Occam's Razor applied to the Petri net coverability problem

Thomas Geffroy, Jérôme Leroux, Grégoire Sutre

Occam's Razor Applied to the Petri Net Coverability Problem

Thomas Geffroy, Jérôme Leroux, Grégoire Sutre

Univ. Bordeaux & CNRS, LaBRI, UMR 5800, Talence, France

Abstract

The verification of safety properties for concurrent systems often reduces to the *coverability* problem for Petri nets. This problem was shown to be ExpSpace-complete forty years ago. Driven by the concurrency revolution, it has regained a lot of interest over the last decade. In this paper, we propose a generic and simple approach to solve this problem. Our method is inspired from the recent approach of Blondin, Finkel, Haase and Haddad presented at TACAS in 2016. Basically, we combine forward invariant generation techniques for Petri nets with backward reachability for well-structured transition systems. An experimental evaluation demonstrates the efficiency of our approach.

Keywords: Petri net, coverability problem, model-checking, invariant, pre-processing

1. Introduction

Context. The analysis of concurrent systems with unboundedly many processes classically uses the so-called *counter abstraction* [1]. The main idea is to forget about the identity of each process, so as to make processes indistinguishable. Assuming that each process is modeled by a finite-state automaton, it is then enough to count, for each state q, how many processes are in state q. The resulting model is a Petri net, with no a priori bound on the number of tokens. The verification of a safety property on the original concurrent system (e.g., mutual exclusion) translates into a *coverability* question on the Petri net: Is it possible to reach a marking that is component-wise larger than a given marking?

Related work. Karp and Miller [2] proved in 1969 that coverability is decidable (but their algorithm is not primitive recursive), Lipton showed that it requires at least exponential space [3], and Rackoff showed that it only requires exponential space [4]. Despite these somewhat negative results, and driven by the concurrency revolution, the coverability problem has regained a lot of interest over the last decade. Recent efficient approaches include target set widening [5] and structural analysis mixed with SMT solving [6, 7]. We believe that the time is ripe to

experiment with new ideas and prototypes for coverability, and to apply them to real-world concurrent systems.

Our work builds notably on [7], which proposes a new approach to the coverability problem and its implementation. The approach of [7] is conceptually simple and exploits recent advances in the theory of Petri nets as well as the power of modern SMT-solvers. In a nutshell, they leverage recent results on coverability in continuous Petri nets [8] to over-approximate coverability under the standard semantics: any marking that is not coverable in a continuous Petri net is also not coverable under the standard semantics. This observation is then exploited inside a backward-coverability framework [9].

Our contribution. We present a generic backward coverability algorithm that relies on an over-approximation of the set of coverable markings — this will be in practice a downward-closed invariant — to prune the exploration of the state space. Our algorithm is in fact a family of algorithms parametrized by over-approximations. It generalizes the `QCover` algorithm presented in [7]. Whereas `QCover` is based on invariants obtained from recent results on continuous Petri nets [8], we present two classical methods for computing over-approximations that can be used in our pruning exploration: the *state equation* for Petri nets [10], and data-flow *sign analysis* [11]. We also show how to leverage sign analysis to simplify Petri net coverability queries.

We have implemented our algorithm as a variant of `QCover` [12] that we call `ICover` [13]. On the 143 Petri net coverability questions that `QCover` solved, the tool `QCover` took 9729 seconds, while `ICover` used only 5213 seconds.

Additionally, we have implemented a pre-processing algorithm that reduces the number of places and transitions of a Petri net without modifying coverability questions. It follows that it can be used safely with any coverability tool. This pre-processing is based on the computation of so-called *omega places*. These are places that can receive arbitrarily larger number of tokens. With this improvement, `ICover` and `QCover` terminate on additional problems. Thanks to all our modifications, the total time to solve the instances is divided by a factor of five.

