathbf{m}_1(p) < \mathbf{m}'(p)$, by definition of $\mathbf{m}'$ again. Hence $\mathbf{m}_1(p) < \overline{\boldsymbol{m}}(p)$. We conclude that $\mathbf{m}_1 \preccurlyeq \overline{\boldsymbol{m}}$ and that $\mathbf{m}_1(p) < \overline{\boldsymbol{m}}(p)$ for every $p \in P'$.

Let $\overline{\boldsymbol{m}}_i$ $(i \geq 1)$ be the marking s.t. $\mathbf{m}_1 \xrightarrow{\sigma^i} \overline{\boldsymbol{m}}_i$, i.e. the marking obtained after having fired i times σ from $\mathbf{m}_1$. Thus, since PN transitions have constant effect,

$$\forall i \geq 1 : \forall p : \overline{\boldsymbol{m}}_i(p) = \mathbf{m}_1(p) + i \cdot (\overline{\boldsymbol{m}}(p) - \mathbf{m}_1(p)) \tag{3}$$

Remark that, for any $i \geq 1 : \overline{\boldsymbol{m}}_i \in \mathsf{Post}^*(\mathbf{m}_1)$, and that, by monotonicity, $\forall i \geq 1 : \overline{\boldsymbol{m}}_i \preccurlyeq \overline{\boldsymbol{m}}_{i+1}$.

In the case where $p \in P'$, the value $\overline{\boldsymbol{m}}(p) - \mathbf{m}_1(p)$ is > 0. Hence, by (3), we have:

$$\forall p \in P' : \forall n \in \mathbb{N} : \exists k : \overline{\boldsymbol{m}}_k(p) > n \tag{4}$$

On the other hand, by definition of the acceleration function, and since $\overline{\boldsymbol{m}}_i \succcurlyeq \mathbf{m}_1$ for any $i \geq 1$:

$$\forall p \notin P' : \forall i \geq 1 : \overline{\boldsymbol{m}}_i(p) \geq \mathbf{m}_1(p) = \mathbf{m}_2(p) = \mathsf{AccelPair}(\mathbf{m}_1, \mathbf{m}_2)(p) \tag{5}$$

Let $\mathbf{m}$ be in $\downarrow^{\preccurlyeq}(\mathsf{AccelPair}(\mathbf{m}_1, \mathbf{m}_2))$. Thus, $\mathbf{m} \preccurlyeq \mathsf{AccelPair}(\mathbf{m}_1, \mathbf{m}_2)$ and for any place p: $\mathbf{m}(p) \neq \omega$. Hence, by (5), for any $p \notin P'$, for any $i \geq 1$, $\mathbf{m}(p) \leq \overline{\boldsymbol{m}}_i(p)$. Moreover, by (4), there exists, for any $p \in P'$, a value $k(p)$ s.t. $\overline{\boldsymbol{m}}_{k(p)}(p) > \mathbf{m}(p)$. Since the sequence $\overline{\boldsymbol{m}}_1, \overline{\boldsymbol{m}}_2, \ldots$ is $\preccurlyeq$-increasing, the marking $\overline{\boldsymbol{m}}_k$, with $k = \max\{k(p) \mid p \in P'\}$ is s.t. for any $p \in P' : \overline{\boldsymbol{m}}_k(p) \geq \mathbf{m}(p)$. We conclude that there exists $k \geq 1$ with $\overline{\boldsymbol{m}}_k \succcurlyeq \mathbf{m}$. Since $\overline{\boldsymbol{m}}_k \in \mathsf{Post}^*(\mathbf{m}_1)$, and since $\mathbf{m}_2 \succcurlyeq \mathbf{m}_1$, there exists, by monotonicity, a marking $\mathbf{m}'$ s.t. $\mathbf{m}' \in \mathsf{Post}^*(\mathbf{m}_2)$ and $\mathbf{m}' \succcurlyeq \overline{\boldsymbol{m}}_k \succcurlyeq \mathbf{m}$. Since this is true for any $\mathbf{m} \in \downarrow^{\preccurlyeq}(\mathsf{AccelPair}(\mathbf{m}_1, \mathbf{m}_2))$, we conclude that: $\downarrow^{\preccurlyeq}(\mathsf{AccelPair}(\mathbf{m}_1, \mathbf{m}_2)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*(\mathbf{m}_2))$. $\square$

B Proof of Lemma 5

First remark that, by definition of $\downarrow^{\sqsubseteq}$ and $\mathsf{Max}^{\sqsubseteq}$, the following holds for any set of pairs S: $\downarrow^{\sqsubseteq}(S) = \downarrow^{\sqsubseteq}\Big(\mathsf{Max}^{\sqsubseteq}(S)\Big)$. Then, we prove the three properties independently.

