Fundamenta Informaticae XX (2014) 1–??
DOI 10.3233/FI-2012-0000
IOS Press

ω-Petri nets: algorithms and complexity

Gilles Geeraerts*
Université libre de Bruxelles, Belgium – Département d'informatique
gigeerae@ulb.ac.be

Alexander Heußner
Otto-Friedrich Universität Bamberg, Germany
alexander.heussner@uni-bamberg.de

M. Praveen
Laboratoire Bordelais de Recherche en Informatique (LaBRI), France
praveen.manjunatha@labri.fr

Jean-François Raskin
Université libre de Bruxelles, Belgium – Département d'informatique
jraskin@ulb.ac.be

Abstract. We introduce ω-Petri nets (ωPN), an extension of *plain* Petri nets with ω-labeled input and output arcs, that is well-suited to analyse *parametric concurrent systems with dynamic thread creation*. Most techniques (such as the Karp and Miller tree or the Rackoff technique) that have been proposed in the setting of *plain Petri nets* do not apply directly to ωPN because ωPN define transition systems that have *infinite branching*. This motivates a thorough analysis of the computational aspects of ωPN. We show that an ωPN can be turned into a plain Petri net that allows us to recover the reachability set of the ωPN, but that does not preserve termination (an ωPN terminates iff it admits no infinitely long execution). This yields complexity bounds for the reachability, (place) boundedness and coverability problems on ωPN. We provide a practical algorithm to compute a coverability set of the ωPN and to decide termination by adapting the classical Karp and Miller tree construction. We also adapt the Rackoff technique to ωPN, to obtain the exact complexity of the termination problem. Finally, we consider the extension of ωPN with reset and transfer arcs, and show how this extension impacts the decidability and complexity of the aforementioned problems.

1. Introduction

In this paper, we introduce ω-Petri nets (ωPN), an extension of *plain* Petri nets (PN) that permits input and output arcs to be labeled by the symbol ω, instead of a natural number. An ω-labeled input arc consumes, non-deterministically, any number of tokens in its input place while an ω-labeled output arc produces non-deterministically any number of tokens in its output place. We claim that ωPN are

*Partially supported by a 'Crédit aux chercheurs' of the F.R.S/FNRS.

```
1 one_task(int k) {
2  // some work...
3 }
4 main(int P) {
5  for i := 1 to P step 1
6   fork(one_task(i))
7 }
```

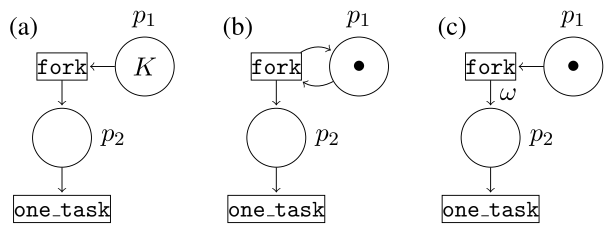

Figure 1. An example of a parametric system with three possible models

particularly well suited for modeling *parametric concurrent systems* — see for instance our recent work on the Grand Central Dispatch (GCD) technology [13] —, and to perform *parametric verification* [16] on those systems, as we illustrate now by means of the example in Fig. 1. The example present a skeleton of a distributed program, in which a `main` function forks P parallel threads (where P is a parameter of the program), each executing the `one_task` function. Many distributed programs follow this abstract skeleton that allows us to perform calculations in parallel, and being able to model precisely such concurrent behaviors is an important issue. In particular, we would like that the model captures the fact that P *is a parameter*, so that we can, for instance, check that the execution of the program always terminates (assuming each individual execution of `one_task` does), *for all possible values of* P. Clearly, the Petri net (a) in Fig. 1 does not capture the parametric nature of the example, as place p_1 contains a fixed number K of tokens. The PN (b), on the other hand captures the fact that the program can `fork` an unbounded number of threads, but *does not preserve termination*: $(\texttt{fork})^\omega$ is an infinite execution of PN (b), while the programme terminates (assuming each `one_task` thread terminates) for all values of P, because the `for` loop in line 5 executes exactly P times. Finally, observe that the ωPN (c) has the desired properties: firing transition `fork` creates *non-deterministically* an *unbounded* albeit *finite* number of tokens in p_2 (to model all the possible executions of the for loop in line 5), and all possible executions of this ωPN terminate, because the number of tokens produced in p_2 remains *finite* and no further token creation in p_2 is allowed after the firing of the `fork` transition.

While close to Petri nets, ωPN are sufficiently different that a thorough and careful study of their computational properties is required. This is the main contribution of the paper. A first example of discrepancy is that the semantics of ωPN is an infinite transition system which is *infinitely branching*. This is not the case for plain PN: their transition systems can be infinite but they are finitely branching. As a consequence, some of the classical techniques for the analysis of Petri nets cannot be applied. Consider for example the *finite unfolding of the transition system* [12] that stops the development of a branch of the reachability tree whenever a node with a smaller ancestor is found. This tree is finite (and can be built effectively) for any plain Petri net and any initial marking because the set of markings $\mathbb{N}^k$ is *well-quasi ordered*, and *finite branching* of plain Petri nets allows for the use of König's lemma[1]. However, this technique cannot be applied to ωPN, as they are infinitely branching. Such peculiarities of ωPN motivate our study of three different tools for analysing them. First, we consider, in Section 3, a variant of the Karp and Miller tree [18] that applies to ωPN. In order to cope with the infinite branching of the semantics of ωPN, we need to introduce in the Karp and Miller tree ω's that are not the result of accelerations but the result of ω-output arcs. Our variant of the Karp and Miller construction is *recursive*,

[1]In fact, this construction is applicable to any well-structured transition system which is finitely branching and permits to decide the termination problem for example.

Table 1. Complexity results on ωPN (with the section numbers where the results are proved). ωIPN+R (ωOPN+R) and ωIPN+T (ωOPN+T) denote resp. Petri nets with reset (R) and transfer (T) arcs with ω on input (output) arcs only.

<table>
<tr><th>Problem</th><th>ωPN</th><th>ωPN+T</th><th>ωPN+R</th></tr>
<tr><td>Reachability</td><td>Decidable and ExpSpace-hard (4)</td><td rowspan="2">Undecidable (7)</td><td rowspan="3">Undecidable (7)</td></tr>
<tr><td>Place-boundedness</td><td rowspan="3">ExpSpace-c (4)</td></tr>
<tr><td>Boundedness</td><td>Decidable (7)</td></tr>
<tr><td>Coverability</td><td colspan="2">Decidable and Ackerman-hard (7)</td></tr>
</table>

Problem	ωPN	ωOPN+T, ωOPN+R	ωIPN+T, ωIPN+R
Termination	ExpSpace-c (5)	Undecidable (7)	Decidable and Ackerman-hard (7)

this allows us to tame the technicality of the proof, and as a consequence, our proof when applied to *plain* Petri nets, provides a simplification of the original proof by Karp and Miller. Second, in Section 4, we show how to construct, from an ωPN, a plain Petri net that preserves its reachability set. This reduction allows us to prove that many bounds on the algorithmic complexity of (plain) PN problems apply to ωPN too. However, it does not preserve *termination*. Thus, we study, in Section 5, as a third contribution, an extension of the self-covering path technique due to Rackoff [22]. This technique allows us to provide a direct proof of ExpSpace upper bounds for several classical decision problems, and in particular, this allows us to prove ExpSpace completeness of the termination problem. Third, in Section 6, we show how to compute, given an ωPN $\mathcal{N}$ and a marking m which is coverable in $\mathcal{N}$, all *concretisations* of $\mathcal{N}$ in which m is still covered. By *concretisation*, we mean a plain PN in which all ω's have been replaced by natural numbers. Such a technique is useful for debugging purpose.

Finally, in Section 7, as an additional contribution, and to get a complete picture, we consider extensions of ωPN with *reset* and *transfer* arcs [8]. For those extensions, the decidability results for reset and transfer nets (without ω arcs) also apply to our extension with the notable exception of the termination problem that becomes, as we show here, undecidable. The summary of our results are given in Table 1.

Related works ωPN are well-structured transition systems [12]. The symbolic backward analysis [1] can be applied to ωPN while the finite tree unfolding is not applicable because of the infinite branching property of ωPN. For the same reason, ωPN are *not* well-structured nets [11]. In a recent work [2], Blondin *et al.* extend the *completion* technique for WSTS to infinitely branching WSTS, a general class that contain ωPN. Their technique are very general, while ours are specialised to ωPN.

In [4], Brazdil *et al.* extends the Rackoff technique to games for vector addition systems with states (VASS) extended by ω output arcs. This extension of the Rackoff technique is close to ours. However, their technical results have been formulated in the setting of games, that we do not consider here.

Several works (see for instance [6, 5]) rely on Petri nets to model *parametric systems* and perform *parametrised verification*. However, in all these works, the dynamic creation of threads uses the same pattern as in Fig. 1 (b), and does not preserve termination. ωPN allow us to model more faithfully the dynamic creation of an unbounded number of threads, and are thus better suited to model new programming paradigms (such as GCD [13]) that have been recently proposed to better support multi-core platforms.

2. ω-Petri nets

Let us define the syntax and semantics of our Petri net extension, called ω *Petri nets* (ωPN for short). Let ω be a symbol that denotes 'any positive integer value'. We extend the arithmetic and the $\leq$ ordering on $\mathbb{Z}$ to $\mathbb{Z} \cup \{\omega\}$ as follows: $\omega + \omega = \omega - \omega = \omega$; and for all $c \in \mathbb{Z}$: $c + \omega = \omega + c = \omega - c = \omega$; $c - \omega = c$; and $c \leq \omega$. The fact that $c - \omega = c$ might sound surprising but will be justified later when we introduce ωPN. An ω-*multiset* (or simply *multiset*) of elements from S is a function $m : S \mapsto \mathbb{N} \cup \{\omega\}$. We denote multisets m of $S = \{s_1, s_2, \ldots, s_n\}$ by extension using the syntax $\{m(s_1) \otimes s_1, m(s_2) \otimes s_2, \ldots, m(s_n) \otimes s_n\}$ (when $m(s) = 1$, we write s instead of $m(s) \otimes s$, and we omit elements $m(s) \otimes s$ when $m(s) = 0$). Given two multisets m_1 and m_2, and an integer value c we let $m_1 + m_2$ be the multiset s.t. $(m_1 + m_2)(p) = m_1(p) + m_2(p)$; $m_1 - m_2$ be the multiset s.t. $(m_1 - m_2)(p) = m_1(p) - m_2(p)$; and $c \cdot m_1$ be the multiset s.t. $(c \cdot m_1)(p) = c \times m_1(p)$ for all $p \in P$.

Syntax Syntactically, ωPN extend plain Petri nets [21, 23] by allowing (input and output) arcs to be labeled by ω. Intuitively, if a transition t has ω as output (resp. input) effect on place p, the firing of t non-deterministically creates (consumes) a positive number of tokens in p.

Definition 2.1. A *Petri net with* ω-*arcs* (ωPN) is a tuple $\mathcal{N} = \langle P, T \rangle$ where: P is a finite set of *places*; T a finite set of *transitions*. Each transition is a pair $t = (I, O)$, where: $I : P \to \mathbb{N} \cup \{\omega\}$ and $O : P \to \mathbb{N} \cup \{\omega\}$, give respectively the input (output) effect $I(p)$ ($O(p)$) of t on place p.

By abuse of notation, we denote by $I(t)$ (resp. $O(t)$) the functions s.t. $t = (I(t), O(t))$. When convenient, we sometimes regard $I(t)$ or $O(t)$ as ω-*multisets* of places. Whenever there is p s.t. $O(t)(p) = \omega$ (resp. $I(t)(p) = \omega$), we say that t is an ω-*output-transition* (ω-*input-transition*). A transition t is an ω-transition iff it is an ω-output-transition or an ω-input-transition. Otherwise, t is a *plain* transition. Note that a (plain) Petri net is an ωPN with plain transitions only. Moreover, when an ωPN contains no ω-output-transitions (resp. no ω-input transitions), we say that it is an ω-input-PN (ω-output-PN), or ωIPN (ωOPN) for short. For all transitions t, we denote by $\mathit{effect}(t)$ the function $O(t) - I(t)$. Note that $\mathit{effect}(t)(p)$ could be ω for some p (in particular when $O(t)(p) = I(t)(p) = \omega$). Intuitively, $\mathit{effect}(t)(p) = \omega$ models the fact that firing t can increase the marking of p by an arbitrary number of tokens. Finally, observe that $O(t)(p) = c \neq \omega$ and $I(t)(p) = \omega$ *implies* $\mathit{effect}(t)(p) = c - \omega = c$. This models the fact that firing t can at most increase the marking of p by c tokens. Thus, intuitively, the value $\mathit{effect}(t)(p)$ models the *maximal possible effect* of t on p. We extend the definition of effect to sequences of transitions $\sigma = t_1 t_2 \cdots t_n$ by letting $\mathit{effect}(\sigma) = \sum_{i=1}^{n} \mathit{effect}(t_i)$.

A *marking* is a function $P \mapsto \mathbb{N}$. An ω-*marking* is a function $P \mapsto \mathbb{N} \cup \{\omega\}$, i.e. an ω-multiset on P. Any marking is an ω-marking. For all transitions $t = (I, O)$, I and O are both ω-markings. We denote by $\mathbf{0}$ the marking s.t. $\mathbf{0}(p) = 0$ for all $p \in P$. For all ω-markings m, we let $\omega(m)$ be the set of places $\{p \mid m(p) = \omega\}$, and let $\mathrm{nb}\omega\,(m) = |\omega(m)|$. We define *the concretisation* of m as the set of all markings that coincide with m on all places $p \not\in \omega(m)$, and take an arbitrary value in any place from $\omega(m)$. Formally: $\gamma(m) = \{m' \mid \forall p \not\in \omega(m) : m'(p) = m(p)\}$. We further define a family of orderings on ω-markings as follows. For any $P' \subseteq P$, we let $m_1 \preceq_{P'} m_2$ iff (i) for all $p \in P'$:

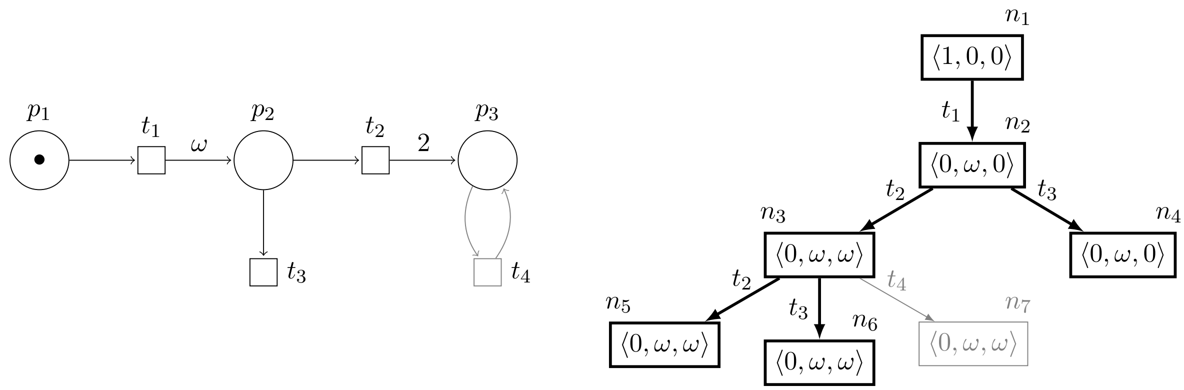

Figure 2. Left: an example ωPN $\mathcal{N}_1$. The ωPN $\mathcal{N}_1'$ is obtained by removing transition t_4 (gray). Right: The KM trees $\mathcal{T}_1$ (whole tree) and $\mathcal{T}_1'$ (bold nodes, i.e. w/o n_7) of resp. $\mathcal{N}_1$ and $\mathcal{N}_1'$.

$m_1(p) \leq m_2(p)$, and (ii) for all $p \in P \setminus P'$: $m_1(p) = m_2(p)$. We abbreviate $\preceq_P$ by $\preceq$ (where P is the set of places of the ωPN). It is well-known that $\preceq$ is a *well-quasi ordering* (wqo), that is, we can extract, from any infinite sequence $m_1, m_2, \ldots, m_i, \ldots$ of markings, an infinite subsequence $\overline{m}_1, \overline{m}_2, \ldots, \overline{m}_i, \ldots$ s.t. $\overline{m}_i \preceq \overline{m}_{i+1}$ for all $i \geq 1$. For all ω-markings m, we let $\downarrow(m)$ be the *downward-closure* of m, defined as $\downarrow(m) = \{m' \mid m' \text{ is a marking and } m' \preceq m\}$. We extend $\downarrow$ to sets of ω-markings: $\downarrow(S) = \cup_{m \in S} \downarrow(m)$. A set D of markings is *downward-closed* iff $\downarrow(D) = D$. It is well-known that (possibly infinite) downward-closed sets of markings can always be represented by a finite set of ω-markings, because the set of ω-markings forms an *adequate domain of limits* [14]: for all downward-closed sets D of markings, there exists a finite set M of ω-markings s.t. $\downarrow(M) = D$. We associate, to each ωPN, an *initial marking* m_0. From now on, we consider mostly initialised ωPN $\langle P, T, m_0\rangle$.

Example 2.2. An example of an ωPN (actually an ωOPN) $\mathcal{N}_1 = \langle P, T, m_0\rangle$ is shown in Fig. 2 (left). In this example, $P = \{p_1, p_2, p_3\}$, $T = \{t_1, t_2, t_3, t_4\}$, $m_0(p_1) = 1$ and $m_0(p_2) = m_0(p_3) = 0$. t_1 is the only ω-transition, with $O(t_1)(p_2) = \omega$. This ωPN will serve as a running example in this section.

Semantics Let m be an *ω-marking*. A transition $t = (I, O)$ is *firable from* m iff: $m(p) \succeq I(p)$ for all p s.t. $I(p) \neq \omega$. We consider two kinds of possible effects for t. The first is the *concrete semantics* and applies only when m is a *marking*. In this case, firing t yields a new marking m' s.t. for all $p \in P$: $m'(p) = m(p) - i + o$ where: (i) $i = I(t)(p)$ if $I(t)(p) \neq \omega$; (ii) $i \in \{0, \ldots, m(p)\}$ if $I(t)(p) = \omega$; (iii) $o = O(t)(p)$ if $O(t)(p) \neq \omega$; (iv) and $o \geq 0$ if $O(t)(p) = \omega$. This is denoted by $m \xrightarrow{t} m'$. Thus, intuitively, $I(t)(p) = \omega$ (resp. $O(t)(p) = \omega$) means that t consumes (produces) an arbitrary number of tokens in p when fired. Note that, in the concrete semantics, ω-transitions are *non-deterministic*: when t is an ω-transition with an ω-output arc that is firable in m, there are *infinitely many* m' s.t. $m \xrightarrow{t} m'$. The latter semantics is the *ω-semantics*. In this case, firing $t = (I, O)$ yields the (unique) ω-marking $m' = m - I + O$ (denoted $m \xrightarrow{t}_\omega m'$). Note that $m \xrightarrow{t} m'$ iff $m \xrightarrow{t}_\omega m'$ when m and m' are markings.