Outline. Sections 2 and 3 recall the Petri net coverability problem and the classical backward reachability approach to solve it. Section 4 presents our backward coverability algorithm with pruning based on over-approximations of the coverability set. In Sections 5 and 6, we recall two classical methods for computing such over-approximations, namely the state (in-)equation and sign analysis. Section 7 is dedicated to the experimental evaluation of the tool `ICover`. In Section 8, we provide mathematical foundations for explaining our empirical good results based on the notion of limit-reachability in continuous Petri nets [14]. Pre-processing based on omega places is presented in Section 9.

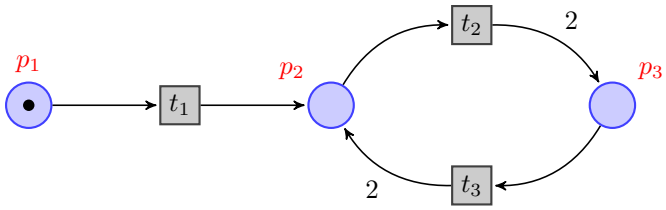

Figure 1: Simple Petri net example

2. The Coverability Problem for Petri nets

A Petri net is a tuple $\mathcal{N} = (P, T, F, Init)$ comprising a finite set P of *places*, a finite set T of *transitions* disjoint from P, a *flow* function F from $(P \times T) \cup (T \times P)$ to $\mathbb{N}$, and a set $Init \subseteq \mathbb{N}^P$ of *initial* markings. It is understood that $\mathbb{N}^P$ denotes the set of total maps from P to $\mathbb{N}$. Elements of $\mathbb{N}^P$ are called *markings*. Intuitively, a marking specifies how many *tokens* are in each place of the net. Tokens are consumed and produced through the firing of transitions. A transition $t \in T$ may fire only if it is enabled, meaning that each place p contains at least $F(p,t)$ tokens. Firing an enabled transition t modifies the contents of each place p by first removing $F(p,t)$ tokens and then adding $F(t,p)$ tokens. To clarify this intuitive description of the Petri net semantics, we introduce, for each transition $t \in T$, the t-step binary relation $\xrightarrow{t}$ over $\mathbb{N}^P$, defined by

$$m \xrightarrow{t} m' \;\Leftrightarrow\; \forall p \in P : m(p) \geq F(p,t) \wedge m'(p) = m(p) - F(p,t) + F(t,p)$$

The one-step binary relation $\rightarrow$ is the union of these t-step relations. Formally, $m \rightarrow m' \Leftrightarrow \exists t \in T : m \xrightarrow{t} m'$. The many-step binary relation $\xrightarrow{*}$ is the reflexive-transitive closure of $\rightarrow$.

Example 2.1. Figure 1 depicts a simple Petri net $\mathcal{N} = (P, T, F, Init)$ with places $P = \{p_1, p_2, p_3\}$, transitions $T = \{t_1, t_2, t_3\}$ and flow function F such that $F(p_1, t_1) = 1$, $F(p_2, t_2) = 1$, $F(p_3, t_3) = 1$, $F(t_1, p_2) = 1$, $F(t_2, p_3) = 2$, $F(t_3, p_2) = 2$, and $F(p,t) = F(t,p) = 0$ for all other cases. The set of initial markings is $Init = \{(1,0,0)\}$. The sequence of transitions $t_1 t_2 t_3$ may fire from the initial marking. Indeed, $(1,0,0) \xrightarrow{t_1} (0,1,0) \xrightarrow{t_2} (0,0,2) \xrightarrow{t_3} (0,2,1)$. □

One of the most fundamental verification questions on Petri nets is coverability. In its simplest form, the coverability problem asks whether it is possible, by firing a sequence of transitions, to put a token in a given place. In essence, the coverability problem for Petri nets corresponds to the control-state reachability problem for other models of computation, such as counter machines, which are equipped with control states. The formal definition of coverability relies on a partial order over markings, defined hereafter.