B.1 for any $i \geq 0$: $\overline{\mathsf{Post}}(V_i) \cup \overline{\mathsf{Accel}}(V_i) \subseteq \downarrow^{\sqsubseteq}(V_i \cup F_i \cup O_i)$

The proof is by induction on i.
Base case ($i = 0$) $V_0 = \emptyset$ implies that $\overline{\mathsf{Post}}(V_0) = \emptyset$ and $\overline{\mathsf{Accel}}(V_0) = \emptyset$. We conclude that $\overline{\mathsf{Post}}(V_0) \cup \overline{\mathsf{Accel}}(V_0) = \emptyset \subseteq \downarrow^{\sqsubseteq}(V_0 \cup F_0 \cup O_0)$.
Inductive case ($i > 0$) Let us consider a pair $(\mathbf{m}, \mathbf{m}') \in V_i$ and let us show that $\overline{\mathsf{Post}}((\mathbf{m}, \mathbf{m}')) \cup \overline{\mathsf{Accel}}((\mathbf{m}, \mathbf{m}')) \subseteq \downarrow^{\sqsubseteq}(V_i \cup F_i \cup O_i)$. We consider two cases: either $(\mathbf{m}, \mathbf{m}') \in V_{i-1}$ or $(\mathbf{m}, \mathbf{m}') \in F_{i-1} \setminus V_{i-1}$.

In the *first case*, $(\mathbf{m}, \mathbf{m}') \in V_{i-1}$. By induction hyp., $\overline{\mathsf{Post}}(V_{i-1}) \cup \overline{\mathsf{Accel}}(V_{i-1}) \subseteq \downarrow^{\sqsubseteq}(V_{i-1} \cup F_{i-1} \cup O_{i-1})$. Note that $\downarrow^{\sqsubseteq}(V_{i-1} \cup F_{i-1} \cup O_{i-1}) = \downarrow^{\sqsubseteq}(V_{i-1} \cup F_{i-1}) \cup \downarrow^{\sqsubseteq}(O_{i-1})$. Hence, since $\downarrow^{\sqsubseteq}(O_{i-1}) \subseteq \downarrow^{\sqsubseteq}(O_i)$:

$$\begin{aligned}
&\overline{\mathsf{Post}}(V_{i-1}) \cup \overline{\mathsf{Accel}}(V_{i-1})\\
\subseteq\ &\downarrow^{\sqsubseteq}(V_{i-1} \cup F_{i-1}) \cup \downarrow^{\sqsubseteq}(O_{i-1})\\
\subseteq\ &\downarrow^{\sqsubseteq}(V_{i-1} \cup F_{i-1}) \cup \downarrow^{\sqsubseteq}(O_i)
\end{aligned}$$

We also have that:

$$\begin{aligned}
&\downarrow^{\sqsubseteq}(V_{i-1} \cup F_{i-1}) \cup \downarrow^{\sqsubseteq}(O_i)\\
=\ &\downarrow^{\sqsubseteq}\Big(\mathsf{Max}^{\sqsubseteq}(V_{i-1} \cup F_{i-1})\Big) \cup \downarrow^{\sqsubseteq}(O_i)\\
=\ &\downarrow^{\sqsubseteq}\Big(\mathsf{Max}^{\sqsubseteq}(V_{i-1} \cup F_{i-1}) \setminus \downarrow^{\sqsubseteq}(O_i)\Big) \cup \downarrow^{\sqsubseteq}(O_i)\\
=\ &\downarrow^{\sqsubseteq}(V_i) \cup \downarrow^{\sqsubseteq}(O_i)\\
=\ &\downarrow^{\sqsubseteq}(V_i \cup O_i)
\end{aligned}$$

We conclude that:

$$\overline{\mathsf{Post}}((\mathbf{m}, \mathbf{m}')) \cup \overline{\mathsf{Accel}}((\mathbf{m}, \mathbf{m}')) \subseteq \downarrow^{\sqsubseteq}(V_i \cup O_i) \subseteq \downarrow^{\sqsubseteq}(V_i \cup F_i \cup O_i)$$

In the *second case*, $(\mathbf{m}, \mathbf{m}') \notin V_{i-1}$ but $(\mathbf{m}, \mathbf{m}') \in F_{i-1}$. We have:

$$\begin{aligned}
&\downarrow^{\sqsubseteq}(V_i \cup F_i \cup O_i)\\
=\ &\downarrow^{\sqsubseteq}(V_i) \cup \downarrow^{\sqsubseteq}(F_i) \cup \downarrow^{\sqsubseteq}(O_i)\\
=\ &\downarrow^{\sqsubseteq}(V_i) \cup \downarrow^{\sqsubseteq}\Big(\mathsf{Max}^{\sqsubseteq}\big(\big(\overline{\mathsf{Post}}(F_{i-1}) \cup \overline{\mathsf{Accel}}(F_{i-1})\big) \setminus \downarrow^{\sqsubseteq}(V_i \cup O_i)\big)\Big) \cup \downarrow^{\sqsubseteq}(O_i)
\end{aligned}$$

Furthermore, we have that:

$$\begin{aligned}
&\downarrow^{\sqsubseteq}\Big(\mathsf{Max}^{\sqsubseteq}\big(\big(\overline{\mathsf{Post}}(F_{i-1}) \cup \overline{\mathsf{Accel}}(F_{i-1})\big) \setminus \downarrow^{\sqsubseteq}(V_i \cup O_i)\big)\Big)\\
=\ &\downarrow^{\sqsubseteq}\big(\big(\overline{\mathsf{Post}}(F_{i-1}) \cup \overline{\mathsf{Accel}}(F_{i-1})\big) \setminus \downarrow^{\sqsubseteq}(V_i \cup O_i)\big)
\end{aligned}$$