We extend the $\rightarrow$ and $\rightarrow_\omega$ relations to finite or infinite sequences of transitions in the usual way. Also we write $m \xrightarrow{\sigma}$ iff σ is *firable* from m. More precisely, for a finite sequence of transitions $\sigma = t_1 \cdots t_n$, we write $m \xrightarrow{\sigma}$ iff there are $m_1, \ldots, m_n$ s.t. for all $1 \leq i \leq n$: $m_{i-1} \xrightarrow{t_i} m_i$. For an infinite sequence of transitions $\sigma = t_1 \cdots t_j \cdots$, we write $m_0 \xrightarrow{\sigma}$ iff there are $m_1, \ldots, m_j, \ldots$ s.t. for all $i \geq 1$: $m_{i-1} \xrightarrow{t_i} m_i$.

Given an ωPN $\mathcal{N} = \langle P, T, m_0 \rangle$, an *execution* of $\mathcal{N}$ is either a finite sequence of the form $m_0, t_1, m_1, t_2, \ldots, t_n, m_n$ s.t. $m_0 \xrightarrow{t_1} m_1 \xrightarrow{t_2} \cdots \xrightarrow{t_n} m_n$, or an infinite sequence of the form $m_0, t_1, m_1, t_2, \ldots, t_j, m_j, \ldots$ s.t. for all $j \geq 1$: $m_{j-1} \xrightarrow{t_j} m_j$. We denote by $\mathsf{Reach}(\mathcal{N})$ the set of markings $\{m \mid \exists \sigma \text{ s.t. } m_0 \xrightarrow{\sigma} m\}$ that are reachable from m_0 in $\mathcal{N}$. Finally, a *finite set of ω-markings* $\mathcal{CS}$ is a *coverability set* of $\mathcal{N}$ (with initial marking m_0) iff $\downarrow(\mathcal{CS}) = \downarrow(\mathsf{Reach}(\mathcal{N}))$. That is, any coverability set $\mathcal{CS}$ is a *finite representation of the downward-closure of $\mathcal{N}$'s reachable markings*.

Example 2.3. The sequence $t_1 t_2^K$ is firable for all $K \geq 0$ in $\mathcal{N}_1$ (Fig. 2): for each $K \geq 0$, one possible execution corresponding to $t_1 t_2^K$ is given by $\langle 1,0,0 \rangle \xrightarrow{t_1} \langle 0,3K,0 \rangle \xrightarrow{t_2} \langle 0,3K-1,2 \rangle \xrightarrow{t_2} \langle 0,3K-2,4 \rangle \xrightarrow{t_2} \cdots \xrightarrow{t_2} \langle 0,2K,2K \rangle$. There are other possible executions corresponding to the same sequence of transitions, because the number of tokens created by t_1 in p_2 is chosen non-deterministically. Also, $t_1 t_2 t_4^\omega$ is an infinite firable sequence of transitions. The set of reachable markings in $\mathcal{N}_1$ is $\mathsf{Reach}(\mathcal{N}) = \{\langle 1,0,0 \rangle\} \cup \{\langle 0, i, 2 \times j \rangle \mid i, j \in \mathbb{N}\}$. The set of ω markings $\mathcal{CS} = \{\langle 1,0,0 \rangle, \langle 0,\omega,\omega \rangle\}$ is a coverability set of $\mathcal{N}$. Note that $\downarrow(\mathcal{CS}) \supsetneq \mathsf{Reach}(\mathcal{N})$: for instance, $\langle 0,1,1 \rangle \in \downarrow(\mathcal{CS})$, but $\langle 0,1,1 \rangle$ is not reachable.

Let us now observe two properties of the semantics of ωPN, that will be useful for the proofs of Section 3. First, when firing a sequence of transitions σ that have non ω-labeled arcs on to and from some place p, the effect of σ on p is as in a plain PN. Second, the set of markings that are reachable by a given sequence of transitions σ is upward-closed[2] w.r.t. $\preceq_{P'}$, where P' is the set of places where the effect of σ is ω.

Lemma 2.4. Let m and m' be two markings and let $\sigma = t_1 \cdots t_n$ be a sequence of transitions of an ωPN s.t. $m \xrightarrow{\sigma} m'$. Let p be a place s.t. for all $1 \leq i \leq n$: $O(t_i)(p) \neq \omega \neq I(t_i)(p)$. Then, $m'(p) = m(p) + \mathit{effect}(\sigma)(p)$.

Lemma 2.5. Let m_1, m_2 and m_3 be three markings, and let σ be a sequence of transitions s.t. (i) $m_1 \xrightarrow{\sigma} m_2$; and (ii) $m_3 \succeq_{P'} m_2$ with $P' = \{p \mid \mathit{effect}(\sigma)(p) = \omega\}$. Then, $m_1 \xrightarrow{\sigma} m_3$.

Problems We consider the following problems, where $\mathcal{N} = (P, T, m_0)$ is an ωPN. (1) The *reachability problem* asks, given a marking m, whether $m \in \mathsf{Reach}(N)$. (2) The *place boundedness problem* asks, given a place p of $\mathcal{N}$, whether there exists $K \in \mathbb{N}$ s.t. for all $m \in \mathsf{Reach}(\mathcal{N})$: $m(p) \leq K$. If the answer is positive, we say that p is *bounded* (from m_0). (3) The *boundedness problem* asks whether all places of $\mathcal{N}$ are bounded (from m_0). (4) The *covering problem* asks, given a marking m of $\mathcal{N}$, if there is $m' \in \mathsf{Reach}(\mathcal{N})$ s.t. $m' \succeq m$. (5) The *termination problem* asks whether all executions of $\mathcal{N}$ are finite.

A *coverability set* of the ωPN is sufficient to solve *boundedness*, *place boundedness* and *covering*, as in the case of Petri nets. If $\mathcal{CS}$ is a coverability set of $\mathcal{N}$, then: (i) p is bounded iff $m(p) \neq \omega$ for all $m \in \mathcal{CS}$; (ii) $\mathcal{N}$ is bounded iff $m(p) \neq \omega$ for all p and for all $m \in \mathcal{CS}$; and (iii), $\mathcal{N}$ can cover m iff there exists $m' \in \mathcal{CS}$ s.t. $m \preceq m'$. As in the plain Petri nets case, a sufficient and necessary condition of non-termination is the existence of a *self covering execution*. A *self covering execution* of an ωPN $\mathcal{N} = \langle P, T, m_0 \rangle$ is a *finite* execution of the form $m_0 \xrightarrow{t_1} m_1 \cdots \xrightarrow{t_k} m_k \xrightarrow{t_{k+1}} \cdots \xrightarrow{t_n} m_n$ with $m_n \succeq m_k$:

[2] A set $U \subseteq S$ is upward-closed wrt to a partial order $\leq$ iff for all $u \in U$ and $s \in S$: $u \leq s$ implies that $s \in U$.

Lemma 2.6. An ωPN terminates iff it admits no self-covering execution.

Proof:
Assume $\mathcal{N} = \langle P, T, m_0 \rangle$ admits an infinite execution $m_0 \xrightarrow{t_1} m_1 \rightarrow t_2 \cdots \xrightarrow{t_j} m_j \xrightarrow{t_{j+1}} \cdots$. Since $\preceq$ is a well-quasi ordering on the markings, there are two positions α and β in the execution s.t. $\alpha \leq \beta$ and $m_\alpha \preceq m_\beta$. Hence, $m_0 \xrightarrow{t_1} m_1 \xrightarrow{t_2} \cdots \xrightarrow{t_\beta} m_\beta$ is a self-covering execution.

For the reverse implication, assume $\mathcal{N} = \langle P, T, m_0 \rangle$ admits a self-covering execution $m_0 \xrightarrow{t_1} m_1 \rightarrow t_2 \cdots \xrightarrow{t_n} m_n$ and assume $0 \leq k < n$ is a position s.t. $m_k \preceq m_n$. Then, by monotonicity, it is possible to fire infinitely often the $t_{k+1} \cdots t_n$ sequence from m_k. More precisely, one can check that the following is an infinite execution of $\mathcal{N}$: $m_0 \xrightarrow{t_1} m_1 \cdots \xrightarrow{t_k} m_k \xrightarrow{t_{k+1}} m^0_{k+1} \cdots \xrightarrow{t_n} m^0_n \xrightarrow{t_{k+1}} m^1_{k+1} \cdots \xrightarrow{t_n} m^1_n \xrightarrow{t_{k+1}} m^2_{k+1} \cdots \xrightarrow{t_n} m^2_n \cdots \xrightarrow{t_{k+1}} m^j_{k+1} \cdots \xrightarrow{t_n} m^j_n \cdots$, where for all $1 \leq i \leq n-k$: $m^0_{k+i} = m_{k+i}$, for all $j \geq 1$, $m^j_{k+1} = m^{j-1}_n + (m_{k+1} - m_k)$ and for all $2 \leq i \leq n-k$: $m^j_i = m^j_{i-1} + (m_{k+i} - m_{k+i-1})$. $\square$

Example 2.7. Consider again the ωPN $\mathcal{N}_1$ in Fig. 2. Recall from Example 2.3 that, for all $K \geq 0$, $t_1 t_2^K$ is firable and allows to *reach* $\langle 0, 2K, 2K \rangle$. All these markings are thus *reachable*. These sequences of transitions also show that p_2 and p_3 are *unbounded* (hence, $\mathcal{N}_1$ is unbounded too), while p_1 is *bounded*. Marking $\langle 0, 1, 1 \rangle$ is *not reachable* but *coverable*, while $\langle 2, 0, 0 \rangle$ is neither reachable nor coverable. Finally, $\mathcal{N}_1$ does not terminate (because $t_1 t_2 t_4^\omega$ is firable), while $\mathcal{N}'_1$ does. In particular, *in* $\mathcal{N}'_1$, t_3 can fire only a *finite* number of times, because t_1 will always create a finite (albeit unbounded) number of tokens in p_2. This an important difference between ωPN and plain PN: no unbounded PNs terminates, while there are unbounded ωPN that terminate, e.g. $\mathcal{N}'_1$.

3. A Karp and Miller procedure for ωPN

In this section, we present an extension of the classical Karp & Miller procedure [18], adapted to ωPN. We show that the finite tree built by this algorithm (coined the KM tree), allows us, as in the case of PNs, to decide *boundedness*, *place boundedness*, *coverability* and *termination* on ωPN.

Before describing the algorithm, we discuss intuitively the KM trees of the ωPN $\mathcal{N}_1$ and $\mathcal{N}'_1$ given in Fig. 2. Their respective KM trees (for the initial marking $m_0 = \langle 1, 0, 0 \rangle$) are $\mathcal{T}_1$ and $\mathcal{T}'_1$, respectively the tree in Fig. 2 and its *subtree made of the bold nodes* (i.e., excluding n_7). As can be observed, the nodes and edges of a KM tree are labeled by ω-markings and transitions respectively. The relationship between a KM tree and the executions of the corresponding ωPN can be formalised using the notion of *stuttering path*. Intuitively, a stuttering path is a sequence of nodes $n_1, n_2, \ldots, n_k$ s.t. for all $i \geq 2$: either n_i is a son of n_{i-1}, or n_i is an *ancestor* of n_{i-1} *that has the same label* as n_{i-1}. For instance, $\pi = n_1, n_2, n_4, n_2, n_3, n_6, n_3, n_5, n_3, n_5$ is a stuttering path in $\mathcal{T}'_1$. Then, we claim (i) that *every execution of the* ω*PN is simulated by a stuttering path* in its KM tree, and that (ii) *every stuttering path in the* KM *tree corresponds to a family of executions of the* ω*PN*, where an arbitrary number of tokens can be produced in the places marked by ω in the KM tree. For instance, the execution $m_0, t_1, \langle 0, 42, 0 \rangle, t_3, \langle 0, 41, 0 \rangle, t_2, \langle 0, 40, 2 \rangle, t_3, \langle 0, 39, 2 \rangle, t_2, \langle 0, 38, 4 \rangle, t_2, \langle 0, 37, 6 \rangle$, of $\mathcal{N}'_1$ is witnessed in $\mathcal{T}'_1$ by the stuttering path π given above – observe that the sequence of edge labels in π's equals the sequence of transitions of the execution, and that all markings along the execution are *covered* by the labels of the corresponding nodes in π: $m_0 \in \gamma(n_1)$, $\langle 0, 42, 0 \rangle \in \gamma(n_2)$, and so forth. On the other hand, the stuttering path n_1, n_2, n_3 of $\mathcal{N}_1$ summarises all the (infinitely many) possible executions obtained by firing a sequence of the form $t_1 t_2^n$. Indeed, for all $k \geq 1$, $\ell \geq 0$: $m_0, t_1, \langle 0, k+\ell, 0\rangle, t_2, \langle 0, k+\ell-1, 2\rangle, t_2, \ldots, t_2, \langle 0, k, 2\times\ell\rangle$ is an execution of $\mathcal{N}_1$, so, an arbitrary number of tokens can be obtained in both p_2 and p_3 by firing sequences of the form $t_1 t_2^n$. Finally, observe that a *self-covering execution* of $\mathcal{N}_1$, such as $m_0, t_1, \langle 0,1,0\rangle, t_2, \langle 0,0,2\rangle, t_4, \langle 0,0,2\rangle$ can be detected in $\mathcal{T}_1$, by considering the path n_1, n_2, n_3, n_7, and noting that the label of (n_3, n_7) is t_4 with $\mathit{effect}(t_4) \succeq \mathbf{0}$.

3.1. The `Build-KM` algorithm

Let us now show how to build algorithmically the KM of an ωPN. Recall that, *in the case of plain PNs*, the Karp & Miller tree [18] can be regarded as a *finite over-approximation of the (potentially infinite) reachability tree of the PN*. Thus, the Karp & Miller algorithm works by unfolding the transition relation of the PN, and adds two ingredients to guarantee that the tree is finite. First, a node n that has an ancestor n' *with the same label* is *not developed* (it has no children). Second, when a node n with label m has an ancestor n' with label $m' \prec m$, an *acceleration function* is applied to produce a marking m_ω s.t. $m_\omega(p) = \omega$ if $m(p) > m'(p)$ and $m_\omega(p) = m(p)$ otherwise. This acceleration is *sound* wrt to coverability since the sequence of transition that has produced the branch (n, n') can be iterated an arbitrary number of times, thus producing arbitrary large numbers of tokens in the places marked by ω in m_ω. Remark that these two constructions are not sufficient to ensure termination of the algorithm in the case of ωPN, as ωPN are *not finitely branching* (firing an ω-output-transition can produce infinitely many different successors). To cope with this difficulty, our solution unfolds the ω-*semantics* $\rightarrow_\omega$ instead of the concrete semantics $\rightarrow$. This has an important consequence: whereas the presence of a node labeled by m with $m(p) = \omega$ in the KM tree of a PN $\mathcal{N}$ *implies* that $\mathcal{N}$ *does not terminate*, this is *not true anymore* in the case of ωPN. For instance, all nodes but n_1 in $\mathcal{T}_1'$ (Fig. 2) are marked by ω, yet the corresponding ωPN $\mathcal{N}_1'$ (Fig. 2) *does terminate*.

Our version of the Karp & Miller tree adapted to ωPN is given in Fig. 3. It builds a tree $\mathcal{T} = \langle N, E, \lambda, \mu, n_0\rangle$ where: N is a set of nodes; $E \subseteq N \times N$ is a set of edges; $\lambda : N \mapsto (\mathbb{N} \cup \{\omega\})^P$ is a function that labels nodes by ω-markings[3]; $\mu : E \mapsto T$ is a labeling function that labels arcs by transitions; and $n_0 \in N$ is the root of the tree. For each edge e, we let $\mathit{effect}(e) = \mathit{effect}(\mu(e))$. Let E^+ and E^* be respectively the transitive and the transitive reflexive closure of E. A *stuttering path* is a finite sequence $n_0, n_1, \ldots, n_\ell$ s.t. for all $1 \leq i \leq \ell$: *either* $(n_{i-1}, n_i) \in E$ *or* $(n_i, n_{i-1}) \in E^+$ and $\lambda(n_i) = \lambda(n_{i-1})$. A stuttering path $n_0, n_1, \ldots, n_\ell$ is a *(plain) path* iff $(n_{i-1}, n_i) \in E$ for all $1 \leq i \leq \ell$. Given two nodes n and n' s.t. $(n, n') \in E^*$, we denote by $n \rightsquigarrow n'$ the (unique path) from n to n'. Given a stuttering path $\pi = n_0, n_1, \ldots, n_\ell$, we denote by $\mu(\pi)$ the sequence $\mu(n_0, n_1)\mu(n_1, n_2)\cdots\mu(n_{\ell-1}, n_\ell)$ assuming $\mu(n_i, n_{i+1}) = \varepsilon$ when $(n_i, n_{i+1}) \notin E$; and by $\mathit{effect}(\pi) = \sum_{i=1}^{\ell} \mathit{effect}(n_{i-1}, n_i)$, letting $\mathit{effect}(n_{i-1}, n_i) = \mathbf{0}$ when $(n_i, n_{i+1}) \notin E$.

`Build-KM` follows the intuition given above. At all times, it maintains a frontier `U` of tree nodes that are candidate for development (initially, `U` $= \{n_0\}$, with $\lambda(n_0) = m_0$). Then, `Build-KM` iteratively picks up a node n from `U` (see line 4), and develops it (line 6 onwards) if n has no ancestor n' with the same label (line 5). *Developing* a node n amounts to computing all the marking m_w s.t. $\lambda(n) \rightarrow_\omega m_w$ (line 17), performing accelerations (line 19) if need be, and inserting the resulting children in the tree. Note that `Build-KM` is *recursive* (see line 9): every time a marking m with an extra ω is created, it

[3] We extend λ to set of nodes S in the usual way: $\lambda(S) = \{\lambda(n) \mid n \in S\}$.

Input: an ωOPN $\mathcal{N} = \langle P, T\rangle$ and an ω-marking m_0. **Output**: the KM of $\mathcal{N}$, starting from m_0.
Build-KM $(\mathcal{N}, m_0)$:

```
T := ⟨N, E, λ, μ, n_0⟩ where N = {n_0} with λ(n_0) = m_0
U := {n_0}
while U ≠ ∅:
    select and remove n from U
    if ∄n̄ st (n̄, n) ∈ E^+ and λ(n) = λ(n̄):
        forall t in T s.t. ∀p ∈ P: I(t)(p) ≠ ω implies λ(n)(p) ≥ I(t)(p):
            m' := Post(N, λ(n), t)
            if nbω(m') > nbω(λ(n)):
                T' := Build-KM(N, m')
                add all edge and nodes of T' to T
                let n' be the root of T'
            else
                n' := new node with λ(n') = m'
                U := U ∪ {n'}
            E := E ∪ (n, n') s.t. μ(n, n') = t.
return T

Post(N, n, t):
m_w := λ(n) − I(t) + O(t)
if ∃n̄ : (n̄, n) ∈ E^+ ∧ λ(n̄) ≺ λ(n)):
    for all p s.t. effect(n̄ ⇝ n · t)(p) > 0: m_w(p) := ω
return m_w
```

Figure 3. The algorithm to build the KM of an ωPN.

performs a recursive call to Build-KM$(\mathcal{N}, m)$, using m as initial marking[4].