Let $\leq$ denote the usual total order on $\mathbb{N}$. We extend $\leq$ over $\mathbb{N}^P$ component-wise, by $m \leq m' \Leftrightarrow \forall p \in P : m(p) \leq m'(p)$. This extension is a partial order over $\mathbb{N}^P$. Given two markings m and m' in $\mathbb{N}^P$, we say that m *covers* m' when

$m \geq m'$. The *coverability problem* asks, given a Petri net $\mathcal{N} = (P, T, F, Init)$ and a *target* marking $m_{final} \in \mathbb{N}^P$, whether there exist a marking $m \in \mathbb{N}^P$ and an initial marking $m_{init} \in Init$ such that $m_{init} \xrightarrow{*} m$ and $m \geq m_{final}$. The main goal of this paper is to provide a simple, yet efficient procedure for solving this problem. Our method is inspired from the recent approach of [7]. Basically, we combine forward invariant generation techniques for Petri nets with backward reachability for well-structured transition systems [9, 15]. Before delving into the details, we need some additional notations.

For a transition $t \in T$ and a set $S \subseteq \mathbb{N}^P$ of markings, we let $pre^t_{\mathcal{N}}(S)$ denote the predecessors of S via the transition t. Similarly, $pre_{\mathcal{N}}(S)$ and $pre^*_{\mathcal{N}}(S)$ denote the one-step and many-step predecessors of S, respectively. Formally, the functions $pre^t_{\mathcal{N}}$, $pre_{\mathcal{N}}$ and $pre^*_{\mathcal{N}}$ from $2^{\mathbb{N}^P}$ to $2^{\mathbb{N}^P}$ are defined by

$$\begin{array}{rcl} pre^t_{\mathcal{N}}(S) & = & \{m \in \mathbb{N}^P \mid \exists m' \in S : m \xrightarrow{t} m'\} \\ pre_{\mathcal{N}}(S) & = & \{m \in \mathbb{N}^P \mid \exists m' \in S : m \rightarrow m'\} \\ pre^*_{\mathcal{N}}(S) & = & \{m \in \mathbb{N}^P \mid \exists m' \in S : m \xrightarrow{*} m'\} \end{array}$$

Given a subset $S \subseteq \mathbb{N}^P$ of markings, we let $\uparrow S$ and $\downarrow S$ denote its *upward closure* and *downward closure*, respectively. These are defined by

$$\begin{array}{rcl} \uparrow S & = & \{u \in \mathbb{N}^P \mid \exists m \in S : u \geq m\} \\ \downarrow S & = & \{d \in \mathbb{N}^P \mid \exists m \in S : d \leq m\} \end{array}$$

A subset $S \subseteq \mathbb{N}^P$ is called *upward-closed* when $S = \uparrow S$, and it is called *downward-closed* when $S = \downarrow S$.

Notation 2.2. For the remainder of the paper, to avoid clutter, we will simply write $\uparrow m$ in place of $\uparrow\{m\}$ for singletons, when this causes no confusion. □

Recall that the coverability problem asks whether there exists $m_{init} \in Init$ and a marking $m \in \mathbb{N}^P$ such that $m_{init} \xrightarrow{*} m$ and $m \geq m_{final}$. This problem is equivalently phrased as the question whether $Init$ intersects $pre^*_{\mathcal{N}}(\uparrow m_{final})$. This formulation can be seen as a backward analysis question. We may also phrase the coverability problem in terms of a forward analysis question, using the notion of coverability set.

Given a Petri net $\mathcal{N} = (P, T, F, Init)$, the *coverability set* of $\mathcal{N}$ is the set $Cov_{\mathcal{N}} = \downarrow\{m \in \mathbb{N}^P \mid \exists m_{init} \in Init, m_{init} \xrightarrow{*} m\}$. It is readily seen that the coverability problem is equivalent to the question whether m_{final} belongs to $Cov_{\mathcal{N}}$. We are now equipped with the necessary notions to present our mixed forward/backward approach for the coverability problem.