Hence, it follows that:

$$\begin{aligned}
&\downarrow^{\sqsubseteq}(V_i \cup F_i \cup O_i)\\
=\ &\downarrow^{\sqsubseteq}(V_i) \cup \downarrow^{\sqsubseteq}\big(\big(\overline{\mathsf{Post}}(F_{i-1}) \cup \overline{\mathsf{Accel}}(F_{i-1})\big) \setminus \downarrow^{\sqsubseteq}(V_i \cup O_i)\big) \cup \downarrow^{\sqsubseteq}(O_i)\\
=\ &\downarrow^{\sqsubseteq}\big(\big(\overline{\mathsf{Post}}(F_{i-1}) \cup \overline{\mathsf{Accel}}(F_{i-1})\big) \setminus \downarrow^{\sqsubseteq}(V_i \cup O_i) \cup V_i \cup O_i\big)\\
=\ &\downarrow^{\sqsubseteq}\big(V_i \cup \overline{\mathsf{Post}}(F_{i-1}) \cup \overline{\mathsf{Accel}}(F_{i-1}) \cup O_i\big)
\end{aligned}$$

Hence, for all $(\mathbf{m}_1, \mathbf{m}_1') \in \overline{\mathsf{Post}}((\mathbf{m}, \mathbf{m}')) \cup \overline{\mathsf{Accel}}((\mathbf{m}, \mathbf{m}'))$ we have $(\mathbf{m}_1, \mathbf{m}_1') \in {\downarrow^{\sqsubseteq}}(V_i \cup F_i \cup O_i)$. We conclude that $\overline{\mathsf{Post}}(V_i) \cup \overline{\mathsf{Accel}}(V_i) \subseteq {\downarrow^{\sqsubseteq}}(V_i \cup F_i \cup O_i)$. □

B.2 for any $i \geq 0$**:** ${\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_i \cup O_i)) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_{i+1} \cup O_{i+1}))$

By definition, $\forall i \geq 1 : V_i \cup O_i = \Big(\mathsf{Max}^{\sqsubseteq}\,(V_{i-1} \cup F_{i-1}) \setminus {\downarrow^{\sqsubseteq}}(O_i)\Big) \cup O_i$. Hence:

$$\begin{aligned} & {\downarrow^{\sqsubseteq}}(V_i \cup O_i) \\ = & {\downarrow^{\sqsubseteq}}\Big(\Big((\mathsf{Max}^{\sqsubseteq}\,(V_{i-1} \cup F_{i-1})) \setminus {\downarrow^{\sqsubseteq}}(O_i)\Big) \cup O_i\Big) \\ = & {\downarrow^{\sqsubseteq}}\Big(\mathsf{Max}^{\sqsubseteq}\,(V_{i-1} \cup F_{i-1}) \cup O_i\Big) \end{aligned}$$

Furthermore,

$${\downarrow^{\sqsubseteq}}\Big(\mathsf{Max}^{\sqsubseteq}\,(V_{i-1} \cup F_{i-1}) \cup O_i\Big) = {\downarrow^{\sqsubseteq}}(V_{i-1} \cup F_{i-1} \cup O_i)$$

Hence, ${\downarrow^{\sqsubseteq}}(V_i \cup O_i) = {\downarrow^{\sqsubseteq}}(V_{i-1} \cup F_{i-1} \cup O_i) \supseteq {\downarrow^{\sqsubseteq}}(V_{i-1} \cup O_{i-1})$, because $O_{i-1} \subseteq O_i$. It follows that for all pair $(\mathbf{m}_1, \mathbf{m}_1') \in V_{i-1} \cup O_{i-1}$ there exists a pair $(\mathbf{m}_2, \mathbf{m}_2') \in V_i \cup O_i$ such that $(\mathbf{m}_1, \mathbf{m}_1') \sqsubseteq (\mathbf{m}_2, \mathbf{m}_2')$; and for all $\mathbf{m} \in \mathsf{Flatten}\,(V_{i-1} \cup O_{i-1})$ there is $\mathbf{m}' \in \mathsf{Flatten}\,(V_i \cup O_i)$ such that $\mathbf{m} \preccurlyeq \mathbf{m}'$. Hence, ${\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_i \cup O_i)) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_{i+1} \cup O_{i+1}))$ for all $i \geq 0$. □

B.3 for any $i \geq 1$**, for any** $(\mathbf{m}_1, \mathbf{m}_2) \in F_i \cup V_i$**:** ${\downarrow^{\preccurlyeq}}(\mathbf{m}_2) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_1))$

The proof is by induction on i.

Base case ($i = 1$**)** $F_0 = \{(\mathbf{m}_0, \mathbf{m}_0)\}$ by definition. Hence, $V_1 = \{(\mathbf{m}_0, \mathbf{m}_0)\}$ and $\mathbf{m}_0 \in \mathsf{Post}^*\,(\mathbf{m}_0)$. Thus, ${\downarrow^{\preccurlyeq}}(\mathbf{m}_0) \in {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_0))$ since ${\downarrow^{\preccurlyeq}}$ is monotonic. For any $(\mathbf{m}_1, \mathbf{m}_2) \in F_1$, we have $(\mathbf{m}_1, \mathbf{m}_2) \in \overline{\mathsf{Post}}((\mathbf{m}_0, \mathbf{m}_0))$. Hence, for any $(m_1, m_2) \in F_i$, $\mathbf{m}_1 = \mathbf{m}_0$ and $\{\mathbf{m}_2\} \in \mathsf{Post}\,(\mathbf{m}_0)$. It follows that $\{\mathbf{m}_2\} \subseteq \mathsf{Post}^*\,(\mathbf{m}_1)$, hence ${\downarrow^{\preccurlyeq}}(\mathbf{m}_2) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_1))$ by $\subseteq$-monotony of ${\downarrow^{\preccurlyeq}}$.