In the rest of the section we prove correctness of this algorithm. We first establish termination, then soundness and finally completeness. To this end, we rely on the following notions. Symmetrically to *self-covering executions* we define the notion of *self-covering (stuttering) path* in a tree: a (stuttering) path π is *self-covering* iff $\pi = \pi_1\pi_2$ with $\mathit{effect}(\pi_2) \geq \mathbf{0}$. A self-covering stuttering path $\pi = \pi_1\pi_2$ is *ω-maximal* iff for all nodes n, n' along π_2: $\mathrm{nb}\omega\,(n) = \mathrm{nb}\omega\,(n')$.

3.2. Correctness of the Build-KM algorithm

Termination Let us show that Build-KM always terminates. First observe that the depth of recursive calls is at most by $|P| + 1$, as the number of places marked by ω along a branch does not decrease, and since we perform a recursive call only when a place gets marked by ω and was not marked before. Moreover, the branching degree of the tree is bounded by the number $|T|$ of transitions. Thus, by König's lemma, an infinite tree would contain an infinite branch. We rule out this possibility by a classical wqo argument: if there were an infinite branch in the tree computed by Build-KM$(\mathcal{N}, m_0)$, then there would be two nodes n_1 along the branch n_2 (where n_1 is an ancestor of n_2) s.t. $\lambda(n_1) \preceq \lambda(n_2)$ and $\mathit{effect}(n_1 \rightsquigarrow n_2) \succeq \mathbf{0}$. Since the depth of recursive calls is bounded, we can assume, w.l.o.g., that n_1

[4]Although this differs from classical presentations of the Karp & Miller technique, we have retained it because it simplifies the proofs of correctness.

and n_2 have been built during the same recursive call, hence $\lambda(n_1) \prec \lambda(n_2)$ is not possible, because this would trigger an acceleration, create an extra ω and start a new recursive call. Thus, $\lambda(n_1) = \lambda(n_2)$, but in this case the algorithm stops developing the branch (line 5).

Proposition 3.1. For all ωPN $\mathcal{N}$ and marking m_0, $\texttt{Build-KM}(\mathcal{N}, m_0)$ terminates.

Proof:
The proof is by contradiction. Assume $\texttt{Build-KM}(\mathcal{N}, m_0)$ does not terminate. First observe that the recursion depth is always bounded: since a recursive call is performed only when a new ω has been created, the recursion depth is, at any time, at most equal to $|P| + 1$, where P is the set of places of $\mathcal{N}$

Thus, if $\texttt{Build-KM}(\mathcal{N}, m_0)$ does not terminate, it is necessarily because the main **while** loop does not terminate (the other loop of the algorithm is the **forall** starting in line 6, which always execute at most $|T|$ iterations, where T is the set of transitions of $\mathcal{N}$). In this loop, one node is removed from `U` at each iteration. Since the algorithm builds a tree, a node that has been removed from `U` will never be inserted again in `U`. Hence, the tree $\mathcal{T}$ built by $\texttt{Build-KM}(\mathcal{N}, m_0)$ is infinite.

By König's lemma, and since $\mathcal{T}$ is finitely branching, it contains an infinite path π. Since the recursion depth is bounded, π can be split into a finite prefix π_1 and an infinite suffix π_2 s.t. all the nodes in π_2 have been built during the same recursive call.

Let us assume $\pi_2 = n_0, n_1, \ldots, n_m, \ldots$ Since $\preceq$ is a well-quasi-ordering on ω-markings, there are k and ℓ s.t. $0 \leq k < \ell$ and $\lambda(n_k) \preceq \lambda(n_\ell)$. Clearly, $\lambda(n_k) = \lambda(n_\ell)$ is not possible because of the test of line 5 that prevents the development of n_ℓ in this case. Thus, $\lambda(n_k) \prec \lambda(n_\ell)$. This means that, for all $p \in P$: $\lambda(n_k)(p) \leq \lambda(n_\ell)(p)$, and that there exists p s.t. $\lambda(n_k)(p) < \lambda(n_\ell)(p)$. Let $p^<$ be such a place. By definition of the `Post` function, and of the acceleration (line 19), the only possibility is that $\lambda(n_\ell)(p^<) = \omega \neq \lambda(n_k)(p^<)$. However, in this case, when $\lambda(n_\ell)$ is returned by `Post`, a new recursive call is triggered, which contradicts the hypothesis that n_ℓ and n_k have been built during the same recursive call, which contradicts the assumption. □

Then, following the intuition that we have sketched at the beginning of the section, we show that KM is *sound* (Lemma 3.2) and *complete* (Lemma 3.6). We first establish these results assuming that the ωPN $\mathcal{N}$ given as parameter is an ωOPN, then prove that the results extend to the general case of ωPN.

Soundness To establish *soundness* of our algorithm, we show that, for every path $n_0, \ldots, n_k$ in the tree returned by $\texttt{Build-KM}(\mathcal{N}, m_0)$, and for every target marking $m \in \gamma(\lambda(n_k))$, we can find an execution of $\mathcal{N}$ reaching a marking $m' \in \gamma(n_k)$ that covers m. This implies that, if $\lambda(n_k)(p) = \omega$ for some p, then, we can find a family of executions that reach a marking in $\gamma(n_k)$ with an arbitrary number of tokens in p. For instance, consider the path n_1, n_2, n_3 in $\mathcal{T}_1'$ (Fig. 2), and let $m = \langle 0, 2, 4\rangle$. Then, a corresponding execution is $\langle 1, 0, 0\rangle \xrightarrow{t_1} \langle 0, 4, 0\rangle \xrightarrow{t_2} \langle 0, 3, 2\rangle \xrightarrow{t_2} \langle 0, 2, 4\rangle$. Remark that the execution is not necessarily the sequence of transitions labeling the path in the tree: in this case, we need to iterate t_2 to transfer tokens from p_2 to p_3, which is summarised in one edge (n_2, n_3) in $\mathcal{T}_1$, by the acceleration.

Thus, the soundness property is given by the following lemma:

Lemma 3.2. Let $\mathcal{N}$ be an ωOPN, let m_0 be an ω-marking and let $\mathcal{T}$ be the tree returned by $\texttt{Build-KM}(\mathcal{N}, m_0)$. Let $\pi = n_0, \ldots, n_k$ be a stuttering path in $\mathcal{T}$, and let m be a marking in $\gamma(\lambda(n_k))$. Then, there exists an execution $\rho_\pi = m_0 \xrightarrow{t_1} m_1 \cdots \xrightarrow{t_\ell} m_\ell$ of $\mathcal{N}$ s.t. $m_\ell \in \gamma(\lambda(n_k))$, $m_\ell \succeq m$

and $m_0 \in \gamma(\lambda(n_0))$. Moreover, when for all $0 \leq i \leq j \leq k$: $\mathrm{nb}\omega\,(n_i) = \mathrm{nb}\omega\,(n_j)$, we have: $t_1 \cdots t_\ell = \mu(\pi)$.

To prove Lemma 3.2, we need auxiliary results and definitions. First, we state the *place monotonicity* property of ωPN. Let m_1 and m_2 be two markings, and let $P' \subseteq P$ be a set of places s.t. $m_2 \succeq_{P'} m_1$. Let σ be a sequence of transitions and let m_3 be a marking[5] s.t. $m_1 \xrightarrow{\sigma} m_3$. Then, there is m_4 s.t. $m_2 \xrightarrow{\sigma} m_4$ and $m_4 \succeq_{P'} m_3$. Also, when no ω's are introduced in the labels of the nodes, the sequence of labels along a branch coincides with the effect of the transitions labelling this branch. Formally:

Lemma 3.3. Let $\mathcal{N}$ be an ωOPN, let m_0 be an ω-marking and let $\mathcal{T}$ be the tree returned by $\texttt{Build-KM}(\mathcal{N}, m_0)$. Let n_1, n_2 be two nodes of $\mathcal{T}$ s.t. $(n_1, n_2) \in E^+$. Then, for all p s.t. $\lambda(n_1)(p) \neq \omega$ and $\lambda(n_2)(p) \neq \omega$, we have: $\lambda(n_2)(p) = \lambda(n_1)(p) + \mathit{effect}(\sigma)(p)$.

The next technical definitions allows us to characterise when a sequence of transition is firable from a given marking. Let $\sigma = t_1 \cdots t_n$ be a sequence of transitions of an ωOPN, s.t. for all $1 \leq i \leq n-1$, for all $p \in P$: $O(t_i)(p) \neq \omega$. Let m be a marking and let p be a place. Then, we let $\mathsf{AllowsFiring}$ be the predicate s.t. $\mathsf{AllowsFiring}(\sigma, m, p)$ is true iff: $\forall 1 \leq i \leq n : m(p) + \mathit{effect}(t_1 \cdots t_{i-1})(p) \geq I(t_i)(p)$. Note that σ is firable from m iff for all $p \in P$: $\mathsf{AllowsFiring}(\sigma, m, p)$. We extend the definition of $\mathsf{AllowsFiring}$ to sequences of transitions containing one ω-output-transition. Let $\sigma = t_1 \cdots t_n$ be a sequence of transitions, let p be a place, and let $1 \leq j \leq n$ be the least position s.t. $O(t_j)(p) = \omega$. Then $\mathsf{AllowsFiring}(\sigma, m, p)$ holds iff $\mathsf{AllowsFiring}(t_1 \cdots t_j, m, p)$ holds. Again, σ is firable from m iff for all $p \in P$: $\mathsf{AllowsFiring}(\sigma, m, p)$. Indeed, $\mathsf{AllowsFiring}(t_1 \cdots t_j, m, p)$ ensures that, when firing σ from m, p will never be negative along $t_1 \cdots t_j$. Moreover, t_j can create an arbitrary large number of tokens in p, since $O(t_j)(p) = \omega$, which allows to ensure that p will never be negative along $t_{j+1} \cdots t_n$. Given this definition of $\mathsf{AllowsFiring}$ it is easy to observe that: (i) $m(p) \geq I(\sigma)(p)$ implies $\mathsf{AllowsFiring}(\sigma, m, p)$; and (ii) $\mathsf{AllowsFiring}(\sigma, m, p)$ and $\mathit{effect}(\sigma)(p) \geq 0$ implies $\mathsf{AllowsFiring}(\sigma^K, m, p)$ for all $K \geq 1$.

Lemma 3.4. Let $\mathcal{N}$ be an ωOPN, let m_0 be an ω-marking, and let $\mathcal{T}$ be the tree returned by $\texttt{Build-KM}(\mathcal{N}, m_0)$, let $e = (n_1, n_2)$ be an edge of $\mathcal{T}$ and let m be a marking in $\gamma(\lambda(n_2))$. Then, there are $m_1 \in \gamma(\lambda(n_1))$, $m_2 \in \gamma(\lambda(n_2))$ and a sequence of transitions σ_π of $\mathcal{N}$ s.t. $m_1 \xrightarrow{\sigma_\pi} m_2$ and $m_2 \succeq m$. Moreover, when $\mathrm{nb}\omega\,(\lambda(n_1)) = \mathrm{nb}\omega\,(\lambda(n_2))$, $\sigma_\pi = \mu(e)$ is such a sequence of transitions.

Proof:

Edges are created by $\texttt{Build-KM}$ in line 15 only. Thus, by the test of the `forall` loop (line 6), and since we are considering an ωOPN:

$$\lambda(n_1) \quad \geq \quad I(\mu(e)) \tag{1}$$

Moreover, when creating an edge (n, n') (line 15), n' is either a fresh node s.t. $\lambda(n')$ is the ω-marking returned by $\texttt{Post}(\mathcal{N}, \lambda(n), t)$, or n' is the root of the subtree $\mathcal{T}'$ returned by the recursive call $\texttt{Build-KM}(\mathcal{N}, m')$, with $\mu(n, n') = t$ in both cases. However, in the latter case, the root of $\mathcal{T}'$ is m', i.e., the marking returned by $\texttt{Post}(\mathcal{N}, \lambda(n), t)$ too. Since this holds for all edges, we conclude that $\lambda(n_2)$ is the ω-marking m' returned by $\texttt{Post}(\mathcal{N}, \lambda(n_1), \mu(e))$. Considering the definition of the $\texttt{Post}$ function, we see that m' is either $\lambda(n_1) - I(t) + O(t)$ (when the condition of the `if` in line 18 is not

[5] Note that due to the ω's, the effect of σ is now non-deterministic, and there can be several such m_3.

satisfied), or the result m_ω of an acceleration (when the condition of the `if` in line 18 is satisfied). We consider these two cases separately.

CASE A: **the condition of the `if` in line 18 has not been satisfied (i.e., no acceleration has occurred).** Then, $\lambda(n_2)$ is the marking m' computed in line 17:

$$\lambda(n_2) = \lambda(n_1) - I(\mu(e)) + O(\mu(e)) \tag{2}$$

We let m_1 be the marking s.t. for all places $p \in P$: (i) $m_1(p) = \lambda(n_1)(p)$ if $\lambda(n_1)(p) \neq \omega$; and (ii) $m_1(p) = I(\mu(e))(p) + m(p)$ otherwise. And we let m_2 be the marking s.t., for all places $p \in P$: (i) $m_2(p) = m_1(p) + O(\mu(e))(p) - I(\mu(e))(p)$ if $O(\mu(e))(p) \neq \omega$; and (ii) $m_2(p) = m_1(p) - I(\mu(e))(p) + m(p)$ otherwise. Finally, we let: $\sigma_\pi = \mu(e)$.

Let us show that m_1, m_2 and $\sigma_\pi = \mu(e)$ satisfy the lemma. First, we observe that $m_1 \in \gamma(\lambda(n_1))$, by definition. Then, we further observe that there are only four possibilities regarding the possible values of $\lambda(n_1)(p)$, $\lambda(n_2)(p)$ and $O(\mu(e))(p)$, as shown in the following table. Indeed, n_2 is a successor of n_1 in the tree, so $\omega(n_2) \supseteq \omega(n_1)$. Moreover, $\lambda(n_2)(p) = \omega \neq \lambda(n_1)(p)$ holds for some p iff $O(\mu(e))(p) = \omega$, as we have assumed that the condition of the `if` in line 18 has not been satisfied. Those four cases are summarised in the table below, with the values of $m_1(p)$ and $m_2(p)$, obtained by definition:

Case	$\lambda(n_1)(p)$	$\lambda(n_2)(p)$	$O(\mu(e))(p)$	$m_1(p)$	$m_2(p)$
1	$=\omega$	$=\omega$	$=\omega$	$I(\mu(e))(p) + m(p)$	$2 \times m(p)$
2	$=\omega$	$=\omega$	$\neq\omega$	$I(\mu(e))(p) + m(p)$	$m(p) + O(\mu(e))(p)$
3	$\neq\omega$	$=\omega$	$=\omega$	$\lambda(n_1)(p)$	$\lambda(n_1)(p) - I(\mu(e))(p) + m(p)$
4	$\neq\omega$	$\neq\omega$	$\neq\omega$	$\lambda(n_1)(p)$	$\lambda(n_1)(p) + O(\mu(e))(p) - I(\mu(e))(p)$

To prove that $m_2 \in \gamma(\lambda(n_2))$, we must show that $m_2(p) = \lambda(n_2)(p)$ for all p s.t. $\lambda(n_2)(p) \neq \omega$, which corresponds only to case 4. In this case, by the table above, we have $m_2(p) = \lambda(n_1)(p) + O(\mu(e))(p) - I(\mu(e))(p)$, which is equal to $\lambda(n_2)(p)$ by (2)

Then, it remains to show that $m_1 \xrightarrow{\mu(e)} m_2$. First, we show that, $\mu(e)$ is firable from m_1, i.e. that for all $p \in P$: $m_1(p) \geq I(\mu(e))(p)$. In case 1 and 2, we have $m_1(p) = I(\mu(e))(p) + m(p) \geq I(\mu(e))(p)$. In cases 3 and 4, we have $m_1(p) = \lambda(n_1)(p)$, with $\lambda(n_1)(p) \geq I(\mu(e))(p)$ by (1). Thus, $\mu(e)$ is firable from m_1. Then, we must show that m_2 can be obtained as a successor of m_1 by $\mu(e)$. In cases 1 and 3, the effect of $\mu(e)$ is to remove $I(\mu(e))(p)$ tokens from p and to produce an arbitrary number K of tokens in p. Hence, in case 1, by firing $\mu(e)$ from m_1, we obtain $I(\mu(e))(p) + m(p) - I(\mu(e))(p) + K = m(p) + K$ tokens in p. In case 3, by firing $\mu(e)$ from m_1, we obtain $\lambda(n_1)(p) - I(\mu(e))(p) + K$ tokens in p. In both cases, by letting $K = m(p)$, we obtain $m_2(p)$. In cases 2 and 4, the effect of $\mu(e)$ on place p is equal to $O(\mu(e))(p) - I(\mu(e))(p)$. Hence, in case 2, by firing $\mu(e)$ from m_1, we obtain $I(\mu(e))(p) + m(p) - I(\mu(e))(p) + O(\mu(e))(p) = m(p) + O(\mu(e))(p)$ tokens in p. In case 4, by firing $\mu(e)$ from m_1, we obtain $\lambda(n_1)(p) - I(\mu(e))(p) + O(\mu(e))(p)$ tokens in p. In both cases, these values correspond exactly to $m_2(p)$.

We conclude this case by observing that $\mathrm{nb}\omega\left(\lambda(n_1)\right) = \mathrm{nb}\omega\left(\lambda(n_2)\right)$ implies that no acceleration has been performed, which is the present case. We have thus shown that when $\mathrm{nb}\omega\left(\lambda(n_1)\right) = \mathrm{nb}\omega\left(\lambda(n_2)\right)$, $\sigma_\pi = \mu(e)$ is a sequence of transitions that satisfies the lemma.