3. Classical Backward Coverability Analysis

The classical backward reachability approach for the coverability problem [9, 15] consists in computing a growing sequence $U_0 \subseteq U_1 \subseteq \cdots$ of upward-closed subsets of $\mathbb{N}^P$, that converges to $pre^*_{\mathcal{N}}(\uparrow m_{final})$. The sequence (U_k) is defined

by $U_k = pre_{\mathcal{N}}^{\leq k}(\uparrow m_{final})$, where the notation $pre_{\mathcal{N}}^{\leq k}$ stands for the predecessors in at most k steps. Formally, given a set $S \subseteq \mathbb{N}^P$ of markings, the function $pre_{\mathcal{N}}^{\leq k}$ is defined by

$$pre_{\mathcal{N}}^{\leq k}(S) \;=\; S \cup pre_{\mathcal{N}}(S) \cup \cdots \cup pre_{\mathcal{N}}^{k}(S)$$

Each set U_k is upward-closed because $pre_{\mathcal{N}}$ preserves upward closure. The convergence of the sequence (U_k) is guaranteed by the following lemma.

Lemma 3.1. *Every growing sequence of upward-closed subsets of $\mathbb{N}^P$ is ultimately stationary.*

Proof. By contradiction, assume that there exists an infinite strictly growing sequence $U_0 \subset U_1 \subset \cdots$ of upward-closed subsets of $\mathbb{N}^P$. For each k, there exists $m_k \in U_{k+1} \setminus U_k$. As the partial order $\leq$ on $\mathbb{N}^P$ is a well-quasi-order by Dickson's Lemma, there exists $h < k$ such that $m_h \leq m_k$. It follows that $m_k \in U_{h+1}$ since U_h is upward-closed, which contradicts $m_k \notin U_k$. □

Of course, we cannot directly compute the sets U_k since they may be infinite (in fact, they are either empty or infinite). Instead, we can compute finite sets $A_k \subseteq \mathbb{N}^P$ such that $U_k = \uparrow A_k$. The existence of such finite sets is guaranteed by the following lemma. A *basis* of an upward-closed subset $U \subseteq \mathbb{N}^P$ is any set $A \subseteq \mathbb{N}^P$ such that $U = \uparrow A$. Recall that a *minimal* element of a subset $S \subseteq \mathbb{N}^P$ is any $m \in S$ such that $u \leq m \Rightarrow u = m$ for every $u \in S$.

Lemma 3.2. *For every subset $S \subseteq \mathbb{N}^P$, the set $\operatorname{Min} S$ of its minimal elements is finite and satisfies $\uparrow S = \uparrow \operatorname{Min} S$.*

Proof. The partial order $\leq$ on $\mathbb{N}^P$ is a well-quasi-order by Dickson's Lemma. Therefore, the set $\operatorname{Min} S$ of minimal elements of S is necessarily finite. Moreover, $S \subseteq \uparrow \operatorname{Min} S$ since $\leq$ is well-founded. It follows that $\uparrow S = \uparrow \operatorname{Min} S$. □

Corollary 3.3. *Every upward-closed subset $U \subseteq \mathbb{N}^P$ admits a finite basis.*

Recall that we want to compute finite bases A_k of the upward-closed sets U_k. To see how A_{k+1} can be computed from A_k, we introduce, for each transition $t \in T$, the *covering predecessor* function $cpre_{\mathcal{N}}^t : \mathbb{N}^P \to \mathbb{N}^P$ defined by

$$cpre_{\mathcal{N}}^t(m)(p) \;=\; F(p,t) + \max(0, m(p) - F(t,p))$$

Informally, $cpre_{\mathcal{N}}^t(m)$ is the least marking that can cover m in one step by firing the transition t. This property will be formally stated in Lemma 3.4. The function $cpre_{\mathcal{N}}^t$ is extended to sets of markings by $cpre_{\mathcal{N}}^t(S) = \{cpre_{\mathcal{N}}^t(m) \mid m \in S\}$, and is further extended to all transitions by $cpre_{\mathcal{N}}(S) = \{cpre_{\mathcal{N}}^t(m) \mid t \in T, m \in S\}$.