Inductive case ($i = k+1$**)** By construction, $(\mathbf{m}_1, \mathbf{m}_2) \in V_{k+1}$ implies that $(\mathbf{m}_1, \mathbf{m}_2) \in V_k \cup F_k$. By induction hypothesis we conclude that ${\downarrow^{\preccurlyeq}}(\mathbf{m}_2) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_1))$. For any $(\mathbf{m}_1, \mathbf{m}_2) \in F_k$: ${\downarrow^{\preccurlyeq}}(\mathbf{m}_2) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_1))$. Let us show that the same holds for any $(\mathbf{m}_1, \mathbf{m}_2)$ in F_{k+1}. We consider two cases:

1. If $(\mathbf{m}_1, \mathbf{m}_2) \in \overline{\mathsf{Accel}}\,(F_k)$, then, there exists $\mathbf{m}_3$ s.t. $(\mathbf{m}_3, \mathbf{m}_1) \in F_k$, $\mathbf{m}_3 \prec \mathbf{m}_1$ and $\mathbf{m}_2 = \mathsf{AccelPair}\,(\mathbf{m}_3, \mathbf{m}_1)$. By induct. hypoth., ${\downarrow^{\preccurlyeq}}(\mathbf{m}_1) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_3))$. By Lemma 3, this implies that ${\downarrow^{\preccurlyeq}}(\mathbf{m}_2) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_1))$.
2. If $(\mathbf{m}_1, \mathbf{m}_2) \in \overline{\mathsf{Post}}\,(F_k)$, then there exists, by construction, a pair $(\mathbf{m}_3, \mathbf{m}_4) \in F_k$ such that $\mathbf{m}_2 \in \mathsf{Post}\,(\mathbf{m}_4)$ and either $\mathbf{m}_3 = \mathbf{m}_1$ or $\mathbf{m}_4 = \mathbf{m}_1$. In the first case, by induction hypothesis ${\downarrow^{\preccurlyeq}}(\mathbf{m}_4) \in {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_1))$, hence ${\downarrow^{\preccurlyeq}}(\mathsf{Post}\,(\mathbf{m}_4)) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_1))$ by Lemma 7 since ${\downarrow^{\preccurlyeq}}(\mathsf{Post}\,(\mathbf{m}_4)) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_4))$. Finally, $\mathbf{m}_2 \in \mathsf{Post}\,(\mathbf{m}_4)$ by construction. It implies ${\downarrow^{\preccurlyeq}}(\mathbf{m}_2) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}\,(\mathbf{m}_4))$ by monotonicity of ${\downarrow^{\preccurlyeq}}$. We conclude that ${\downarrow^{\preccurlyeq}}(\mathbf{m}_2) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_1))$. In the second case, we have that $\mathbf{m}_2 \in \mathsf{Post}\,(\mathbf{m}_1)$, hence $\{\mathbf{m}_2\} \subseteq \mathsf{Post}^*\,(\mathbf{m}_1)$. Thus, ${\downarrow^{\preccurlyeq}}(\mathbf{m}_2) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Post}^*\,(\mathbf{m}_1))$ by monotonicity of ${\downarrow^{\preccurlyeq}}$. □

C Proof of Lemma 6

The proof is by induction on the length[3] ℓ of the branch ending in n.

Base case ($\ell = 1$) In that case, $n = root$ and $\Lambda(root) = \mathbf{m}_0$. By construction, $V_1 = \{(\mathbf{m}_0, \mathbf{m}_0)\} \setminus {\downarrow^{\preccurlyeq}}(O_1)$, hence $\mathbf{m}_0 \in {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_1 \cup O_1))$. Moreover, we have that $1 \leq \sum_{n' \in \mathsf{Anc}(\mathcal{T}, root)} (|\varsigma(n')| + 3) = |\varsigma(root)| + 3 = 3$.

Inductive case ($\ell > 1$) Let $n_1, n_2, \ldots, n_{\ell-1}, n_\ell$ be a branch of $\mathcal{T}$ (hence, $n_1 = root$) of length ℓ. By induction hypothesis, there exists $k \leq \sum_{j=1}^{\ell-1}(|\varsigma(n_j)| + 3)$ s.t. ${\downarrow^{\preccurlyeq}}(\Lambda(n_{\ell-1})) \subseteq \mathsf{Flatten}\,(T_k \cup O_k)$. We consider two cases: either ${\downarrow^{\preccurlyeq}}(\Lambda(n_{\ell-1})) \subseteq \mathsf{Flatten}\,(O_k)$ or not.