CASE B: **the condition of the `if` in line 18 has been satisfied (an acceleration has occurred)**. Note that in this case, n_1 is the node called n in the condition of the **if**, and $\mu(e)$ is the transition called t in the same condition. Let $\overline{\sigma}$ be the sequence of transitions labelling the path from $\overline{n}$ to n_1. Let P^{Acc} denote the set of places: $\{p \mid \mathit{effect}(\overline{\sigma}(p)) > 0 \wedge \lambda(n_2)(p) \neq \omega \wedge O(\mu(e))(p) \neq \omega\}$. Let $K = \max_{p \in P^{Acc}}\{m(p)\}$ and let σ_π be the sequence of transitions $\mu(e)\big(\overline{\sigma} \cdot \mu(e)\big)^K$. From those definitions, only the following cases are possible, for all places p:

case	$\lambda(\overline{n})(p)$	$\lambda(n_1)(p)$	$\lambda(n_2)(p)$	$\mathit{effect}(\overline{\sigma})(p)$	$\mathit{effect}(\mu(e))(p)$	Remark
1	ω	ω	ω	$\in \mathbb{Z} \cup \{\omega\}$	$\in \mathbb{Z} \cup \{\omega\}$	
2	$\neq \omega$	$\neq \omega$	$\neq \omega$	$\neq \omega$	$\neq \omega$	
3	$\neq \omega$	$\neq \omega$	ω	$\neq \omega$	ω	
4	$\neq \omega$	$\neq \omega$	ω	$\neq \omega$	$\neq \omega$	$\mathit{effect}(\overline{\sigma} \cdot \mu(e))(p) > 0$

Only those four cases are possible because $\overline{n}$ is an ancestor of n_1, which is itself an ancestor of n_2. Moreover, by construction, $\mathrm{nb}\omega\,(\overline{n}) = \mathrm{nb}\omega\,(n_1)$, since those two nodes have been computed during the same recursive call. Thus, the occurrence of a fresh ω can only appear between n_1 and n_2, either because $\mathit{effect}(\mu(e))(p) = \omega$ (case 3), or because we have performed an acceleration (case 4). Note that the latter only occurs when $\mathit{effect}(\overline{\sigma} \cdot \mu(e))(p) > 0$.

Let us next define the marking m_1, as: (i) $m_1(p) = \lambda(n_1)(p)$ if $\lambda(n_1)(p) \neq \omega$; and (ii) $m_1(p) = I(\sigma_\pi)(p) + m(p)$ otherwise; where $I(\sigma_\pi)(p)$ denotes $\sum_{i=1}^{n} I(t_i)(p)$ for $\sigma_\pi = t_1, \dots, t_n$. Observe that, by definition: $m_1 \in \gamma(\lambda(n_1))$. Then, let us prove that σ_π is firable from m_1. First observe that, if p is a place s.t. $\lambda(n_1)(p) = \omega$, then $\mathsf{AllowsFiring}(\sigma_\pi, m_1, p)$ holds, because, in this case, $m_1(p) \geq I(\sigma_\pi)(p)$, by def. of m_1. Then, assume p is a place s.t. $\lambda(n_1)(p) \neq \omega$. In this case, by definition, $m_1(p) = \lambda(n_1)$. First observe that, by construction, and since we consider ωOPN (see line 6 of the algorithm):

$$\forall p : \lambda(n_1)(p) \geq I(\mu(e))(p) \tag{3}$$

Let us now consider all the possible cases, which are cases 2, 3 and 4 from the table above (case 1 cannot occur since we have assumed that $\lambda(n_1)(p) \neq \omega$):

- *In case 2*, since the condition of the **if** (line 18) is satisfied, we know that $\mathit{effect}(\overline{\sigma} \cdot \mu(e))(p) \geq 0$. Since $\lambda(\overline{n})(p) \neq \omega$, and $\lambda(n_1)(p) \neq \omega$, we can apply Lemma 3.3, and conclude that: $\lambda(n_2)(p) = \lambda(\overline{n})(p) + \mathit{effect}(\overline{\sigma} \cdot \mu(e))(p) = \lambda(\overline{n})(p) + \mathit{effect}(\overline{\sigma})(p) + \mathit{effect}(\mu(e))(p) = \lambda(n_1)(p) + \mathit{effect}(\mu(e))(p)$. Thus:

$$\lambda(n_1)(p) + \mathit{effect}(\mu(e))(p) \geq \lambda(\overline{n})(p) \tag{4}$$

  since $\mathit{effect}(\overline{\sigma} \cdot \mu(e))(p) \geq 0$. By applying CASE A (above) iteratively along the branch from $\overline{n}$ to n_1, we have $\mathsf{AllowsFiring}(\overline{\sigma}, \lambda(\overline{n}), p)$. Hence, $\mathsf{AllowsFiring}(\overline{\sigma}, \lambda(n_1)(p) + \mathit{effect}(\mu(e))(p), p)$ holds too, by (4). Finally, by (3), we conclude that $\mathsf{AllowsFiring}(\mu(e) \cdot \overline{\sigma}, \lambda(n_1)(p), p)$ holds. However, $\mathit{effect}(\mu(e) \cdot \overline{\sigma})(p) = \mathit{effect}(\overline{\sigma} \cdot \mu(e))(p) \geq 0$. Thus, since $\mu(e) \cdot \overline{\sigma}$ has a positive effect on p, we conclude that $\mathsf{AllowsFiring}\left((\mu(e) \cdot \overline{\sigma})^K, \lambda(n_1)(p), p\right)$ holds too, for all $K \geq 1$. Finally, since $\mathit{effect}\left((\mu(e) \cdot \overline{\sigma})^K\right)(p) \geq 0$, we conclude that $\lambda(n_1)(p) + \mathit{effect}\left((\mu(e) \cdot \overline{\sigma})^K\right) \geq \lambda(n_1)(p)$. Thus, by (3), $\lambda(n_1)(p) + \mathit{effect}\left((\mu(e) \cdot \overline{\sigma})^K\right) \geq I(\mu(e))$ and we can thus fire $\mu(e)$ once again after firing $(\mu(e) \cdot \overline{\sigma})^K$. Hence, $\mathsf{AllowsFiring}\left((\mu(e) \cdot \overline{\sigma})^K \cdot \mu(e), \lambda(n_1), p\right)$ holds, with $\sigma_\pi = (\mu(e) \cdot \overline{\sigma})^K \cdot \mu(e)$.

- *In case 3*: by (3), since $O(\mu(e))(p) = \omega$, and since $\mu(e)$ is the first transition of σ_π, we immediately conclude that $\mathsf{AllowsFiring}(\sigma_\pi, \lambda(n_1), p)$.

- *In case 4*, we can adapt the reasoning of case 2 as follows. First remember, that, in case 4, $\mathit{effect}(\overline{\sigma} \cdot \mu(e))(p) > 0$. Since $\lambda(\overline{n})(p) \neq \omega$, and $\lambda(n_1)(p) \neq \omega$, we can apply Lemma 3.3, and conclude that $\lambda(n_1)(p) = \lambda(\overline{n})(p) + \mathit{effect}(\overline{\sigma})(p)$. Thus: $\lambda(n_1)(p) + \mathit{effect}(\mu(e))(p) = \lambda(\overline{n})(p) + \mathit{effect}(\overline{\sigma})(p) + \mathit{effect}(\mu(e))(p) = \lambda(\overline{n})(p) + \mathit{effect}(\overline{\sigma} \cdot \mu(e))(p)$ with $\mathit{effect}(\overline{\sigma} \cdot \mu(e))(p) > 0$. Hence: $\lambda(n_1)(p) + \mathit{effect}(\mu(e))(p) > \lambda(\overline{n})(p)$. This implies (4), and we can reuse the arguments of case 2 to show that $\mathsf{AllowsFiring}\,(\sigma_\pi, \lambda(n_1), p)$ holds in the present case too.

Thus, for all p s.t. $\lambda(n_1)(p) \neq \omega$: $\mathsf{AllowsFiring}(\sigma_\pi, \lambda(n_1), p)$ holds. However, $\lambda(n_1)(p) \neq \omega$ implies that $m_1(p) = \lambda(n_1)(p)$, hence, $\mathsf{AllowsFiring}(\sigma_\pi, m_1, p)$ holds in those cases. Thus, we conclude that $\mathsf{AllowsFiring}(\sigma_\pi, m_1, p)$ holds for all places p, and thus, that σ_π is firable from m_1.

To conclude the proof let us build a marking m_2 that respects the conditions given in the statement of the lemma. Let $\overline{m}$ be a marking s.t. $m_1 \xrightarrow{\sigma_\pi} \overline{m}$. We know that such a marking exists since σ_π is firable from m_1. We first observe that, by Lemma 2.4:

$$\forall p \text{ s.t. } \mathit{effect}(\sigma_\pi)(p) \neq \omega : \overline{m}(p) = m_1(p) + \mathit{effect}(\sigma_\pi)(p) \tag{5}$$

From $\overline{m}$, we define the marking m_2 s.t. (i) $m_2(p) = \overline{m}(p)$ if $\mathit{effect}(\sigma_\pi)(p) \neq \omega$; and (ii) $m_2(p) = \max\{\overline{m}(p), m(p)\}$ otherwise. Clearly, $m_2 \succeq_{P'} \overline{m}$, for $P' = \{p \mid \mathit{effect}(\sigma_\pi)(p) = \omega$. Hence, by Lemma 2.5, $m_1 \xrightarrow{\sigma_\pi} m_2$ holds. Let us conclude the proof by showing that $m_2 \in \gamma(\lambda(n_2))$, and that $m_2 \geq m$, as requested. Since m has been assumed to be in $\gamma(\lambda(n_2))$ too, it is sufficient to show that for all place p: (i) $\lambda(n_2)(p) = \omega$ implies $m_2(p) \geq m$, and (ii) $\lambda(n_2)(p) \neq \omega$ implies $m_2(p) = \lambda(n_2)(p)$.

Thus, we consider each place p separately, by reviewing the four cases given in the table above:

- *In case 1*, $m_1(p) = I(\sigma_\pi)(p) + m(p)$ and $\lambda(n_2)(p) = \omega$. Let us show that $m_2(p) \geq m(p)$. We consider two further cases:

  1. either $\mathit{effect}(\sigma_\pi)(p) \neq \omega$. In this case:

  $$\begin{aligned} m_2(p) &= \overline{m}(p) && \text{By def of } m_2 \\ &= m_1(p) + \mathit{effect}(\sigma_\pi)(p) && \text{By (5)} \\ &= I(\sigma_\pi)(p) + \mathit{effect}(\sigma_\pi)(p) + m(p) && \text{By def. of } m_1 \\ &\geq m(p) \end{aligned}$$

  2. or $\mathit{effect}(\sigma_\pi)(p) = \omega$. Then, $m_2(p) \geq m(p)$ by def. of m_2.

- *In case 2*, we know that $\mathit{effect}(\mu(e))(p) \neq \omega$ and $\mathit{effect}(\overline{\sigma})(p) \neq \omega$, hence $\mathit{effect}(\overline{\sigma} \cdot \mu(e)) \neq \omega$ and $\mathit{effect}(\sigma_\pi) \neq \omega$ either. Then:

$$\begin{aligned} m_2(p) &= \overline{m}(p) && \text{By def. of } m_2 \\ &= m_1(p) + \mathit{effect}(\sigma_\pi)(p) && \text{By (5)} \\ &= \lambda(n_1)(p) + \mathit{effect}(\sigma_\pi)(p) && \text{By def of } m_1 \\ &= \lambda(n_2)(p) && \text{Lemma 3.3 and } \mathit{effect}(\overline{\sigma} \cdot \mu(e)) \neq \omega \end{aligned}$$

- *In case 3*, $\lambda(n_2)(p) = \omega$ and $\mathit{effect}(\sigma_\pi)(p) = \omega$ too. Hence, $m_2(p) \geq m(p)$ by def. of m_2.

- *In case 4*, $\lambda(n_2)(p) = \omega$ again, and $m_1(p) = \lambda(n_1)(p)$, by def. of m_1. Moreover, we have $\mathit{effect}(\sigma_\pi)(p) \neq \omega$, because $\mathit{effect}(\overline{\sigma})(p) \neq \omega$ and $\mathit{effect}(\mu(e))(p) \neq \omega$. Finally, since in case 4, we have $\mathit{effect}(\overline{\sigma} \cdot \mu(e))(p) > 0$, and since $\sigma_\pi = \mu(e)\big(\overline{\sigma} \cdot \mu(e)\big)^K$, we conclude that $\mathit{effect}(\sigma_\pi)(p) \geq K - \mathit{effect}(\mu(e))(p)$. Thus:

$$\begin{aligned}
m_2(p) &= \overline{m}(p) && \text{By def. of } m_2\\
&= m_1(p) + \mathit{effect}(\sigma_\pi)(p) && \text{By (5)}\\
&\geq m_1(p) + K - \mathit{effect}(\mu(e))(p) && \text{See above}\\
&= m_1(p) + K - I(\mu(e))(p) + O(\mu(e))(p) && \text{Def. of } \mathit{effect}\\
&\geq K + m_1(p) - I(\mu(e))(p) &&\\
&\geq K + \lambda(n_1)(p) - I(\mu(e))(p) && \text{By def. of } m_1\\
&\geq K && \text{By (3)}\\
&\geq m(p) && p \in P^{Acc} \text{ and by def of } \sigma^\pi
\end{aligned}$$

□

We are now ready to prove the soundness result stated in Lemma 3.2:

Proof:
We build, by induction on the length k of the path in the tree, a corresponding execution of $\mathcal{N}$. The induction works backward, starting from the end of the path.
Base case, $k = 0$. Since $n_k = n_0$, we can take $m_0 = m$, which clearly satisfies the Lemma since $m \in \lambda(n_k) = \lambda(n_0)$.
Inductive case, $k > 0$. The induction hypothesis is that there are a sequence of transitions σ and two markings m_1 and m_k s.t. $m_1 \xrightarrow{\sigma} m_k$, $m_1 \in \gamma(\lambda(n_1))$, $m_k \in \gamma(\lambda(n_k))$, and $m_k \geq m$. In the case where (n_0, n_1) is not an edge of $\mathcal{T}$ (i.e., n_1 is an ancestor of n_0), we know that $\lambda(n_0) = \lambda(n_1)$ by definition of stuttering and let $\rho_\pi = m_1 \xrightarrow{\sigma} m_k$. Otherwise, we can apply Lemma 3.4, and conclude that there are σ', m_0 and m'_1 s.t. $m_0 \xrightarrow{\sigma'} m'_1$, $m_0 \in \gamma(\lambda(n_0))$, $m'_1 \in \gamma(\lambda(n_1))$ and $m'_1 \succeq m_1$. Since $m'_1 \succeq m_1$, σ is also firable from m'_1. Let $m'_k = m'_1 + (m_k - m_1)$. Clearly, $m_0 \xrightarrow{\sigma'} m'_1 \xrightarrow{\sigma} m'_k$. Moreover, $m'_k \succeq m_k \succeq m$, by monotonicity. Let us show that $m'_k \in \gamma(\lambda(n_k))$. Since m'_1 and m_1 are both in $\gamma(\lambda(n_1))$: $m_1(p) = m'_1(p)$ for all p s.t. $\lambda(n_1)(p) \neq \omega$. Thus, by strong monotonicity, we conclude that $m_k(p) = m'_k(p)$ for all p s.t. $\lambda(n_1)(p) \neq \omega$. However, for all places p, $\lambda(n_k)(p) \neq \omega$ implies $\lambda(n_1)(p) \neq \omega$, as the number of ω's increase along a path in the tree. Thus we conclude that $m_k(p) = m'_k(p)$ for all p s.t. $\lambda(n_k)(p) \neq \omega$. Since $m_k(p) = \lambda(n_k)(p)$ for all p s.t. $\lambda(n_k)(p) \neq \omega$ because $m_k \in \gamma(\lambda(n_k))$ by induction hypothesis, we conclude that $m'_k \in \gamma(\lambda(n_k))$ too. Thus, m_0, m'_k and $\sigma' \cdot \sigma$ fulfill the statement of the lemma.

Finally, observe that, when all the nodes along the path π have the same number of ω's, Lemma 3.4 guarantees that $\mu(\pi)$ can be chosen for the sequence of transitions σ. □

Completeness Proving completeness amounts to showing that every execution (starting from m_0) of an ωPN $\mathcal{N}$ is witnessed by a stuttering path in Build-KM$(\mathcal{N}, m_0)$. It relies on the following property:

Lemma 3.5. Let $\mathcal{N}$ be an ωOPN, let m_0 be an ω-marking, and let $\mathcal{T}$ be the tree returned by Build-KM$(\mathcal{N}, m_0)$. Then, for all nodes n of Build-KM$(\mathcal{N}, m_0)$: (i) either n has no successor in the tree and has an ancestor $\overline{n}$ s.t. $\lambda(\overline{n}) = \lambda(n)$; or (ii) the set of successors of n corresponds to all the $\rightarrow_\omega$ possible successors of $\lambda(n)$, i.e.: $\{\mu(n, n') \mid (n, n') \in E\} = \{t \mid \lambda(n) \xrightarrow{t}_\omega\}$. Moreover, for each n' s.t. $(n, n') \in E$ and $\mu(n, n') = t$: $\lambda(n') \succeq \lambda(n) + \mathit{effect}(t)$.

Proof:
Observe that each time a node is created, it is inserted into `U`, or a recursive call is performed on this node. In both cases, the node will eventually be considered in line 5. If the condition of the `if` in line 5 is not satisfied, n has an ancestor $\overline{n}$ s.t. $\lambda(\overline{n}) = \lambda(n)$. Otherwise, all transitions t that are firable from $\lambda(n)$ are considered in the loop in lines 6 onward, and a corresponding edge (n, n') with $\mu(n, n') = t$ is added to the tree in line 15. The label $\lambda(n')$ of this node is either $\lambda(n) + \mathit{effect}(t)$, or a $\succeq$-larger marking, in the case where an acceleration has been performed during the `Post`, in line 19. Thus in both cases, $\lambda(n') \succeq \lambda(n) + \mathit{effect}(t)$. The algorithm terminates because `U` has become empty. Thus, all the nodes that have eventually been constructed by the algorithm fall into these two cases. Hence the Lemma. □

We can now state the completeness property:

Lemma 3.6. Let $\mathcal{N}$ be an ωOPN with set of transitions T, let m_0 be an initial marking and let $m_0 \xrightarrow{t_1} m_1 \xrightarrow{t_2} \cdots \xrightarrow{t_n} m_n$ be an execution of $\mathcal{N}$. Then, there are a *stuttering* path $\pi = n_0, n_1, \ldots, n_k$ in `Build-KM`$(\mathcal{N}, m_0)$ and a monotonic increasing mapping $h : \{1, \ldots, n\} \mapsto \{0, \ldots, k\}$ s.t.: $\mu(\pi) = t_1 t_2 \cdots t_n$ and $m_i \preceq \lambda(n_{h(i)})$ for all $0 \leq i \leq n$.