Lemma 3.4. *It holds that $pre_{\mathcal{N}}^t(\uparrow m) = \uparrow cpre_{\mathcal{N}}^t(m)$ for every marking $m \in \mathbb{N}^P$.*

Proof. Let $u \in pre_{\mathcal{N}}^t(\uparrow m)$. There exists $v \geq m$ such that $u \xrightarrow{t} v$. Consider a place $p \in P$. It holds that $u(p) \geq F(p,t)$ and $v(p) = u(p) - F(p,t) + F(t,p)$ since $u \xrightarrow{t} v$. We consider two cases.

- If $m(p) \leq F(p,t)$ then $cpre^t_{\mathcal{N}}(m)(p) = F(p,t) \leq u(p)$.
- If $m(p) \geq F(p,t)$ then $cpre^t_{\mathcal{N}}(m)(p) = F(p,t) + m(p) - F(t,p)$. Since $v \geq m$, we get that $cpre^t_{\mathcal{N}}(m)(p) \leq F(p,t) + v(p) - F(t,p) = u(p)$.

In both cases, we obtain that $cpre^t_{\mathcal{N}}(m)(p) \leq u(p)$. We have thus shown that $u \in \uparrow cpre^t_{\mathcal{N}}(m)$.

Conversely, let $u \in \uparrow cpre^t_{\mathcal{N}}(m)$. This means that $u(p) \geq cpre^t_{\mathcal{N}}(m)(p)$ for every place $p \in P$. Therefore, $u(p) \geq F(p,t)$ and $u(p) \geq F(p,t) + m(p) - F(t,p)$. It follows that $u \xrightarrow{t} v$ for the marking $v \geq m$ defined by $v(p) = u(p) - F(p,t) + F(t,p)$. We have thus shown that $u \in pre^t_{\mathcal{N}}(\uparrow m)$. □

Corollary 3.5. *It holds that* $pre_{\mathcal{N}}(\uparrow S) = \uparrow cpre_{\mathcal{N}}(S)$ *for every subset* $S \subseteq \mathbb{N}^P$.

We now have the necessary ingredients to define a computable sequence (A_k) of finite subsets of $\mathbb{N}^P$ such that $U_k = \uparrow A_k$ for all k. The sequence (A_k) is defined as follows:

$$\begin{array}{rcl} A_0 & = & \{m_{final}\} \\ A_{k+1} & = & \mathrm{Min}\left(cpre_{\mathcal{N}}(A_k) \cup A_k\right) \end{array}$$

The classical backward coverability algorithm [9, 15] consists in computing $A_0, A_1, \ldots$ until a fixpoint is reached, i.e., $A_{k+1} = A_k$. The resulting set A_k is a basis of $pre^*_{\mathcal{N}}(\uparrow m_{final})$, and so the coverability question reduces to the check whether $Init$ intersects $\uparrow A_k$.

4. Backward Coverability Analysis with Pruning

This section presents our method to solve the coverability problem for Petri nets. Our approach can be seen as a refinement of the classical backward coverability analysis presented in Section 3. We start with the mathematical foundations of our approach, with no regard for implementability. We will discuss implementation issues at the end of this section.

The sequence (A_k) of the classical backward coverability analysis is purely backward. It does not exploit any information that one may already have on the Petri net under analysis. Here, we modify the sequence (A_k) in order to leverage an a priori known over-approximation of the coverability set. In practice, this means that we narrow the backward reachability search by pruning some markings that are known to be not coverable.

Assume that we are given a set $I \subseteq \mathbb{N}^P$ of "potentially useful" markings. We modify the sequence (A_k) into the sequence (B^I_k) defined by:

$$\begin{array}{rcl} B^I_0 & = & \{m_{final}\} \cap I \\ B^I_{k+1} & = & \mathrm{Min}\left((cpre_{\mathcal{N}}(B^I_k) \cap I) \cup B^I_k\right) \end{array}$$

On the contrary to the classical sequence (A_k), B^I_{k+1} does not consider all one-step predecessors of B^I_k, but discards those that are not in I. Note that by taking $I = \mathbb{N}^P$, we obtain the same sequence as before, i.e., $B^{\mathbb{N}^P}_k = A_k$ for all k.