In the first case, ${\downarrow^{\preccurlyeq}}\big(\mathsf{Post}\,\big({\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(\mathsf{Oracle}\,(i)))\big)\big) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(\mathsf{Oracle}\,(i)))$ for all $i \geq 0$, by property of the oracle. From the construction of O_i ($i \geq 0$), we have: ${\downarrow^{\preccurlyeq}}\big(\mathsf{Post}\,\big({\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(O_i))\big)\big) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(O_i))$. As a consequence, we have: ${\downarrow^{\preccurlyeq}}\big(\mathsf{Post}^*\,\big({\downarrow^{\preccurlyeq}}(\Lambda(n_{\ell-1}))\big)\big) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(O_k))$. On the other hand, ${\downarrow^{\preccurlyeq}}(\Lambda(n_\ell)) \subseteq {\downarrow^{\preccurlyeq}}\big(\mathsf{Post}^*\,\big({\downarrow^{\preccurlyeq}}(\Lambda(n_{\ell-1}))\big)\big)$ by prop. of the Karp& Miller tree [1]. Hence, ${\downarrow^{\preccurlyeq}}(\Lambda(n_\ell)) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(O_k))$. Finally, ${\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(O_i)) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(O_{i+1}))$, for any $i \geq 0$ We conclude that:

$$ {\downarrow^{\preccurlyeq}}(\Lambda(n_\ell)) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(O_{k'})) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_{k'} \cup O_{k'})) $$

for all $k' \geq k$.

In the second case, ${\downarrow^{\preccurlyeq}}(\Lambda(n_{\ell-1})) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_k))$. Let us consider the sequence of transitions $\varsigma(n_\ell)$. We consider two cases:

1. In the case where $\varsigma(n_\ell)$ is the empty sequence, there exists a transition t s.t. $\Lambda(n_{\ell-1}) \xrightarrow{t} \Lambda(n_\ell)$. Furthermore, there exists $\overline{\boldsymbol{m}}_{\ell-1} \in \mathsf{Flatten}\,(V_k)$ s.t. $\overline{\boldsymbol{m}}_{\ell-1} \succcurlyeq \Lambda(n_{\ell-1})$ by induction hypothesis. Hence t is firable from $\overline{\boldsymbol{m}}_{\ell-1}$ and $\overline{\boldsymbol{m}}_{\ell-1} \xrightarrow{t} \overline{\boldsymbol{m}}$ implies that $\Lambda(n_\ell) \preccurlyeq \overline{\boldsymbol{m}}$. By Lemma 5.1 and since $\overline{\boldsymbol{m}}_{\ell-1} \in \mathsf{Flatten}\,(V_k)$, we have that $(\overline{\boldsymbol{m}}_{\ell-1}, \overline{\boldsymbol{m}}) \in {\downarrow^{\sqsubseteq}}(V_k \cup F_k \cup O_k) = {\downarrow^{\sqsubseteq}}(V_k \cup F_k) \cup {\downarrow^{\sqsubseteq}}(O_k) \subseteq {\downarrow^{\sqsubseteq}}(V_k \cup F_k) \cup {\downarrow^{\sqsubseteq}}(O_{k+1}) = {\downarrow^{\sqsubseteq}}(V_k \cup F_k \cup O_{k+1})$ since ${\downarrow^{\sqsubseteq}}(O_k) \subseteq {\downarrow^{\sqsubseteq}}(O_{k+1})$. Hence:

   $$ (\overline{\boldsymbol{m}}_{\ell-1}, \overline{\boldsymbol{m}}) \in {\downarrow^{\sqsubseteq}}\Big(\Big(\mathsf{Max}^{\sqsubseteq}\,(V_k \cup F_k) \setminus {\downarrow^{\sqsubseteq}}(O_{k+1})\Big) \cup O_{k+1}\Big) = {\downarrow^{\sqsubseteq}}(V_{k+1} \cup O_{k+1}) $$

   We conclude that $\Lambda(n_\ell) \in {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_{k+1} \cup O_{k+1}))$. Hence ${\downarrow^{\preccurlyeq}}(\Lambda(n_\ell)) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_{k+1} \cup O_{k+1}))$. Moreover, $k+1 \leq k+3 \leq \sum_{j=1}^{\ell-1}(|\varsigma(n_j)|+3)+3 = \sum_{j=1}^{\ell}(|\varsigma(n_j)| + 3)$. Finally, by Lemma 5.2 we conclude that $\forall k' \geq k+1 :$ ${\downarrow^{\preccurlyeq}}(\Lambda(n_\ell)) \subseteq {\downarrow^{\preccurlyeq}}(\mathsf{Flatten}\,(V_{k'} \cup F_{k'}))$.

2. In the case where $\varsigma(n_\ell)$ is not empty, then let $\mathbf{m}'$ be s.t. $\mathsf{M}(n_\ell) \xrightarrow{\varsigma(n_\ell)} \mathbf{m}'$. By Lemma 4, we have $\mathsf{AccelPair}\,(\mathsf{M}(n_\ell), \mathbf{m}') \succcurlyeq \Lambda(n_\ell)$. Let us show, by induction on the length of $\varsigma(n_\ell)$, that either $\Lambda(n_\ell) \in {\downarrow^{\preccurlyeq}}\big(\mathsf{Flatten}\,\big(O_{k+|\varsigma(n_\ell)|+3}\big)\big)$ or there exists in $V_{k+|\varsigma(n_\ell)|+2} \cup O_{k+|\varsigma(n_\ell)|+2}$ a pair $(\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}')$ s.t. $(\mathsf{M}(n_\ell), \mathbf{m}') \sqsubseteq (\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}')$. First remark that there is, in $\mathsf{Flatten}\,(V_{k+1} \cup O_{k+1})$, a marking $\widehat{\boldsymbol{m}}$ s.t. $\mathsf{M}(n_\ell) \preccurlyeq \widehat{\boldsymbol{m}}$, because, by definition of $\mathsf{M}(n_\ell)$, there exists a transition t s.t. $\Lambda(n_{\ell-1}) \xrightarrow{t} \mathsf{M}(n_\ell)$.