Proof:
The proof is by induction on the length of the execution.
Base case: $n = 0$ We let $h(0) = 0$. By construction $\lambda(n_0) = m_0$, hence the lemma.
Inductive case: $n > 0$ The induction hypothesis is that there are a path $\pi = n_0, \ldots n_\ell$ and a mapping $h : \{0, \ldots, n-1\} \mapsto \{0, \ldots, \ell\}$ satisfying the lemma for the execution prefix $m_0 \xrightarrow{t_1} m_1 \xrightarrow{t_2} \cdots \xrightarrow{t_{n-1}} m_{n-1}$. By Lemma 3.5, we consider two cases for n_ℓ. The first case is when the set of successors of n_ℓ corresponds to the set of all transitions that are firable from $\lambda(n_\ell)$. Since, by induction hypothesis, $n_\ell \succeq m_{n-1}$, and since t_n is firable from m_{n-1}, we conclude that t_n is firable from $\lambda(n_\ell)$ by monotonicity. Hence, n_ℓ has a successor n s.t. $\mu(n_\ell, n) = t_n$. Still by Lemma 3.5, $\lambda(n) \succeq \lambda(n_\ell) + \mathit{effect}(t_n) \succeq m_{n-1} + \mathit{effect}(t_n) \succeq m_n$. Hence, we let $n_{\ell+1} = n$, and $h(n) = \ell + 1$. The second case is when the set of successors of n_ℓ is empty. In this case, by Lemma 3.5, there exists an ancestor n of n_ℓ s.t. $\lambda(n) = \lambda(n_\ell)$. Let $n_{\ell+1}$ be such a node. Moreover, as $n_{\ell+1} \neq n_\ell$, and $n_{\ell+1}$ is an ancestor of n_ℓ, $n_{\ell+1}$ must have at least one successor. Hence, by Lemma 3.5, $n_{\ell+1}$ is fully developed, and we can apply the same reasoning as above to conclude that there is a successor n' of $n_{\ell+1}$ s.t. $\lambda(n') \succeq m_n$ and $\mu(n_{\ell+1}, n') = t_n$. Let $n_{\ell+2}$ be such a node. We conclude by letting $h(n) = \ell + 2$. □

3.3. From ωOPN to ωPN

We have shown completeness and soundness of the `Build-KM` algorithm for ωOPN. Let us show that each ωPN $\mathcal{N}$ can be turned into an ωOPN $\mathsf{remI}\omega(\mathcal{N})$ that (i) terminates iff $\mathcal{N}$ terminates and (ii) that has the same coverability sets as $\mathcal{N}$. The ωOPN $\mathsf{remI}\omega(\mathcal{N})$ is obtained from $\mathcal{N}$ by replacing each transition $t \in T$ by a transition $t' \in T'$ s.t. $O(t') = O(t)$ and $I(t') = \{I(t)(p) \otimes p \mid I(t)(p) \neq \omega\}$. Intuitively, t' is obtained from t by deleting all ω input arcs. Since t' always consumes *less tokens* than t does, the following is easy to establish:

Lemma 3.7. Let $\mathcal{N}$ be an ωPN. For all executions $m_0, t'_1, m_1, \ldots, t'_n, m_n$ of $\mathsf{reml}\omega(\mathcal{N})$: $m_0, t_1, m_1, \ldots, t_n, m_n$ is an execution of $\mathcal{N}$. For all finite (resp. infinite) executions $m_0, t_1, m_1, \ldots, t_n, m_n$ ($m_0, t_1, m_1, \ldots, t_j, m_j, \ldots$) of $\mathcal{N}$, there is an execution $m_0, t'_1, m'_1, \ldots, t'_n, m'_n$ ($m_0, t_1, m'_1, \ldots, t_j, m'_j, \ldots$) of $\mathsf{reml}\omega(\mathcal{N})$, s.t. $m_i \preceq m'_i$ for all i.

Proof:
The first point follows immediately from the definition of $\mathsf{reml}\omega(\mathcal{N})$ and from the fact that consuming 0 tokens in each place p s.t. $I(t_i)(p) = \omega$ is a valid choice when firing each transition t_i in $\mathcal{N}$. The second point is easily shown by induction on the execution, because firing each t_i produces the same amount of tokens that t'_i; consumes the same amount of token as each t'_i in all places s.t. $I(t_i)(p) \neq \omega$, and consumes, in each place p s.t. $I(t_i)(p) = \omega$ a number of tokens that is larger than or equal to the number of tokens consumed by t'_i. □

Intuitively, this means that, when solving coverability, (place) boundedness or termination on an ωPN $\mathcal{N}$, we can analyse $\mathsf{reml}\omega(\mathcal{N})$ instead, because $\mathcal{N}$ terminates iff $\mathsf{reml}\omega(\mathcal{N})$ terminates, and removing the ω-labeled input arcs from $\mathcal{N}$ does not allow to reach higher markings. Finally, we observe that, for all ωPN $\mathcal{N}$, and all initial marking m_0: the trees returned by $\texttt{Build-KM}(\mathcal{N}, m_0)$ and $\texttt{Build-KM}(\mathsf{reml}\omega(\mathcal{N}, m_0))$ respectively are isomorphic[6]. This is because we have defined $c - \omega$ to be equal to c: applying this rule when computing the effect of a transition t (line 17), is equivalent to computing the effect of the corresponding t' in $\mathsf{reml}\omega(\mathcal{N})$, i.e. letting $I(t')(p) = 0$ for all p s.t. $I(t)(p) = \omega$. Thus, we can lift Lemma 3.2 and Lemma 3.6 to ωPN. This establish correctness of the algorithm for the general ωPN case.

3.4. Applications of the Karp & Miller tree

With these results we conclude that the Karp & Miller tree can be used to compute a coverability set and to decide termination of ωPN.

Theorem 3.8. Let $\mathcal{N}$ be an ωPN with initial marking m_0, and let $\mathcal{T} = \langle N, E, \lambda, \mu, n_0 \rangle$ be the tree returned by $\texttt{Build-KM}(\mathcal{N}, m_0)$. Then: (i) $\lambda(N)$ is a coverability set of $\mathcal{N}$ and (ii) $\mathcal{N}$ does not terminate iff $\mathcal{T}$ contains an ω-maximal self-covering stuttering path.

Proof:
Point (i) follows from Lemma 3.2 (lifted to ωPN). Let us now prove both directions of point (ii).

First, we show that if $\texttt{Build-KM}(\mathcal{N}, m_0)$ contains an ω-maximal self-covering stuttering path, then $\mathcal{N}$ admits a self-covering execution from m_0. Let $n_0, \ldots, n_k, n_{k+1}, \ldots, n_\ell$ be an ω-maximal self-covering stuttering path, and assume that $\mathit{effect}(n_{k+1}, \ldots, n_\ell) \geq \mathbf{0}$. Let us apply Lemma 3.2 (lifted to ωPN), by letting $m = \mathbf{0}$ and $\pi = \pi_2$, and let m_1 and m_2 be markings s.t. $m_1 \xrightarrow{\mu(\pi_2)} m_2$. The existence of m_1 and m_2 is guaranteed by Lemma 3.2 (lifted to ωPN), because all the nodes along π_2 have the same number of ω's as we are considering an ω-*maximal* self-covering stuttering path. Since

[6] That is, if $\texttt{Build-KM}(\mathcal{N}, m_0)$ returns $\langle N, E, \lambda, \mu, n_0 \rangle$ and $\texttt{Build-KM}(\mathsf{reml}\omega(\mathcal{N}, m_0))$ returns $\langle N', E', \lambda', \mu', n'_0 \rangle$, then, there is a bijection $h : N \mapsto N'$ s.t. (i) $h(n_0) = n'_0$, (ii) for all $n \in N$: $\lambda(n) = \lambda(h(n))$, (iii) for all n_1, n_2 in N: $(n_1, n_2) \in E$ iff $(h(n_1), h(n_2)) \in E'$, (iv) for all $(n_1, n_2) \in E$: $\mu(n_1, n_2) = \mu'(h(n_1), h(n_2))$.

effect(π_2) is positive, so is *effect*$(\mu(\pi_2))$. Thus, there exists[7] m_2' s.t. $m_1 \xrightarrow{\mu(\pi_2)} m_2'$ and $m_2' \succeq m_1$. By invoking Lemma 3.2 (lifted to ωPN) again, letting $\pi = \pi_1$ and $m = m_1$, we conclude to the existence of a sequence of transitions σ, a marking m_0 and a marking $m_1' \succeq m_1$ s.t. $m_0 \xrightarrow{\sigma} m_1'$. Since $m_1' \succeq m_1$, $\mu(\pi_2)$ is again firable from m_1'. Let $\overline{m}_2 = m_2 + m_1' - m_1$. Clearly, $m_1' \xrightarrow{\mu(\pi_2)} \overline{m}_2$, with $\overline{m}_2 \succeq m_1'$. Hence, $m_0 \xrightarrow{\sigma} m_1' \xrightarrow{\mu(\pi_2)} \overline{m}_2$ is a self-covering execution of $\mathcal{N}$.

Second, let us show that, if $\mathcal{N}$ admits a self-covering execution from m_0, then $\texttt{Build-KM}(\mathcal{N}, m_0)$ contains an ω-maximal self-covering stuttering path. Let $\rho = m_0 \xrightarrow{t_1} m_1 \cdots \xrightarrow{t_n} m_n$ be a self-covering execution and assume $0 \leq k < n$ is a position s.t. $m_k \preceq m_n$. Let σ_1 denote $t_1, \dots t_k$ and σ_2 denote $t_{k+1}, \dots t_n$. Let us consider the execution ρ', defined as follows

$$\rho' = m_0 \xrightarrow{\sigma_1} m_k \underbrace{\xrightarrow{t_{k+1}} m_{k+1} \cdots \xrightarrow{t_n} m_n}_{\sigma_2} \underbrace{\xrightarrow{t_{k+1}} m_{n+1} \cdots \xrightarrow{t_n} m_{2n-k}}_{\sigma_2} \cdots$$
$$\cdots \underbrace{\xrightarrow{t_{k+1}} m_{(|P|+1)n-|P|k+1} \cdots \xrightarrow{t_n} m_{(|P|+2)n-(|P|+1)k}}_{\sigma_2}$$

where for all $n+1 \leq j \leq (|P|+2)n - (|P|+1)k$: $m_j - m_{j-1} = m_{f(j)} - m_{f(j-1)}$ with f the function defined as $f(x) = \big((x-k) \mod (n-k)\big) + k$ for all x. Intuitively, ρ' amounts to firing $\sigma_1(\sigma_2)^{|P|+1}$ (where P is the set of places of $\mathcal{N}$) from m_0, by using, each time we fire σ_2, the same effect as the one that was used to obtain ρ (remember that the effect of σ_2 is non-deterministic when ω's are produced). It is easy to check that ρ' is indeed an execution of $\mathcal{N}$, because ρ is a self-covering execution.

Let $n_0, n_1, \dots n_\ell$ and h be the stuttering path in $\texttt{Build-KM}(\mathcal{N}, m_0)$ and the mapping corresponding to ρ' (and whose existence is established by Lemma 3.6). Since, $m_k \preceq m_n$, *effect*$(t_{k+1} \cdots t_n) \geq \mathbf{0}$ and by Lemma 3.6 (lifted to ωPN), all the following stuttering paths of the form $n_0, \dots, n_{h(j \times n-(j-1)\times k)}$, for $1 \leq j \leq |P|+2$, are self-covering. Let us show that one of them is ω-maximal, i.e. that there is $1 \leq j \leq |P|+1$ s.t. $\mathrm{nb}\omega\left(n_{h(jn-(j-1)k)}\right) = \mathrm{nb}\omega\left(n_{h((j+1)n-jk)}\right)$. Assume it is not the case. Since the number of ω's can only increase along a stuttering path, this means that

$$0 \leq \mathrm{nb}\omega\left(n_{h(n)}\right) < \mathrm{nb}\omega\left(n_{h(2n-k)}\right) < \mathrm{nb}\omega\left(n_{h(3n-2k)}\right) < \mathrm{nb}\omega\left(n_{h((|P|+2)n-(|P|+1)k)}\right)$$

However, this implies that $\mathrm{nb}\omega\left(n_{h((|P|+2)n-(|P|+1)k)}\right) > |P|$, which is not possible as P is the set of places of $\mathcal{N}$. Hence, we conclude that there exists an ω-maximal self-covering stuttering path in $\texttt{Build-KM}(\mathcal{N}, m_0)$. □

4. From ωPN to plain PN

Let us show that we can, from any ωPN $\mathcal{N}$, build a plain PN $\mathcal{N}'$ whose set of reachable markings allows us to recover the reachability set of $\mathcal{N}$. This construction allows us to solve reachability, coverability and (place) boundednes using algorithms on Petri nets [14, 15, 24, 27]. The idea of the construction is depicted in Fig. 4, and can be outlined as follows. A transition t in the ωPN is simulated in *three steps*

[7] Note that although *effect*$(\mu(\pi_2)) \succeq \mathbf{0}$, we have no guarantee that $m_2 \succeq m_1$, as we could have *effect*$(\mu(\pi_2)) = \omega$ for some p, and maybe the amount of tokens that has been produced in p by $\mu(\pi_2)$ to yield m_2 does not allow to have $m_2(p) \geq m_1(p)$. However, in this case, it is always possible to reach a marking with enough tokens in p to cover $m_1(p)$, since *effect*$(\mu(\pi_2)) = \omega$.

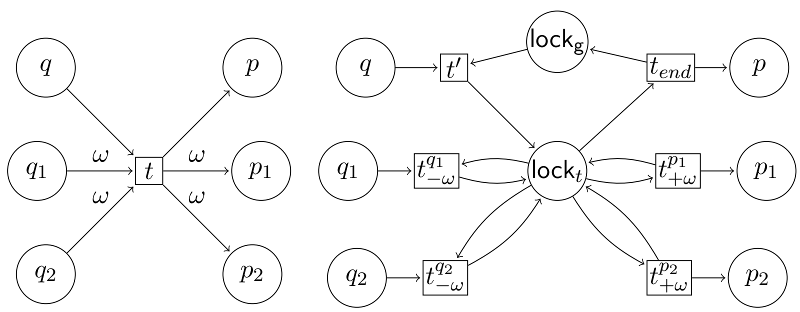

Figure 4. Transforming an ωPN into a plain PN.

in the PN. First, t' fires, which (i) moves a token from the *global lock* $\mathsf{lock_g}$ to the local lock lock_t and (ii) consumes the same fixed amount of tokens as t, i.e., if $I_t(p) \neq \omega$, then, t' consumes $I_t(p)$ tokens in p, for all p. Once t' has fired, all transitions are blocked but the $t_{-\omega}^{q_i}$ and $t_{+\omega}^{p_i}$ transitions, that can be fired an arbitrary number of times to simulate the ω-arcs of t. Finally, t_{end} moves the lock back to $\mathsf{lock_g}$, and produces $O_t(p)$ tokens in all p s.t. $O_t(p) \neq \omega$.

Formally, we turn the ωPN $\mathcal{N} = \langle P, T, m_0 \rangle$ into a plain PN $\mathcal{N}' = \langle P', T', m_0' \rangle$ using the following procedure. Assume that $T = T_{plain} \uplus T_\omega$, where T_ω is the set of ω-transitions of $\mathcal{N}$. Then:

1. We add to the net one place (called the *global lock*) $\mathsf{lock_g}$, and for each ω-transition t, one place lock_t. That is, $P' = P \cup \{\mathsf{lock_g}\} \cup \{\mathsf{lock}_t \mid t \in T_\omega\}$.

2. Each transition t in $\mathcal{N}$ is replaced by a set of transitions T_t in $\mathcal{N}'$. In the case where t is a plain transition, T_t contains a single transition that has the same effect as t, except that it also tests for the presence of a token in $\mathsf{lock_g}$. In the case where t is an ω-transition, T_t is a set of plain transitions that simulate the effect of t, as in Fig. 4. Formally, $T' = \cup_{t \in T} T_t$, where the T_t sets are defined as follows. If t is a plain transition, then $T_t = \{t'\}$, where, $I(t') = I(t) \cup \{\mathsf{lock_g}\}$ and $O(t') = O(t) \cup \{\mathsf{lock_g}\}$. If t is an ω-transition, then: $T_t = \{t', t_{end}\} \cup \{t_{-\omega}^p \mid I(t)(p) = \omega\} \cup \{t_{+\omega}^p \mid O(t)(p) = \omega\}$ where $I(t') = I(t) + \{\mathsf{lock_g}\}$; $O(t') = I(t_{end}) = \{\mathsf{lock}_t\}$; $O(t_{end}) = \{\mathsf{lock_g}\} + O(t)$. Furthermore, for all p s.t. $I(t)(p) = \omega$: $I(t_{-\omega}^p) = \{p, \mathsf{lock}_t\}$ and $O(t_{-\omega}^p) = \{\mathsf{lock}_t\}$. Finally, for all p s.t. $O(t)(p) = \omega$: $I(t_{+\omega}^p) = \{\mathsf{lock}_t\}$ and $O(t_{-\omega}^p) = \{p, \mathsf{lock}_t\}$.

3. We let f be the function that associates each marking m of $\mathcal{N}$ to the marking $f(m)$ of $\mathcal{N}'$ s.t. $m'(\mathsf{lock_g}) = 1$; for all $p \in P$: $m'(p) = m(p)$; and for all $p \notin P \cup \{\mathsf{lock_g}\}$: $m'(p) = 0$. Then, the initial marking of $\mathcal{N}'$ is $f(m_0)$.

Lemma 4.1. Let $\mathcal{N}$ be an ωPN and let $\mathcal{N}'$ be its corresponding PN. Then $m \in \mathsf{Reach}(\mathcal{N})$ iff $f(m) \in \mathsf{Reach}(\mathcal{N}')$.

Since the above construction can be carried out in polynomial time, the complexities for reachability [19, 20], (place) boundedness and coverability [22] for PNs carry on to ωPN:

Corollary 4.2. Reachability for ωPN is decidable and EXPSPACE-hard. Coverability, boundedness and place boundedness for ωPN are EXPSPACE-c.

This justifies the result given in Table 1 for reachability, coverability and (place) boundedness, for ωPN.

From our point of view, this translation from ωPN to plain Petri nets does not lower the interest of the ωPN model. Indeed, as argued in the introduction, ω-arcs are a natural way of modeling the creation and consumption of a non-deterministically chosen amount of resources (processus for instance). Moreover, the translation incurs a (polynomial) blow-up in the number of places and transitions. But, most of all, the above construction *does not preserve termination*, which motivates, in the first place, the introduction of ωPN (recall the example from the introduction). For instance, assume that the leftmost part of Fig. 4 is an ωPN $\mathcal{N} = \langle P, T, m_0 \rangle$ with $m_0(q) = 1$. Clearly, all executions of $\mathcal{N}$ are finite, while $t'(t^{p_1}_{+\omega})^\omega$ is an infinite transition sequence that is firable in $\mathcal{N}'$. However, observe that this infinite transition sequence is "unfair" in the sense that it never fires t_{end} even though it is enabled infinitely often. Formally, an infinite transition sequence has *finite delay property* with respect to sets $T_1, \ldots, T_k \subseteq T$ of transitions if for every T_j, either 1) infinitely many transitions of the sequence are from T_j or 2) infinitely many positions of the sequence disable all transitions in T_j. The problem of checking the existence of a firable infinite transition sequence with the finite delay property (called the *weak fairness problem*) is decidable [17]. By setting $T_1 = \{t^{p_1}_{+\omega}, t^{p_2}_{+\omega}\}$ and $T_2 = \{t_{end}\}$ in the Petri net on the right part of Fig. 4, the termination problem for ωPN can be reduced to the weak fairness problem for plain Petri nets. However, the proof of decidability in [17] uses a reduction to the reachability problem, which is not known to be solvable by Karp & Miller trees. We have seen in Section 3 that Karp & Miller trees can be extended to solve termination directly on ωPN. Constructing these trees require non-primitive recursive space (and so do all known algorithms for the reachability problem), but problems solvable by Karp & Miller trees generally turn out to be in EXPSPACE. Indeed, in the next section we show that the Rackoff technique [22] can be generalised to ωPN, and prove that termination is EXPSPACE-c for ωPN.