Obviously, the usefulness (B_k^I) crucially depends on the choice of I. For instance, if I does not contain m_{final} then $B_k^I = \emptyset$ regardless of whether m_{final} belongs to $Cov_{\mathcal{N}}$ or not. Intuitively, for the sequence (B_k^I) to be useful, we need it to preserve coverability in the following sense: if m_{final} belongs to $Cov_{\mathcal{N}}$ then $Init$ intersects $\uparrow B_k^I$ for some k. A natural requirement for I would be that it contains the reachability set of $\mathcal{N}$, that is the set $\{m \in \mathbb{N}^P \mid \exists m_{init} \in Init : m_{init} \xrightarrow{*} m\}$. But, as shown in Example 4.1 below, this requirement does not ensure that (B_k^I) preserves coverability. Instead of the reachability set, we require in Lemma 4.2 that I contains the coverability set.

Example 4.1. Consider the very simple Petri net with only one place p, one transition t such that $F(p,t) = 1$ and $F(t,p) = 3$, and one initial marking $m_{init} = p$. Since $p \xrightarrow{t} 3p \xrightarrow{t} 5p$, we get that $m_{final} = 4p$ belongs to $Cov_{\mathcal{N}}$. Now consider the set $I = \{p, 3p, 5p, \ldots\}$. It is readily seen that I is the reachability set of $\mathcal{N}$. But since $m_{final} \notin I$, we get that $B_k^I = \emptyset$ for all k. So (B_k^I) does not preserve coverability. □

Lemma 4.2. *If $Cov_{\mathcal{N}} \subseteq I$ then it holds that*

$$pre_{\mathcal{N}}^{\leq k}(\uparrow m_{final}) \cap Cov_{\mathcal{N}} \;\subseteq\; \uparrow B_k^I \;\subseteq\; pre_{\mathcal{N}}^{\leq k}(\uparrow m_{final})$$

for all $k \in \mathbb{N}$.

Proof. Both inclusions are proved by induction on k. We first prove the second inclusion. The basis holds because $\uparrow B_0^I \subseteq \uparrow m_{final} = pre_{\mathcal{N}}^{\leq 0}(\uparrow m_{final})$. For the induction step, assume that $\uparrow B_k^I \subseteq pre_{\mathcal{N}}^{\leq k}(\uparrow m_{final})$. We get that

$$\begin{array}{lll}
\uparrow B_{k+1}^I &= \uparrow(cpre_{\mathcal{N}}(B_k^I) \cap I) \;\cup\; \uparrow B_k^I & \text{[Lemma 3.2]} \\
&\subseteq \uparrow cpre_{\mathcal{N}}(B_k^I) \;\cup\; \uparrow B_k^I & \\
&= pre_{\mathcal{N}}(\uparrow B_k^I) \;\cup\; \uparrow B_k^I & \text{[Corollary 3.5]} \\
&\subseteq pre_{\mathcal{N}}^{\leq k+1}(\uparrow m_{final}) & [\uparrow B_k^I \subseteq pre_{\mathcal{N}}^{\leq k}(\uparrow m_{final})]
\end{array}$$

We now prove the first inclusion. We consider two cases for the basis. If $m_{final} \in I$ then $(pre_{\mathcal{N}}^{\leq 0}(\uparrow m_{final}) \cap Cov_{\mathcal{N}}) \subseteq \uparrow m_{final} = \uparrow B_0^I$. If $m_{final} \notin I$ then $m_{final} \notin Cov_{\mathcal{N}}$. Moreover, since $Cov_{\mathcal{N}}$ is downward-closed, $\uparrow m_{final} \cap Cov_{\mathcal{N}}$ is empty. Therefore, $pre_{\mathcal{N}}^{\leq 0}(\uparrow m_{final}) \cap Cov_{\mathcal{N}}$ is empty, and so it is contained in $\uparrow B_0^I$. For the induction step, assume that $(pre_{\mathcal{N}}^{\leq k}(\uparrow m_{final}) \cap Cov_{\mathcal{N}}) \subseteq \uparrow B_k^I$. The definition of (B_k^I) entails, with Lemma 3.2, that $\uparrow B_k^I \subseteq \uparrow B_{k+1}^I$. So we only need to prove that $(pre_{\mathcal{N}}^{k+1}(\uparrow m_{final}) \cap Cov_{\mathcal{N}}) \subseteq \uparrow B_{k+1}^I$. Let us first observe that