[3] The length of a tree's branch is defined as the number of nodes it contains.

Hence, we can invoke the arguments used in point 1 of the present proof. Thus, $\varsigma(n_\ell)$ is firable from $\widehat{\boldsymbol{m}} \in \mathsf{Flatten}(V_{k+1} \cup O_{k+1})$.

Base case ($|\varsigma(n_\ell)| = 1$) In this case, $\varsigma(n_\ell) = t \in T$. Either $\widehat{\boldsymbol{m}} \in \mathsf{Flatten}(O_{k+1})$ or not. In the first case, note that for all $i \geq 1$: $\downarrow^{\preccurlyeq}(\mathbf{m}_i) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*(\widehat{\boldsymbol{m}}))$, where $\mathbf{m}_i$ is the marking s.t. $\widehat{\boldsymbol{m}} \xrightarrow{t^i} \mathbf{m}_i$. Furthermore, $\downarrow^{\preccurlyeq}(\Lambda(n_\ell)) = \bigcup_{i\geq 1} \downarrow^{\preccurlyeq}(\mathbf{m}_i)$ by property of the Karp& Miller tree [1]. Hence, $\downarrow^{\preccurlyeq}(\Lambda(n_\ell)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*(\widehat{\boldsymbol{m}}))$. Following the same reasoning as above, we obtain that $\downarrow^{\preccurlyeq}\big(\mathsf{Post}\left(\downarrow^{\preccurlyeq}(\mathsf{Flatten}(O_{k+1}))\right)\big) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Flatten}(O_{k+1}))$. Since $\widehat{\boldsymbol{m}} \in \downarrow^{\preccurlyeq}(\mathsf{Flatten}(O_{k+1}))$, we have $\downarrow^{\preccurlyeq}(\Lambda(n_\ell)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Flatten}(O_{k+1}))$. Finally, since $\forall i \geq 0 : \downarrow^{\preccurlyeq}(\mathsf{Flatten}(O_i)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Flatten}(O_{i+1}))$ we conclude that $\downarrow^{\preccurlyeq}(\Lambda(n_\ell)) \subseteq \downarrow^{\preccurlyeq}\big(\mathsf{Flatten}\left(O_{k+|\varsigma(n_\ell)|+3}\right)\big)$.

In the other case, we have that $\widehat{\boldsymbol{m}} \in \mathsf{Flatten}(V_{k+1})$. Since t is firable from $\widehat{\boldsymbol{m}}$ and $\widehat{\boldsymbol{m}} \in \mathsf{Flatten}(V_{k+1})$, by Lemma 5.1 we have that the pair $(\widehat{\boldsymbol{m}}, \widehat{\boldsymbol{m}}')$ with $\widehat{\boldsymbol{m}} \xrightarrow{t} \widehat{\boldsymbol{m}}'$ is in $\downarrow^{\sqsubseteq}(V_{k+1} \cup F_{k+1} \cup O_{k+1})$. Remark that $(\mathsf{M}(n_\ell), \mathbf{m}') \sqsubseteq (\widehat{\boldsymbol{m}}, \widehat{\boldsymbol{m}}')$ because PN transitions have constant effects. By construction, there is in $\downarrow^{\sqsubseteq}(V_{k+2} \cup O_{k+2})$ a pair $(\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}')$ s.t. $(\widehat{\boldsymbol{m}}, \widehat{\boldsymbol{m}}') \sqsubseteq (\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}')$, hence $(\mathsf{M}(n_\ell), \mathbf{m}') \sqsubseteq (\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}')$.

Inductive case ($|\varsigma(n_\ell)| = m+1$) Let us assume that $\varsigma(n_\ell) = \sigma \cdot t$, where $|\sigma| = m$. Let $\mathbf{m}''$ be the marking s.t. $\mathsf{M}(n_\ell) \xrightarrow{\sigma} \mathbf{m}'' \xrightarrow{t} \mathbf{m}'$. By induction hypothesis, either $\downarrow^{\preccurlyeq}(\Lambda(n_\ell)) \in \downarrow^{\preccurlyeq}\big(\mathsf{Flatten}\left(O_{k+|\varsigma(n_\ell)|+3}\right)\big)$ or there exists a pair $(\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}'') \in V_{k+m+2}$ s.t. $(\mathsf{M}(n_\ell), \mathbf{m}'') \sqsubseteq (\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}'')$ (hence, $\overline{\boldsymbol{m}}'' \succcurlyeq \mathbf{m}''$). Let us consider the second case. The transition t is firable from $\overline{\boldsymbol{m}}''$. Let $\overline{\boldsymbol{m}}'$ be the marking s.t. $\overline{\boldsymbol{m}}'' \xrightarrow{t} \overline{\boldsymbol{m}}'$. By monotonicity, $\mathbf{m}' \preccurlyeq \overline{\boldsymbol{m}}'$ and by Lemma 5.1 (since $(\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}'') \in V_{k+m+2}$) we have that $(\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}') \in \downarrow^{\sqsubseteq}(V_{k+m+2} \cup F_{k+m+2} \cup O_{k+m+2})$. Furthermore, $(\mathsf{M}(n_\ell), \mathbf{m}') \sqsubseteq (\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}')$ since transitions of PN have constant effect. Hence, $(\mathsf{M}(n_\ell), \mathbf{m}') \in \downarrow^{\sqsubseteq}(V_{k+m+2} \cup F_{k+m+2} \cup O_{k+m+2})$. By construction of V_{k+m+3} and O_{k+m+3}, the pair $(\mathsf{M}(n_\ell), \mathbf{m}'')$ is in $\downarrow^{\sqsubseteq}(V_{k+m+3} \cup O_{k+m+3})$ (with $|\sigma \cdot t| = m+1$).