5. Extending the Rackoff technique for ωPN

In this section, we extend the Rackoff technique to ωPN to prove the existence of short self-covering sequences. For applications of interest, such as the termination problem, it is sufficient to consider ωOPN, as proved in Lemma 3.7. Hence, we only consider ωOPN in this section. As in Rackoff's work [22], the idea here is to use small solutions of linear Diophantine equations to limit lengths of sequences. As in the work of Brazdil *et al.* [4], we modify the effect of a sequence of transitions to ensure that ω-transitions are fired at least once. But the results of [4], proved in the context of games, can not be used here directly for the termination problem.

As observed in [22], beyond some large values, it is not necessary to track the exact value of markings to solve some problems. We use threshold functions $h : \{0, \ldots, |P|\} \to \mathbb{N}$ to specify such large values. Let $\mathrm{nb}\overline{\omega}(m) = |\{p \in P \mid m(p) \in \mathbb{N}\}|$.

Definition 5.1. Let $h : \{0, \ldots, |P|\} \to \mathbb{N}$ be a threshold function. Given an ω-marking m, the markings $[m]_{h\to\omega}$ and $[m]_{\omega\to h}$ are defined as follows:

$$([m]_{h\to\omega})(p) = \begin{cases} m(p) & \text{if } m(p) < h(\mathrm{nb}\overline{\omega}(m)), \\ \omega & \text{otherwise.} \end{cases}$$

$$([m]_{\omega\to h})(p) = \begin{cases} m(p) & \text{if } m(p) \in \mathbb{N}, \\ h(\mathrm{nb}\overline{\omega}(m) + 1) & \text{otherwise.} \end{cases}$$

In $[m]_{h \to \omega}$, values that are too high are abstracted by ω. In $[m]_{\omega \to h}$, ω is replaced by the corresponding natural number. This kind of abstraction is formalized in the following threshold semantics.

Definition 5.2. Given an ωPN $\mathcal{N}$, a transition t, an ω-marking m that enables t and a threshold function h, we define the transition relation $\xrightarrow{t}_h$ as $m \xrightarrow{t}_h [m + \mathit{effect}(t)]_{h \to \omega}$.

The transition relation $\xrightarrow{t}_h$ is extended to sequences of transitions in the usual way. Note that if $m \xrightarrow{t}_h m'$, then $\omega(m) \subseteq \omega(m')$. In words, a place marked ω will stay that way along any transition in threshold semantics.

Let $R = \max\{|\,\mathit{effect}(t)(p)| \mid t \in T, p \in P, \mathit{effect}(t)(p) < \omega\}$. The following proposition says that ω can be replaced by large enough numbers without disabling sequences. The proof is by a routine induction on the length of sequences, using the fact that in an ωOPN, any transition can reduce at most R tokens from a place.

Proposition 5.3. For some ω-markings m_1 and m_2, suppose $m_1 \xrightarrow{\sigma}_h m_2$ and $\omega(m_2) = \omega(m_1)$. If m_1' is a marking such that $m_1' \preceq_{\omega(m_1)} m_1$ and $m_1'(p) \geq R|\sigma|$ for all $p \in \omega(m_1)$, then $m_1' \xrightarrow{\sigma} m_2'$ such that $m_2' \preceq_{\omega(m_2)} m_2$ and $m_2'(p) \geq m_1'(p) - R|\sigma|$.

Definition 5.4. Given an ω-marking m_1 and a threshold function h, an *ω-maximal threshold pumping sequence* (h-PS) enabled at m_1 is a sequence σ of transitions such that $m_1 \xrightarrow{\sigma}_h m_2$, $\mathit{effect}(\sigma) \geq \mathbf{0}$ and $\omega(m_2) = \omega(m_1)$.

In the above definition, note that we require $\mathit{effect}(\sigma)(p) \geq 0$ for *any* place p, irrespective of whether $m_1(p) = \omega$ or not.

Definition 5.5. Suppose σ is an ω-maximal h-PS enabled at m_1 and $\sigma = \sigma_1\sigma_2\sigma_3$ such that $m_1 \xrightarrow{\sigma_1}_h m_3 \xrightarrow{\sigma_2}_h m_3 \xrightarrow{\sigma_3}_h m_2$. We call σ_2 a *simple loop* if all intermediate ω-markings obtained while firing σ_2 from m_3 (except the last one, which is m_3 again) are distinct from one another.

In the above definition, since $m_3 \xrightarrow{\sigma_2}_h m_3$ and $m_1 \xrightarrow{\sigma_1\sigma_3}_h m_2$, one might be tempted to think that $\sigma_1\sigma_3$ is also an ω-maximal h-PS enabled at m_1. This is however not true in general, since there might be some $p \in \omega(m_1)$ such that $\mathit{effect}(\sigma_1\sigma_3)(p) < 0$ (which is compensated by σ_2 with $\mathit{effect}(\sigma_2)(p) > 0$). The presence of the simple loop σ_2 is required due to its compensating effect. The idea of the proof of the following lemma is that if there are a large number of loops, it is enough to retain a few to get a shorter ω-maximal h-PS.

Lemma 5.6. There is a constant d such that for any ωPN $\mathcal{N}$, any threshold function h and any ω-maximal h-PS σ enabled at some ω-marking m_1, there is an ω-maximal h-PS σ' enabled at m_1, whose length is at most $(h(\mathrm{nb}\overline{\omega}\,(m_1))2R)^{d|P|^3}$.

Proof:
This proof is similar to that of [22, Lemma 4.5], with some modifications to handle ω-transitions. It is organized into the following steps. (Step 1) We first associate a vector with a sequence of transitions to measure the effect of the sequence. This is the step that differs most from that of [22, Lemma 4.5]. The idea in this step is similar to the one used in [4, Lemma 7]. (Step 2) Next we remove some simple loops

from σ to obtain σ'' such that for every intermediate ω-marking m in the run $m_1 \xrightarrow{\sigma}_h m_2$, m also occurs in the run $m_1 \xrightarrow{\sigma''}_h m_2$. (Step 3) The sequence σ'' obtained above need not be a h-PS. With the help of the vectors defined in step 1, we formulate a set of linear Diophantine equations that encode the fact that the effects of σ'' and the simple loops that were removed in step 2 combine to give the effect of a h-PS. (Step 4) Then we use the result about existence of small solutions to linear Diophantine equations to construct a sequence σ' that meets the length constraint of the lemma. (Step 5) Finally, we prove that σ' is a h-PS enabled at m_1.

Step 1: Let $P_\omega \subseteq \omega(m_1)$ be the set of places p such that some transition t in σ has $\mathit{effect}(t)(p) = \omega$. If we ensure that for each place $p \in P_\omega$, some transition t with $\mathit{effect}(t)(p) = \omega$ is fired, we can ignore the effect of other transitions on p. This is formalized in the following definition of the effect of any sequence of transitions $\sigma_1 = t_1 \cdots t_r$. We define the function $\Delta_{P_\omega}[\sigma_1] : \omega(m_1) \to \mathbb{Z}$ as follows.

$$\Delta_{P_\omega}[\sigma_1](p) = \begin{cases} 1 & p \in P_\omega, \exists i \in \{1, \ldots, r\} : \mathit{effect}(t_i)(p) = \omega \\ 0 & p \in P_\omega, \forall i \in \{1, \ldots, r\} : \mathit{effect}(t_i)(p) \neq \omega \\ \sum_{1 \le i \le r} \mathit{effect}(t_i)(p) & \text{otherwise} \end{cases}$$

Step 2: Let $m_1 \xrightarrow{\sigma}_h m_2$. From Definition 5.4, we have $\omega(m_2) = \omega(m_1)$. From Definition 5.1, we infer that for any ω-marking m in the run $m_1 \xrightarrow{\sigma}_h m_2$, $m(p) < h(\mathrm{nb}\overline{\omega}\,(m_1))$ for all $p \in P \setminus \omega(m_1)$. Now we remove some simple loops from σ to obtain σ''. To obtain some bounds in the next step, we first make the following observations on loops. Let $|P \setminus \omega(m_1)| = r_1$. Suppose π is a simple loop. There can be at most $h(\mathrm{nb}\overline{\omega}\,(m_1))^{r_1}$ transitions in π, so $-h(\mathrm{nb}\overline{\omega}\,(m_1))^{r_1} R \le \Delta_{P_\omega}[\pi](p) \le h(\mathrm{nb}\overline{\omega}\,(m_1))^{r_1} R$ for any $p \in P$. Let $\vec{B}$ be the matrix whose set of columns is equal to $\{\Delta_{P_\omega}[\pi] \mid \pi \text{ is a simple loop}\}$. There are at most $(h(\mathrm{nb}\overline{\omega}\,(m_1))^{r_1} 2R)^{|P|}$ columns in $\vec{B}$. We use $\vec{b}, \vec{b}', \ldots$ to denote the columns of $\vec{B}$.

Now we remove simple loops from σ according to the following steps. Let $\vec{x}_0 = \mathbf{0}$ be the zero vector whose dimension is equal to the number of columns in $\vec{B}$. Begin the following steps with $i = 0$ and $\sigma_i = \sigma$. (a) Think of the first $(h(\mathrm{nb}\overline{\omega}\,(m_1))^{|P|} + 1)^2$ transitions of σ_i as $h(\mathrm{nb}\overline{\omega}\,(m_1))^{|P|} + 1$ blocks of length $h(\mathrm{nb}\overline{\omega}\,(m_1))^{|P|} + 1$ each. (b) There is at least one block in which all ω-markings also occur in some other block. (c) Let π be a simple loop occurring in the above block. (d) Let σ_{i+1} be the sequence obtained from σ_i by removing π. (e) Let $\vec{x}_{i+1}$ be the vector obtained from $\vec{x}_i$ by incrementing $\vec{x}_i(\Delta_{P_\omega}[\pi])$ by 1. (f) Increment i by 1. (g) If the length of the remaining sequence is more than or equal to $(h(\mathrm{nb}\overline{\omega}\,(m_1))^{|P|} + 1)^2$, go back to step a. Otherwise, stop.

Let n be the value of i when the above process stops. Let $\sigma'' = \sigma_n$ and $\vec{x} = \vec{x}_n$. We remove a simple loop π starting at an ω-marking m only if all the intermediate ω-markings occurring while firing π from m occur at least once more in the remaining sequence. Hence, for every ω-marking m arising while firing σ from m_1, m also arises while firing σ'' from m_1. We have $|\sigma''| \le (h(\mathrm{nb}\overline{\omega}\,(m_1))^{|P|} + 1)^2$. For each column $\vec{b}$ of $\vec{B}$, $\vec{x}(\vec{b})$ contains the number of occurrences of simple loops π removed from σ such that $\Delta_{P_\omega}[\pi] = \vec{b}$.

Step 3: For every $p \in P_\omega$, we want to ensure that there is some transition t in the shorter h-PS that we will build, such that $\mathit{effect}(t)(p) = \omega$. For the other places, we want to ensure that the effect of the shorter h-PS is non-negative. These requirements are expressed in the vector $\vec{d}$, where $\vec{d}(p) = 1$ if $p \in P_\omega$ and $\vec{d}(p) = 0$ if $p \notin P_\omega$.

Recall that for each column $\vec{b}$ of $\vec{B}$, $\vec{x}(\vec{b})$ contains the number of occurrences of simple loops π removed from σ such that $\Delta_{P_\omega}[\pi] = \vec{b}$ and that σ'' is the sequence remaining after all removals. Hence,

$\Delta_{P_\omega}[\sigma] = \vec{B}\vec{x} + \Delta_{P_\omega}[\sigma'']$. Since σ is a h-PS and for every $p \in P_\omega$, there is a transition t in σ such that $\mathit{effect}(t)(p) = \omega$, we have

$$\Delta_{P_\omega}[\sigma] \geq \vec{d} \Rightarrow \vec{B}\vec{x} + \Delta_{P_\omega}[\sigma''] \geq \vec{d} \Rightarrow \vec{B}\vec{x} \geq \vec{d} - \Delta_{P_\omega}[\sigma'']\ . \tag{6}$$

Step 4: We use the following result about the existence of small integral solutions to linear equations [3], which has been used by Rackoff to give EXPSPACE upper bound for the boundedness problems in Petri nets [22, Lemma 4.4].

Let $d_1, d_2 \in \mathbb{N}^+$, let $\vec{A}$ be a $d_1 \times d_2$ integer matrix and let $\vec{a}$ be an integer vector of dimension d_1. Let $d \geq d_2$ be an upper bound on the absolute value of the integers in $\vec{A}$ and $\vec{a}$. Suppose there is a vector $\vec{x} \in \mathbb{N}^{d_2}$ such that $\vec{A}\vec{x} \geq \vec{a}$. Then for some constant c independent of d, d_1, d_2, there exists a vector $\vec{y} \in \mathbb{N}^{d_2}$ such that $\vec{A}\vec{y} \geq \vec{a}$ and $\vec{y}(i) \leq d^{cd_1}$ for all i between 1 and d_2.

We apply the above result to (6). Each entry of $\Delta_{P_\omega}[\sigma'']$ is of absolute value at most $(h(\mathrm{nb}\varpi\,(m_1))^{|P|} + 1)^2 R$. Recall that there are at most $(h(\mathrm{nb}\varpi\,(m_1))^{r_1} 2R)^{|P|}$ columns in $\vec{B}$, with the absolute value of each entry at most $h(\mathrm{nb}\varpi\,(m_1))^{r_1} R$. There are $|P| - r_1$ rows in $\vec{B}$. Hence, we conclude that $\vec{x}$ can be replaced by $\vec{y}$ such that $\vec{B}\vec{y} \geq \vec{d} - \Delta_{P_\omega}[\sigma'']$ and the sum of all entries in $\vec{y}$ is at most $(h(\mathrm{nb}\varpi\,(m_1))2R)^{d'|P|^3}$ for some constant d'. This expression is obtained from simplifying

$$(h(\mathrm{nb}\varpi\,(m_1))^{r_1} 2R)^{|P|}((h(\mathrm{nb}\varpi\,(m_1))^{|P|} + 1)^2 2R)^{d''|P|^2}$$

for some constant d''.

For each column $\vec{b}$ of $\vec{B}$, let $\pi_{\vec{b}}$ be a simple loop of σ such that $\Delta_{P_\omega}[\pi_{\vec{b}}] = \vec{b}$. Recall from step 2 that there is some intermediate ω-marking $m_{\vec{b}}$ occurring while firing σ'' from m_1 such that $m_{\vec{b}}$ is the ω-marking from which the simple loop $\pi_{\vec{b}}$ is fired in σ. Let $i_{\vec{b}}$ be the position in σ'' where $m_{\vec{b}}$ occurs. Let σ' be the sequence obtained from σ'' by inserting $\vec{y}(\vec{b})$ copies of $\pi_{\vec{b}}$ into σ'' at the position $i_{\vec{b}}$ for each column $\vec{b}$ of $\vec{B}$. Since we insert at most $(h(\mathrm{nb}\varpi\,(m_1))2R)^{d'|P|^3}$ simple loops, each of length at most $h(\mathrm{nb}\varpi\,(m_1))^{r_1}$, $|\sigma'| \leq (h(\mathrm{nb}\varpi\,(m_1))2R)^{d'|P|^3} h(\mathrm{nb}\varpi\,(m_1))^{r_1} + (h(\mathrm{nb}\varpi\,(m_1))^{|P|} + 1)^2$. Choose the constant d s.t. $|\sigma'| \leq (h(\mathrm{nb}\varpi\,(m_1))2R)^{d'|P|^3} \times h(\mathrm{nb}\varpi\,(m_1))^{r_1} + (h(\mathrm{nb}\varpi\,(m_1))^{|P|} + 1)^2 \leq (h(\mathrm{nb}\varpi\,(m_1))2R)^{d|P|^3}$. Now we conclude that $|\sigma'| \leq (h(\mathrm{nb}\varpi\,(m_1))2R)^{d|P|^3}$.

Step 5: Now we prove that σ' is a h-PS enabled at m_1. Recall that $m_1 \xrightarrow{\sigma}_h m_2$ and that σ' is obtained from σ by removing or adding extra copies of some simple loops. We infer that $m_1 \xrightarrow{\sigma'}_h m_2$. Now we show that $\mathit{effect}(\sigma') \succeq \mathbf{0}$. Since for any simple loop π in σ, $\mathit{effect}(\pi)(p) = 0$ for all $p \in P \setminus \omega(m_1)$, we have $\mathit{effect}(\sigma')(p) = \mathit{effect}(\sigma)(p) \geq 0$.

For any $p \in P_\omega$, we have $(\vec{B}\vec{y} + \Delta_{P_\omega}[\sigma''])(p) \geq \vec{d}(p) \geq 1$. Hence, $\vec{y}(\Delta_{P_\omega}[\pi]) \geq 1$ and $\Delta_{P_\omega}[\pi](p) = 1$ for some simple loop π or $\Delta_{P_\omega}[\sigma''](p) = 1$. From the definitions of $\Delta_{P_\omega}[\pi]$ and $\Delta_{P_\omega}[\sigma'']$, the only way this can happen is for some transition t in either some simple loop π or σ'' to have $\mathit{effect}(t) = \omega$. Hence, there is some transition t in σ' such that $\mathit{effect}(t)(p) = \omega$. Hence, $\mathit{effect}(\sigma')(p) = \omega$.

For any $p \in \omega(m_1) \setminus P_\omega$, we have $\mathit{effect}(\sigma')(p) = (\vec{B}\vec{y} + \Delta_{P_\omega}[\sigma''])(p) \geq \vec{d}(p) \geq 0$. Hence, $\mathit{effect}(\sigma')(p) \geq 0$. □

Definition 5.7. Let $c = 2d$. The functions $h_1, h_2, \ell : \mathbb{N} \to \mathbb{N}$ are as follows:

$$h_1(0) = 1 \qquad \ell(0) = (2R)^{c|P|^3} \qquad h_2(0) = R$$
$$h_1(i+1) = 2R\ell(i) \qquad \ell(i+1) = (h_1(i+1)2R)^{c|P|^3} \qquad h_2(i+1) = R\ell(i)$$

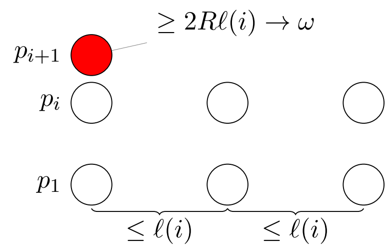

Figure 5. Intuition for the threshold functions

All the above functions are non-decreasing. Due to the selection of the constant c above, we have $(2xR)^{c|P|^3} \geq x^{|P|} + (2xR)^{d|P|^3}$ for all $x \in \mathbb{N}$.

The goal is to prove that if there is a self-covering execution, there is one whose length is at most $\ell(|P|)$. That proof uses the result of Lemma 5.6 and the definition of ℓ above reflects it. For the intuition behind the definition of h_1 and h_2, suppose that the proof of the length upper bound of $\ell(|P|)$ is by induction on $|P|$ and we have proved the result for $|P| = i$. For the case of $i + 1$, we want to decide the value beyond which it is safe to abstract by replacing numbers by ω. As shown in Fig. 5, suppose the initial prefix of a self-covering execution for i places is of length at most $\ell(i)$. Also suppose the pumping portion of the self-covering execution is of length at most $\ell(i)$. The total length is at most $2\ell(i)$. Since each transition can reduce at most R tokens from any place, it is enough to have $2R\ell(i)$ tokens in p_{i+1} to safely replace numbers by ω.

The following lemma shows that if some ω-marking can be reached in threshold semantics, a corresponding marking can be reached in the natural semantics where ω is replaced by a value large enough to solve the termination problem.

Lemma 5.8. For some ω-markings m_3 and m_4, suppose $m_3 \xrightarrow{\sigma}_{h_1} m_4$. Then there is a sequence σ' such that $[m_3]_{\omega\to h_1} \xrightarrow{\sigma'} m_4'$, $m_4' \succeq_{\omega(m_4)} [m_4]_{\omega\to h_2}$ and $|\sigma'| \leq h_1(\mathrm{nb}\overline{\omega}\,(m_3))^{|P|}$.

Proof:
Let σ' be obtained from σ by removing all transitions between any two identical ω-markings occurring in the run $m_3 \xrightarrow{\sigma}_{h_1} m_4$. The number of distinct ω-markings appearing in the run $m_3 \xrightarrow{\sigma'}_{h_1} m_4$ is an upper bound on $|\sigma'|$. Among the ω-markings in this run, m_3 has the maximum number of places not marked ω. Since h_1 is non-decreasing, we infer from the definition of threshold semantics (Definition 5.2) that $h_1(\mathrm{nb}\overline{\omega}\,(m_3))^{|P|}$ is an upper bound on the number of possible distinct ω-markings. Hence, $|\sigma'| \leq h_1(\mathrm{nb}\overline{\omega}\,(m_3))^{|P|}$. We will now prove that for any run $m_3 \xrightarrow{\sigma'}_{h_1} m_4$ where all intermediate ω-markings are distinct from one another, $[m_3]_{\omega\to h_1} \xrightarrow{\sigma'} m_4'$ and $m_4' \succeq_{\omega(m_4)} [m_4]_{\omega\to h_2}$. The proof is by induction on $\mathrm{nb}\omega\,(m_4) - \mathrm{nb}\omega\,(m_3)$ (the number of places where ω is newly introduced).

Base case $\mathrm{nb}\omega\,(m_4) - \mathrm{nb}\omega\,(m_3) = 0$: We have $|\sigma'| \leq h_1(\mathrm{nb}\overline{\omega}\,(m_3))^{|P|} \leq \ell(\mathrm{nb}\overline{\omega}\,(m_3))$. For any $p' \in \omega(m_3)$, we have by Definition 5.1 and Definition 5.7 that $[m_3]_{\omega\to h_1}(p') = h_1(\mathrm{nb}\overline{\omega}\,(m_3) + 1) = 2R\ell(\mathrm{nb}\overline{\omega}\,(m_3))$. We conclude from Proposition 5.3 that $[m_3]_{\omega\to h_1} \xrightarrow{\sigma'} m_4'$ and $m_4' \succeq_{\omega(m_4)} [m_4]_{\omega\to h_2}$.

Induction step: Let m_5 be the first ω-marking after m_3 such that $\mathrm{nb}\omega\,(m_5) > \mathrm{nb}\omega\,(m_3)$. Let $\sigma' = \sigma_1 t \sigma_2$ where $m_3 \xrightarrow{\sigma_1}_{h_1} m_6 \xrightarrow{t}_{h_1} m_5 \xrightarrow{\sigma_2}_{h_1} m_4$. Note that due to our choice of m_5, we have $\omega(m_6) = \omega(m_3)$. In any intermediate marking $m \neq m_3$ in the run $m_3 \xrightarrow{\sigma_1}_{h_1} m_6$, $m(p) < h_1(\text{nb}\overline{\omega}\,(m_3))$ for all $p \in P \setminus \omega(m_3)$ (otherwise, p would have been marked ω, contradicting $\omega(m_6) = \omega(m_3)$). Hence we have $|\sigma_1| \leq h_1(\text{nb}\overline{\omega}\,(m_3))^{|P|}$. For any $p' \in \omega(m_3)$, we have by Definition 5.1 and Definition 5.7 that $[m_3]_{\omega\to h_1}(p') = h_1(\text{nb}\overline{\omega}\,(m_3)+1) = 2R\ell(\text{nb}\overline{\omega}\,(m_3))$. We conclude from Proposition 5.3 that $[m_3]_{\omega\to h_1} \xrightarrow{\sigma_1} m'_6$ where $m'_6 \preceq_{\omega(m_6)} m_6$ and for all $p' \in \omega(m_6)$, $m'_6(p') \geq 2R\ell(\text{nb}\overline{\omega}\,(m_3)) - Rh_1(\text{nb}\overline{\omega}\,(m_3))^{|P|}$. Transition t is enabled at m'_6. Let $m'_6 \xrightarrow{t} m'_5$, where for any p such that $\mathit{effect}(t)(p) = \omega$, we chose $m'_5(p) \geq h_1(\text{nb}\overline{\omega}\,(m_5)+1)$. We now conclude that $m'_5 \succeq_{\omega(m_5)} [m_5]_{\omega\to h_1}$ due to the following reasons:

1. $p \in P \setminus \omega(m_5)$: we have $p \in P \setminus \omega(m_6)$.

$$\begin{aligned} m'_5(p) &= m'_6(p) + \mathit{effect}(t) && [\text{semantics of } \omega\text{PN}\ ] \\ &= m_6(p) + \mathit{effect}(t) && [m'_6 \preceq_{\omega(m_6)} m_6] \\ &= m_5(p) && [[m_6 + \mathit{effect}(t)]_{h_1\to\omega} = m_5, m_5(p) \neq \omega] \\ &= [m_5]_{\omega\to h_1}(p) \end{aligned}$$

2. $p \in \omega(m_5)$, $\mathit{effect}(t)(p) = \omega$: $m'_5(p) \geq h_1(\text{nb}\overline{\omega}\,(m_5)+1)$ by choice.

3. $p \in \omega(m_5)$, $\mathit{effect}(t)(p) \neq \omega$, $p \notin \omega(m_6)$: since $[m_6 + \mathit{effect}(t)]_{h_1\to\omega} = m_5$ and $m_5(p) = \omega$,

$$\begin{aligned} & m_6(p) + \mathit{effect}(t)(p) \geq h_1(\text{nb}\overline{\omega}\,(m_6)) \\ \Rightarrow\ & m_6(p) + \mathit{effect}(t)(p) \geq h_1(\text{nb}\overline{\omega}\,(m_5)+1) && [\text{nb}\omega\,(m_5) > \text{nb}\omega\,(m_6)] \\ \Rightarrow\ & m'_6(p) + \mathit{effect}(t)(p) \geq h_1(\text{nb}\overline{\omega}\,(m_5)+1) && [m'_6 \preceq_{\omega(m_6)} m_6] \\ \Rightarrow\ & m'_5(p) \geq h_1(\text{nb}\overline{\omega}\,(m_5)+1) && [\text{semantics of } \omega\text{PN}\ ] \end{aligned}$$

4. $p \in \omega(m_5)$, $\mathit{effect}(t)(p) \neq \omega$, $p \in \omega(m_6)$:

$$\begin{aligned} m'_5(p) &= m'_6(p) + \mathit{effect}(t)(p) && [\text{semantics of } \omega\text{PN}\ ] \\ &\geq m'_6(p) - R && [\text{Definition of } R] \\ &\geq 2R\ell(\text{nb}\overline{\omega}\,(m_3)) - Rh_1(\text{nb}\overline{\omega}\,(m_3))^{|P|} - R && [p \in \omega(m_6)] \\ &\geq R\ell(\text{nb}\overline{\omega}\,(m_3)) - Rh_1(\text{nb}\overline{\omega}\,(m_3))^{|P|} \\ &= R(h_1(\text{nb}\overline{\omega}\,(m_3))2R)^{c|P|^3} - Rh_1(\text{nb}\overline{\omega}\,(m_3))^{|P|} && [\text{Definition } 5.7] \\ &\geq h_1(\text{nb}\overline{\omega}\,(m_3)) \geq h_1(\text{nb}\overline{\omega}\,(m_5)+1) \end{aligned}$$

The last inequality follows since $\text{nb}\omega\,(m_5) > \text{nb}\omega\,(m_3)$.

Since $\text{nb}\omega\,(m_4) - \text{nb}\omega\,(m_5) < \text{nb}\omega\,(m_4) - \text{nb}\omega\,(m_3)$ and all intermediate ω-markings in the run $m_5 \xrightarrow{\sigma_2}_{h_1} m_4$ are distinct from one another, we have by induction hypothesis that $[m_5]_{\omega\to h_1} \xrightarrow{\sigma_2} m''_4$ and $m''_4 \succeq_{\omega(m_4)} [m_4]_{\omega\to h_2}$. Since $[m_3]_{\omega\to h_1} \xrightarrow{\sigma_1} m'_6 \xrightarrow{t} m'_5$, $m'_5 \succeq_{\omega(m_5)} [m_5]_{\omega\to h_1}$ and $[m_5]_{\omega\to h_1} \xrightarrow{\sigma_2} m''_4$, we infer by strong monotonicity that $[m_3]_{\omega\to h_1} \xrightarrow{\sigma_1 t \sigma_2} m'_4$ and $m'_4 \succeq_{\omega(m_4)} [m_4]_{\omega\to h_2}$. □

Lemma 5.9. If an ωPN $\mathcal{N}$ admits a self-covering execution, then it admits one whose sequence of transitions is of length at most $\ell(|P|)$.

Proof:
Suppose $\sigma = \sigma_1\sigma_2$ is the sequence of transitions in the given self-covering execution such that $m_0 \xrightarrow{\sigma_1} m_1 \xrightarrow{\sigma_2} m_2$ and $m_2 \succeq m_1$. A routine induction on the length of any sequence of transitions σ shows that if $m_3 \xrightarrow{\sigma} m_4$, we have $m_3 \xrightarrow{\sigma}_{h_1} m'_4$ with $m'_4 - m_3 \succeq m_4 - m_3$. Hence, we have $m_0 \xrightarrow{\sigma_1}_{h_1} m'_1 \xrightarrow{\sigma_2}_{h_1} m'_2$ with $m'_2 \succeq m'_1$. By monotonicity, we infer that for any $i \in \mathbb{N}^+$, $m'_i \xrightarrow{\sigma_2}_{h_1} m'_{i+1}$ with $m'_{i+1} \succeq m'_i$. Let $j \in \mathbb{N}^+$ be the first number such that $\omega(m'_j) = \omega(m'_{j+1})$. We have $m_0 \xrightarrow{\sigma_1\sigma_2^{j-1}}_{h_1} m'_j \xrightarrow{\sigma_2}_{h_1} m'_{j+1}$ and σ_2 is an ω-maximal h_1-PS enabled at m'_j.

By Lemma 5.6, there is a h_1-PS σ'_2 enabled at m'_j whose length is at most $(h_1(\text{nb}\overline{\omega}\left(m'_j\right))2R)^{d|P|^3}$. By Lemma 5.8, there is a sequence σ'_1 such that $m_0 \xrightarrow{\sigma'_1} m''_j$, $m''_j \succeq_{\omega(m'_j)} [m'_j]_{\omega\to h_2}$ and $|\sigma'_1| \leq (h_1(|P|))^{|P|}$. By Definition 5.7 and Definition 5.1, we infer that $m''_j(p) = R\ell(\text{nb}\overline{\omega}\left(m'_j\right)) = R(h_1(\text{nb}\overline{\omega}\left(m'_j\right))2R)^{c|P|^3} \geq R|\sigma'_2|$ for all $p \in \omega(m'_j)$. Hence, we infer from Proposition 5.3 that $m_0 \xrightarrow{\sigma'_1} m''_j \xrightarrow{\sigma'_2} m''_{j+1}$. Since σ'_2 is a h_1-PS, $\mathit{effect}(\sigma'_2) \succeq \mathbf{0}$, and so $m''_{j+1} \succeq m''_j$. Therefore, firing $\sigma'_1\sigma'_2$ at m_0 results in a self-covering execution. The length of $\sigma'_1\sigma'_2$ is at most $(h_1(|P|))^{|P|} + (h_1(\text{nb}\overline{\omega}\left(m'_j\right))2R)^{d|P|^3} \leq \ell(|P|)$. □

Lemma 5.10. Let $k = 3c$. Then $\ell(i) \leq (2R)^{k^{i+1}|P|^{3(i+1)}}$ for all $i \in \mathbb{N}$.

Proof:
By induction on i. For the base case $i = 0$, the result is obvious since by Definition 5.7, $\ell(0) = (2R)^{c|P|^3}$. The following proves the induction step.

$$
\begin{aligned}
\ell(i+1) &= (h_1(i+1)2R)^{c|P|^3} = (2R\ell(i)\cdot 2\cdot R)^{c|P|^3} && \text{[Definition 5.7]}\\
&= (4R^2)^{c|P|^3}(\ell(i))^{c|P|^3} = (2R)^{2c|P|^3}(\ell(i))^{c|P|^3} \\
&\leq (2R)^{2c|P|^3}((2R)^{k^{i+1}|P|^{3(i+1)}})^{c|P|^3} && \text{[Induction hypothesis]}\\
&= (2R)^{2c|P|^3}(2R)^{ck^{i+1}|P|^{3(i+2)}} \leq (2R)^{3ck^{i+1}|P|^{3(i+2)}}\\
&= (2R)^{k^{i+2}|P|^{3(i+2)}}
\end{aligned}
$$

□

Theorem 5.11. The termination problem for ωPN is EXPSPACE-c.

Proof:
Since ωPN generalise Petri nets, and since termination is EXPSPACE-c for Petri nets [19, 22], termination is EXPSPACE-hard for ωPN. Let us now show that termination for ωPN is in EXPSPACE. We have from Lemma 2.6 that an ωPN $\mathcal{N}$ does not terminate iff it admits a self-covering execution. From Lemma 5.9, it admits a self-covering execution iff it admits one whose sequence of transitions is of length at most $\ell(|P|)$. The following non-deterministic algorithm can guess and verify the existence of such a sequence. It works with ω-markings, storing ω in the respective places whenever an w-transition is fired. It takes as input an ωPN $\mathcal{N}$, with initial marking m_0. It outputs SUCCESS if a self-covering execution is guessed, FAIL otherwise.

```
1   counter := 0
2   m := m_0
3   if counter > ℓ(|P|) return FAIL
4   else
5       non-deterministically choose a transition t
6       if t is not enabled at m return FAIL
7       else
8               m := m + effect(t)
9               counter := counter + 1
10      non-deterministically go to line 3 or line 11
11  in m, replace ω by Rℓ(|P|)
12  m_1 := m
13  if counter > ℓ(|P|) return FAIL
14  else
15      non-deterministically choose a transition t
16      if t is not enabled at m_1 return FAIL
17      else
18              m_1 : = m_1 + effect(t)
19              counter := counter + 1
20              non-deterministically go to line 13 or line 21
21  if m_1 ⪰ m return SUCCESS else return FAIL
```

The above algorithm tries to guess a sequence of transitions $\sigma_1\sigma_2$ such that $m_0 \xrightarrow{\sigma_1} m \xrightarrow{\sigma_2} m_1$, guessing σ_1 in the loop between lines 3 and 10 and σ_2 in the loop between lines 13 and 20. If $\mathcal{N}$ admits a self-covering execution with sequence of transitions $\sigma_1\sigma_2$ such that $|\sigma_1\sigma_2| \leq \ell(|P|)$, then the execution of the above algorithm that guesses $\sigma_1\sigma_2$ will return SUCCESS. If all executions of $\mathcal{N}$ are finite, then all executions of the above algorithm will return FAIL.

The space required to store the variable "counter" in the above algorithm is at most $\log(\ell(|P|))$. The space required to store m and m_1 is at most $|P|(\|m_0\|_\infty + \log(R\ell(|P|)))$. Using the upper bound given by Lemma 5.10, we conclude that the memory space required by the above algorithm is $\mathcal{O}(|P|\log\|m_0\|_\infty + k^{|P|+1}|P|^{3|P|+4}\log R)$. This can be simplified to $\mathcal{O}(2^{c'|P|\log|P|}(\log R + \log\|m_0\|_\infty))$. Using the well known Savitch's theorem to determinise the above algorithm, we get an EXPSPACE upper bound for the termination problem in ωPN. □

6. Finding concrete counter-examples for coverability

As we have argued in the introduction, ω-arcs can be conveniently used in the setting of parametrised verification to model, in an abstract way, operations involving a parametric number of processes, modeled as tokens. Then, several safety properties of the system can be reduced to the coverability problem. As an example, consider the ωPN in Fig. 6, where two transitions create an unbounded number of processes (in place p). Place C.S. represents here a critical section and we would like to prove that mutual exclusion holds on this place, i.e. it is not possible to reach a marking m in

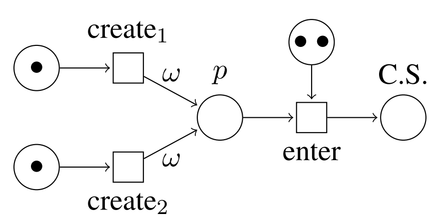

Figure 6. An example of parallel system where a mutex property is not enforced.

the set $\mathsf{Bad} = \{m \mid m(\text{C.S.}) \geq 2\}$ – a typical example of coverability property. Unfortunately, the property does not hold on this example, so it makes sense to ask what would be a *concrete* version of this protocol where the bug still occurs. By *concretisation* of an ωPN, we mean a variant of ωPN obtained by replacing the ω on the arcs by natural numbers. Symmetrically, it is also interesting to compute the largest set of tuples of values that we can use to replace the ω's and keep the system safe (i.e., avoid coverability). On our example, if we let $v = \langle v_1, v_2\rangle$ be a vector where v_1 and v_2 are respectively the labels to replace the ω's on the arcs from create_1 and create_2, the set of possible values for v yielding a PN that can still reach Bad is: $\{v \mid v \succeq (2,0) \vee v \succeq (1,1) \vee v \succeq (0,2)\}$. Observe that this set is upward-closed[8]. It is easy to see that it will always be the case for output arcs: if a PN can reach some upward-closed set U, then, increasing the weights on its output arcs yields a new PN that can still reach U. On the other, if a PN can reach some upward-closed set U, then, *decreasing* the weights on its *input* arcs yields too a new PN that can still reach U. Hence, from now on, we will always assume that we replace all ω on *input arcs* by 0 in any concretisation of an ωPN, which allows us to focus on the computation of the set of vectors that ensure coverability.