Thus, either $\Lambda(n_\ell) \in \downarrow^{\preccurlyeq}\big(\mathsf{Flatten}\left(O_{k+|\varsigma(n_\ell)|+3}\right)\big)$ or there is in $V_{k+|\varsigma(n_\ell)|+2}$ a pair $(\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}')$ s.t. $(\mathsf{M}(n_\ell), \mathbf{m}') \sqsubseteq (\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}')$. In the second case, by Lemma 5.1, we have that $(\overline{\boldsymbol{m}}', \overline{\boldsymbol{m}}'') \in \downarrow^{\sqsubseteq}\big(V_{k+|\varsigma(n_\ell)|+2} \cup F_{k+|\varsigma(n_\ell)|+2} \cup O_{k+|\varsigma(n_\ell)|+2}\big)$ s.t. $\overline{\boldsymbol{m}}'' = \mathsf{AccelPair}(\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}')$. By def. of $\sqsubseteq$, $\mathsf{AccelPair}(\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}') \succcurlyeq \mathsf{AccelPair}(\mathsf{M}(n_\ell), \mathbf{m}')$. Moreover, by Lemma 4, $\mathsf{AccelPair}(\mathsf{M}(n_\ell), \mathbf{m}') \succcurlyeq \Lambda(n_\ell)$. By construction of $V_{k+|\varsigma(n_\ell)|+3}$ and $O_{k+|\varsigma(n_\ell)|+3}$, there is $(\widehat{\boldsymbol{m}}', \widehat{\boldsymbol{m}}'')$ in $V_{k+|\varsigma(n_\ell)|+3} \cup O_{k+|\varsigma(n_\ell)|+3}$ s.t. $(\overline{\boldsymbol{m}}', \overline{\boldsymbol{m}}'') \sqsubseteq (\widehat{\boldsymbol{m}}', \widehat{\boldsymbol{m}}'')$. Hence, we have:

$$\Lambda(n_\ell) \preccurlyeq \mathsf{AccelPair}(\mathsf{M}(n_\ell), \mathbf{m}') \preccurlyeq \mathsf{AccelPair}(\overline{\boldsymbol{m}}, \overline{\boldsymbol{m}}') = \overline{\boldsymbol{m}}'' \preccurlyeq \widehat{\boldsymbol{m}}''$$

with $\widehat{\boldsymbol{m}}'' \in \mathsf{Flatten}\left(T_{k+|\varsigma(n_\ell)|+3} \cup O_{k+|\varsigma(n_\ell)|+3}\right)$. Thus, there exists a marking $\widehat{\boldsymbol{m}}$ in $\mathsf{Flatten}\left(T_{k+|\varsigma(n_\ell)|+3} \cup O_{k+|\varsigma(n_\ell)|+3}\right)$ s.t. $\widehat{\boldsymbol{m}} \succcurlyeq \Lambda(n_\ell)$. Moreover, using induction hypothesis, we obtain: $k + |\varsigma(n_\ell)| + 3 \leq \sum_{j=1}^{\ell}(|\varsigma(n_j)| + 3)$. Hence the lemma. Finally, by Lemma 5 we conclude that $\forall k' \geq k + |\varsigma(n_\ell)| + 3$: $\downarrow^{\preccurlyeq}(\Lambda(n_\ell)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Flatten}(V_{k'} \cup O_{k'}))$. □

D Proof of Lemma 7

By monotonicity of Post, and since $\mathbf{m}_2 \in \mathsf{Post}(\mathbf{m}_1)$ implies $\downarrow^{\preccurlyeq}(\mathbf{m}_2) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}(\mathbf{m}_1))$ (see [12, Lemma 7]) we have:

$$\forall X \subseteq \left(\mathbb{N} \cup \{\omega\}\right)^{|P|} : \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(X\right)) = \downarrow^{\preccurlyeq}\!\left(\mathsf{Post}^*\left(\downarrow^{\preccurlyeq}(X)\right)\right) \tag{6}$$

Thus:

$$\begin{array}{lll} & \downarrow^{\preccurlyeq}(A) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(B\right)) & \\ \Rightarrow & \downarrow^{\preccurlyeq}\!\left(\mathsf{Post}^*\left(\downarrow^{\preccurlyeq}(A)\right)\right) \subseteq \downarrow^{\preccurlyeq}\!\left(\mathsf{Post}^*\left(\downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(B\right))\right)\right) & \text{Monotonicity of } \mathsf{Post} \text{ and } \downarrow^{\preccurlyeq} \\ \Rightarrow & \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(A\right)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathsf{Post}^*\left(B\right)\right)) & \text{By (6)} \\ \Rightarrow & \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(A\right)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(B\right)) & \square \end{array}$$

E Proof of Lemma 8

The proof is by induction on i.
Base case ($i = 0$) Trivial.
Inductive case ($i = k{+}1$) By definition of V_i, $(\mathbf{m}_1, \mathbf{m}_2) \in V_i$ implies that $(\mathbf{m}_1, \mathbf{m}_2) \in V_{i-1} \cup F_{i-1}$. By induction hypothesis, we conclude that $\downarrow^{\preccurlyeq}(\mathbf{m}_2) \in \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}_0\right))$. Furthermore, $\mathsf{Flatten}\left(\mathsf{Oracle}\left(i\right)\right) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}_0\right))$, by property of the oracle. Hence, for all $\mathbf{m} \in \downarrow^{\preccurlyeq}(\mathsf{Flatten}\left(O_i\right))$: $\downarrow^{\preccurlyeq}(\mathbf{m}) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}_0\right))$.

We now show that $(\mathbf{m}', \mathbf{m}) \in F_i$ implies that $\downarrow^{\preccurlyeq}(\mathbf{m}) \in \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}_0\right))$. We consider two cases:

1. If $(\mathbf{m}', \mathbf{m}) \in \overline{\mathsf{Post}}\left(F_{i-1}\right)$, then by construction it implies that there exists a pair $(\mathbf{m}_1, \mathbf{m}_2) \in F_{i-1}$ such that either $\mathbf{m}_1 = \mathbf{m}'$ or $\mathbf{m}_2 = \mathbf{m}'$ and $\mathbf{m} \in \mathsf{Post}\left(\mathbf{m}_2\right)$. By induction hypothesis, since $\mathbf{m}_2 \in \downarrow^{\preccurlyeq}(\mathsf{Flatten}\left(F_{i-1}\right))$, we know that $\downarrow^{\preccurlyeq}(\mathbf{m}_2) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}_0\right))$. By Lemma 7, it follows that $\downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}_2\right)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}_0\right))$. Since $\mathbf{m} \in \mathsf{Post}\left(\mathbf{m}_2\right) \subseteq \mathsf{Post}^*\left(\mathbf{m}_2\right)$, we conclude that $\downarrow^{\preccurlyeq}(\mathbf{m}) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}_2\right))$, by monotonicity of $\downarrow^{\preccurlyeq}$. Hence, $\downarrow^{\preccurlyeq}(\mathbf{m}) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}_0\right))$.
2. If $(\mathbf{m}', \mathbf{m}) \in \overline{\mathsf{Accel}}\left(F_{i-1}\right)$, then there exists, by construction, a pair $(\mathbf{m}'', \mathbf{m}') \in F_{i-1}$ such that $\mathbf{m}'' \prec \mathbf{m}'$ and $\mathbf{m} = \mathsf{AccelPair}\left(\mathbf{m}'', \mathbf{m}'\right)$. By Lemma 5.3, $\downarrow^{\preccurlyeq}(\mathbf{m}') \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}''\right))$. Hence, by Lemma 3, we conclude that $\downarrow^{\preccurlyeq}(\mathbf{m}) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}'\right))$. Furthermore, $\downarrow^{\preccurlyeq}(\mathbf{m}') \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}_0\right))$, by induction hypothesis. By Lemma 7, it follows that $\downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}'\right)) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}_0\right))$. We conclude that $\downarrow^{\preccurlyeq}(\mathbf{m}) \subseteq \downarrow^{\preccurlyeq}(\mathsf{Post}^*\left(\mathbf{m}_0\right))$. $\square$

F Proof of Lemma 9

The proof is by induction on i.
Base case ($i = 0$) Trivial.
Inductive case ($i = k > 0$) First remark that, for any marking $\mathbf{m}$, the following holds: for any $\mathbf{m}' \in \mathsf{Post}\left(\mathbf{m}\right)$, $\mathsf{Nb}\omega\left(\mathbf{m}'\right) = \mathsf{Nb}\omega\left(\mathbf{m}\right)$, because PN transitions have constant effect. Moreover, for any pair of markings $(\mathbf{m}_1, \mathbf{m}_2)$ s.t. $\mathbf{m}_1 \prec \mathbf{m}_2$: $\mathsf{Nb}\omega\left(\mathsf{AccelPair}\left(\mathbf{m}_1, \mathbf{m}_2\right)\right) > \mathsf{Nb}\omega\left(\mathbf{m}_2\right)$. Thus, by definition of F_k, for any $\mathbf{m} \in \mathsf{Flatten}\left(F_k\right)$, there exists $\mathbf{m}' \in \downarrow^{\preccurlyeq}(\mathsf{Flatten}\left(F_{k-1}\right))$ s.t. $\mathsf{Nb}\omega\left(\mathbf{m}\right) \geq \mathsf{Nb}\omega\left(\mathbf{m}'\right)$. However, by induction hypothesis, $\mathsf{Nb}\omega\left(\mathbf{m}'\right) \geq \mathsf{Nb}\omega\left(\mathbf{m}_0\right)$ for any $\mathbf{m}' \in \mathsf{Flatten}\left(F_{k-1}\right)$. Hence the lemma. $\square$