Let us formalise this problem. Let us fix an ωPN $\mathcal{N} = \langle P, T, m_0\rangle$. As in the previous parts of the paper, we slightly abuse notations, and denote by $O_{\mathcal{N}}$ (resp. $I_{\mathcal{N}}$) the functions $T \times P \mapsto (\mathbb{N} \cup \{\omega\})^{|P|}$ s.t. for all p, t: $I_{\mathcal{N}}(t,p) = I(t)(p)$ (resp. $O_{\mathcal{N}}(t,p) = O(t)(p)$) in $\mathcal{N}$. We denote by $A^{\omega}_{\mathcal{N}}$ the set of ω-output arcs of $\mathcal{N}$, i.e. $A^{\omega}_{\mathcal{N}} = \{(t,p) \mid O_{\mathcal{N}}(t,p) = \omega\}$. Let us assume an arbitrary but fixed order on the pairs (t,p) in $A^{\omega}_{\mathcal{N}}$, and let v be an $A^{\omega}_{\mathcal{N}}$-indexed vector of values in $\mathbb{N} \cup \{\omega\}$. We denote by $\mathcal{N}(v)$ the ωPN obtained by replacing each ω-output arc by an arc whose weight is given by v, and removing each ω-input arc. Formally, $\mathcal{N}(v) = \langle P, T', m_0\rangle$ is the ωPN s.t. for all $(t,p) \in T \times P$: $O_{\mathcal{N}(v)}(t,p) = v(t,p)$ if $O_{\mathcal{N}}(t,p) = \omega$, and $O_{\mathcal{N}(v)}(t,p) = O_{\mathcal{N}}(t,p)$ otherwise; $I_{\mathcal{N}(v)}(t,p) = 0$ if $I_{\mathcal{N}}(t,p) = \omega$, and $I_{\mathcal{N}(v)}(t,p) = I_{\mathcal{N}}(t,p)$ otherwise. When v is a vector of natural values only, we say that $\mathcal{N}(v)$ is a *concretisation* of $\mathcal{N}$.

Given an ωPN $\mathcal{N}$ and a marking m, the problem sketched above thus consists to compute the set:

$$\mathsf{CVec}(\mathcal{N}, m) = \{v \mid \mathcal{N}(v) \text{ is a concretisation of } \mathcal{N} \text{ and } m \in \downarrow(\mathsf{Reach}(\mathcal{N}(v)))\}$$

It is easy to see that $\mathsf{CVec}(\mathcal{N}, m)$ is upward-closed for the $\preceq$ ordering, i.e. $m_1 \in \mathsf{CVec}(\mathcal{N}, m)$ implies that $m_2 \in \mathsf{CVec}(\mathcal{N}, m)$ for all m_2 s.t. $m_2 \succeq m_1$. Since $\preceq$ is a wqo, there exists a finite representation of $\mathsf{CVec}(\mathcal{N}, m)$, i.e. a finite set of vectors S s.t. $\uparrow(S) = \mathsf{CVec}(\mathcal{N}, m)$, where $\uparrow(S)$ is the $\preceq$-upward closure of S: $\uparrow(S) = \{v \mid v \in \mathbb{N}^{|A^{\omega}_{\mathcal{N}}|} \wedge \exists v' \in S : v' \preceq v\}$. To compute a finite representation of $\mathsf{CVec}(\mathcal{N}, m)$, we rely on a general algorithm introduced by Valk and Jantzen [26]. They show that a finite representation of any upward-closed set U of k-tuples of natural numbers (for any k) can be computed if U has *the RES property*, i.e. if we can decide, for each $v \in (\mathbb{N} \cup \{\omega\})^K$ whether[9] $\gamma(v) \cap U = \emptyset$. Let us show that this property holds for $\mathsf{CVec}(\mathcal{N}, m)$:

Lemma 6.1. For all ωPN $\mathcal{N}$ and all markings m of $\mathcal{N}$, $\mathsf{CVec}(\mathcal{N}, m)$ has the RES property: for all vectors $v \in (\mathbb{N} \cup \{\omega\})^{|A^{\omega}_{\mathcal{N}}|}$, we can decide whether $\gamma(v) \cap \mathsf{CVec}(\mathcal{N}, m) = \emptyset$.

Proof:
In the case where v is a vector of natural numbers (i.e., v contains no ω), and since $\mathsf{CVec}(\mathcal{N}, m)$ is

[8] Conversely, the set of *safe* values for v (that guarantee to avoid Bad) is downward-closed.

[9] recall that $\gamma(v)$ is the concretisation of v, i.e., the set of all tuples of *naturals* that are smaller than v.

upward-closed, $\gamma(v) \cap \mathsf{CVec}(\mathcal{N}, m) = \emptyset$ iff $v \notin \mathsf{CVec}(\mathcal{N}, m)$. By definition of $\mathsf{CVec}(\mathcal{N}, m)$, this last statement holds iff $m \notin\downarrow(\mathsf{Reach}(\mathcal{N}(v)))$, an instance of the (decidable) coverability problem on PNs.

In the case where v contains at least one ω, we have to decide whether there exists $\overline{v} \in \gamma(v)$ s.t. $\mathcal{N}(\overline{v})$ can cover m. To solve this question, we consider the ωPN $\mathcal{N}(v)$. Observe that $\mathcal{N}(v)$ is not a PN, and thus not a concretisation of $\mathcal{N}$, since we have assumed that v contains at least one ω. On the one hand, assume $m \in\downarrow(\mathsf{Reach}(\mathcal{N}(v)))$, and let $\pi = m_0, t_1, m_1, \ldots t_n, m_n$ be an execution of $\mathcal{N}(v)$ s.t. $m_n \succeq m$. For each output arc $(t,p) \in A^{\omega}_{\mathcal{N}(v)}$, let $M_{(t,p)}$ be the maximal number of tokens produced by this arc along π. Then, we let $\overline{v}$ be the vector s.t. for all $(t,p) \in A^{\omega}_{\mathcal{N}}$: $\overline{v}(t,p) = v(t,p)$ if $v(t,p) \neq \omega$ and $\overline{v}(t,p) = M_{(t,p)}$ otherwise. By monotonicity of Petri nets, is easy to check that π's sequence of transitions $t_1 t_2 \cdots t_n$ is firable in $\mathcal{N}(\overline{v})$ and reaches a marking m'_n s.t. $m'_n \succeq m_n \succeq m$. Hence, $m \in\downarrow(\mathsf{Reach}(\mathcal{N}(\overline{v})))$. On the other hand, assume $m \notin\downarrow(\mathsf{Reach}(\mathcal{N}(v)))$. In this case, it is easy to see that, for all $\overline{v} \in \gamma(v)$: $m \notin\downarrow(\mathsf{Reach}\,\mathcal{N}(\overline{v}))$. Thus, we conclude that there is $\overline{v} \in \gamma(v)$ s.t. $\mathcal{N}(\overline{v})$ can cover m iff the ωPN $\mathcal{N}(v)$ can cover m. This last question is decidable as it is an instance of the coverability problem for ωPN. □

Hence, using the technique of Valk and Jantzen [26, Theorem 2.14], we conclude that:

Corollary 6.2. For all ωPN $\mathcal{N}$ and all marking m, we can compute a finite representation of $\mathsf{CVec}(\mathcal{N}, m)$.

7. Extensions with transfer or reset arcs

In this section, we consider two extensions of ωPN, namely: ωPN with *transfer arcs* (ωPN+T) and ωPN with *reset arcs* (ωPN+R). These extensions have been considered in the case of plain Petri nets: Petri nets with transfer arcs (PN+T) and Petri nets with reset arcs (PN+R) have been extensively studied in the literature [8, 1, 9, 25]. Intuitively, a *transfer arc* allows us to *transfer all the tokens* from a designated place p to a given place q, while a *reset arc consumes all tokens* from a designated place p. Those extensions have been applied in particular to model concurrent multi-threaded programs with inter-thread communication primitives such as *broadcasts* [10, 6]. It is thus natural to combine ω-arcs with resets or transfers, to obtain a rich model for concurrent multi-threaded programs.

Formally, an *extended* ωPN is a tuple $\langle P, T\rangle$, where P is a finite set of places and T is finite set of transitions. Each transition is a pair $t = (I, O)$ where $I : P \mapsto \mathbb{N} \cup \{\omega, \mathsf{T}, \mathsf{R}\}$; $O : P \mapsto \mathbb{N} \cup \{\omega, \mathsf{T}\}$; $|\{p \mid I(p) \in \{\mathsf{T}, \mathsf{R}\}\}| \leq 1$; $|\{p \mid O(p) \in \{\mathsf{T}\}\}| \leq 1$; there is p s.t. $I(p) = \mathsf{T}$ *iff* there is q s.t. $O(q) = \mathsf{T}$; and *if* there is p s.t. $I(p) = \mathsf{R}$, *then*, $O(p) \in \mathbb{N} \cup \{\omega\}$ for all p. A transition (I, O) s.t. $I(p) = \mathsf{T}$ (resp. $I(p) = \mathsf{R}$) for some p is called a *transfer* (*reset*). An ωPN with *transfer arcs* (resp. *with reset arcs*), ωPN+T (ωPN+R) for short, is an extended ωPN that contains no reset (transfer). An ωPN+T s.t. $I(t)(p) \neq \omega$ for all transitions t and places p is an ωOPN+T. The class ωIPN+T is defined symmetrically. An ωPN+T which is both an ωOPN+T *and* an ωIPN+T is a (plain) PN+T. The classes ωOPN+R, ωIPN+R and PN+R are defined accordingly.

Let $t = (I, O)$ be a transfer or a reset. t is *enabled* in a marking m iff for all p: $I(p) \notin \{\omega, \mathsf{T}, \mathsf{R}\}$ implies $m(p) \geq I(p)$. In this case firing t yields a marking $m' = m - m_I + m_O$ (denoted $m \xrightarrow{t} m'$) where for all p: $m_I(p) = m(p)$ if $I(p) \in \{\mathsf{T}, \mathsf{R}\}$; $0 \leq m_I(p) \leq m(p)$ if $I(p) = \omega$; $m_I(p) = I(p)$ if $I(p) \notin \{\mathsf{T}, \mathsf{R}, \omega\}$; $m_O(p) = m(p')$ if $O(p) = I(p') = \mathsf{T}$; $m_O(p) \geq 0$ if $O(p) = \omega$; and $m_O(p) = O(p)$ if $O(p) \notin \{\mathsf{T}, \omega\}$. The semantics of transitions that are neither transfers nor resets is as defined for ωPN.

Let us now investigate the status of the problems listed in Section 2, in the case of ωPN+T and ωPN+R. First, since ωPN+T (ωPN+R) extend PN+T (PN+R), the lower bounds for the latter carry on:

reachability and place-boundedness are undecidable [7] for ωPN+T and ωPN+R; boundedness is undecidable for ωPN+R [9]; and coverability is Ackerman-hard for ωPN+T and ωPN+R [25]. On the other hand, the construction given in Section 4 can be adapted to turn an ωPN+T (resp. ωPN+R) $\mathcal{N}$ into a PN+T (PN+R) $\mathcal{N}'$ satisfying Lemma 4.1 (i.e., projecting $\mathsf{Reach}(\mathcal{N}', m_0)$ on the set of places of $\mathcal{N}$ yields $\mathsf{Reach}(\mathcal{N}, m_0)$). Hence, boundedness for ωPN+T [9], and coverability for both ωPN+T and ωPN+R are decidable [1]. As far as *termination* is concerned, it is decidable [8] and Ackerman-hard [25] for PN+R and PN+T. Termination, however, becomes undecidable for ωOPN+R or ωOPN+T:

Theorem 7.1. Termination is undecidable for ωOPN+T and ωOPN+R with one ω-output-arc.

Proof:
We first prove undecidability for ωOPN+T. The proof is by reduction from the *parametrised termination problem* for *Broadcast protocols* (BP) [10]. It is well-known that PN+T generalise broadcast protocols, hence the following *parametrised termination problem for PN+T* is *undecidable*: 'given a PN+T $\langle P, T\rangle$ and an ω-marking $\overline{m}_0$, does $\langle P, T, m_0\rangle$ terminate *for all* $m_0 \in\downarrow(\overline{m}_0)$?' From a PN+T $\mathcal{N} = \langle P, T\rangle$ and an ω-marking $\overline{m}_0$, we build the ωOPN+T (with only one ω-output-arc) $\mathcal{N}' = \langle P', T', m_0'\rangle$ where $P' = P \uplus \{p_{init}\}$, $T' = T \uplus \{(I, O)\}$, $I = \{p_{init}\}$, $O = \{\omega \otimes p \mid \overline{m}_0(p) = \omega\}$, and $m_0' = \{\overline{m}_0 \otimes p \mid \overline{m}_0(p) \neq \omega\}$. Clearly, $\mathcal{N}'$ terminates iff $\langle P, T, m_0\rangle$ terminates for all $m_0 \in\downarrow(\overline{m}_0)$. Hence, termination for ωOPN+T is *undecidable* too. Finally, we transform any ωOPN+R $\mathcal{N} = \langle P, T, m_0\rangle$ into an ωOPN+T $\mathcal{N}' = \langle P \uplus \{p_{trash}\}, T', m_0\rangle$, s.t. $t' \in T'$ iff either (i) $t' \in T$ and t' is not a reset; or (ii) there is a reset $t \in T$ and a place $p \in P$ s.t. $I(t)(p) = \mathsf{R}$, $I(t')(p) = \mathsf{T}$, $O(t')(p_{trash}) = \mathsf{T}$, for all $p' \neq p$: $I(t')(p') = I(t)(p')$ and for all $p'' \neq p_{trash}$: $O(t')(p'') = O(t)(p'')$. The construction replaces each reset of place p in $\mathcal{N}$ by a transfer from p to a fresh place p_{trash} from which no transition consume, in $\mathcal{N}'$. Since $\mathcal{N}'$ terminates iff $\mathcal{N}$ terminates, termination is undecidable for ωPN+R too. □

However, the construction of Section 4 can be applied to ωIPN+T and ωIPN+R to yield a corresponding PN+T (resp. PN+R) that preserves termination. Hence, termination is decidable and Ackerman-hard for those models. This justifies the results on ωPN+T and ωPN+R given in Table 1.

References

[1] P. A. Abdulla, K. Cerans, B. Jonsson, and Y.-K. Tsay. General Decidability Theorems for Infinite-state Systems. In *Proceedings of LICS'96*, pages 313–321. IEEE Computer Society Press, 1996.

[2] M. Blondin, A. Finkel, and P. McKenzie. Handling infinitely branching WSTS. In *Proceedings of ICALP'14 – Part II*, LNCS. Springer, 2014. To appear.

[3] I. Borosh and L. Treybig. Bounds on positive integral solutions of linear diophantine equations. *Proceedings of the American Mathematical Society*, 55(2):299–304, March 1976.

[4] T. Brázdil, P. Jančar, and A. Kučera. Reachability games on extended vector addition systems with states. In *Proceedings of ICALP'10*, volume 6199 of *LNCS*, pages 478–489. Springer, 2010.

[5] G. Delzanno. Constraint-based verification of parameterized cache coherence protocols. *Formal Methods in System Design*, 23(3):257–301, 2003.

[6] G. Delzanno, J-F. Raskin, and L. Van Begin. Towards the Automated Verification of Multithreaded Java Programs. In *Proceedings of TACAS'02*, volume 2280 of *LNCS*, pages 173–187. Springer, 2002.

[7] C. Dufourd. *Réseaux de Petri avec reset/transfert : Décidabilité et indécidabilité*. PhD thesis, ENS de Cachan, 1998.

[8] C. Dufourd, A. Finkel, and Ph. Schnoebelen. Reset Nets Between Decidability and Undecidability. In *Proceedings of ICALP'98*, volume 1443 of *LNCS*, pages 103–115. Springer, 1998.

[9] C. Dufourd, P. Jančar, and Ph. Schnoebelen. Boundedness of reset p/t nets. In *Proceedings of ICALP'99*, volume 1644 of *LNCS*, pages 301–310. Springer, 1999.

[10] J. Esparza, A. Finkel, and R. Mayr. On the Verification of Broadcast Protocols. In *Proceedings of LICS'99*, pages 352–359. IEEE, 1999.

[11] A. Finkel, P. McKenzie, and C. Picaronny. A well-structured framework for analysing petri net extensions. *Inf. Comput.*, 195(1-2):1–29, November 2004.

[12] A. Finkel and P. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1-2):63–92, 2001.

[13] G. Geeraerts, A. Heußner, and J.F. Raskin. Queue-dispatch asynchronous systems. In *Proceedings of ACSD 2013*, pages 150–159. IEEE, 2013.

[14] G. Geeraerts, J.-F. Raskin, and L. Van Begin. Expand, enlarge and check: New algorithms for the coverability problem of WSTS. *J. Comput. Syst. Sci.*, 72(1):180–203, 2006.

[15] G. Geeraerts, J.-F. Raskin, and L. Van Begin. On the efficient computation of the minimal coverability set of Petri nets. *Int. J. Found. Comput. Sci.*, 21(2):135–165, 2010.

[16] S. German and P. Sistla. Reasoning about systems with many processes. *Journal of the ACM*, 39(3):675–735, 1992.

[17] P. Jančar. Decidability of weak fairness in Petri nets. In *Proceedings of STACS'89*, volume 349 of *LNCS*, pages 446–457. Springer, 1989.

[18] R. M. Karp and R. E. Miller. Parallel Program Schemata. *J. Comput.r and Syst. Sci.*, 3:147–195, 1969.

[19] R. Lipton. The reachability problem requires exponential space. Technical Report 63, Yale, 1976.

[20] E.W. Mayr. An algorithm for the general Petri net reachability problem. *SIAM Journal of Computing*, 3(13):441–460, 1984.

[21] C. A. Petri. *Kommunikation mit automaten*. PhD thesis, Institut fur Instrumentelle Mathematik, Bonn, 1962.

[22] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6:223–231, 1978.

[23] W. Reisig. *Petri Nets: An Introduction*. Springer-Verlag, 1985.

[24] P.-A. Reynier and F. Servais. Minimal coverability set for Petri nets: Karp and Miller algorithm with pruning. In *Proceedings of Petri Nets 2011*, volume 6709 of *LNCS*, pages 69–88. Springer, 2011.

[25] P. Schnoebelen. Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets. In *Proceedings of MFCS'10*, volume 6281 of *LNCS*, pages 616–628. Springer, 2010.

[26] R. Valk and M. Jantzen. The residue of vector sets with applications to decidability problems in Petri nets. *Acta Informatica*, 21(6):643–674, 1985.

[27] A. Valmari and H. Hansen. Old and new algorithms for minimal coverability sets. *Fundam. Inform.*, 131(1):1–25, 2014.