$$pre_{\mathcal{N}}(S) \cap Cov_{\mathcal{N}} \;\subseteq\; pre_{\mathcal{N}}(S \cap Cov_{\mathcal{N}})$$

for every subset $S \subseteq \mathbb{N}^P$. This observation follows from the fact that $m \in Cov_{\mathcal{N}}$ and $m \to m'$ entails that $m' \in Cov_{\mathcal{N}}$. It follows that

$$pre_{\mathcal{N}}(\uparrow B_k^I) \;\supseteq\; pre_{\mathcal{N}}^{k+1}(\uparrow m_{final}) \cap Cov_{\mathcal{N}} \tag{1}$$

We get that

$$\begin{array}{rcll} \uparrow B^I_{k+1} & = & \uparrow(cpre_{\mathcal{N}}(B^I_k) \cap I) \ \cup \ \uparrow B^I_k & [\text{Lemma 3.2}] \\ & \supseteq & \uparrow(cpre_{\mathcal{N}}(B^I_k) \cap Cov_{\mathcal{N}}) & [Cov_{\mathcal{N}} \subseteq I] \\ & \supseteq & (\uparrow cpre_{\mathcal{N}}(B^I_k)) \ \cap \ Cov_{\mathcal{N}} & [Cov_{\mathcal{N}} = \downarrow Cov_{\mathcal{N}}] \\ & = & pre_{\mathcal{N}}(\uparrow B^I_k) \ \cap \ Cov_{\mathcal{N}} & [\text{Corollary 3.5}] \\ & \supseteq & pre^{k+1}_{\mathcal{N}}(\uparrow m_{final}) \ \cap \ Cov_{\mathcal{N}} & [\textit{Equation (1)}] \end{array}$$

This concludes the proof of the lemma. □

Lemma 4.3. *The sequence (B^I_k) is ultimately stationary.*

Proof. The definition of (B^I_k) entails, with Lemma 3.2, that the sequence of upward-closed sets $(\uparrow B^I_k)$ is growing. It follows from Lemma 3.1 that $(\uparrow B^I_k)$ is ultimately stationary. The observation that $B^I_k = \mathrm{Min}(\uparrow B^I_k)$ for every k concludes the proof of the lemma. This observation is an easy consequence of the fact that

$$\mathrm{Min} \uparrow S \ = \ \mathrm{Min}\, S \ = \ \mathrm{Min}(\mathrm{Min}\, S)$$

for every subset $S \subseteq \mathbb{N}^P$. The proof of this fact is left to the reader. □

Lemma 4.4. *Assume that $Cov_{\mathcal{N}} \subseteq I$. It holds that $m_{final} \in Cov_{\mathcal{N}}$ if, and only if, there exists $k \in \mathbb{N}$ such that $\downarrow Init$ intersects B^I_k.*

Proof. Observe that $pre^*_{\mathcal{N}}(\uparrow m_{final}) = \bigcup_k pre^{\leq k}_{\mathcal{N}}(\uparrow m_{final})$. We derive from Lemma 4.2 that

$$pre^*_{\mathcal{N}}(\uparrow m_{final}) \cap Cov_{\mathcal{N}} \ \subseteq \ \bigcup_k \uparrow B^I_k \ \subseteq \ pre^*_{\mathcal{N}}(\uparrow m_{final})$$

Recall that $m_{final} \in Cov_{\mathcal{N}}$ if, and only if, $Init \cap pre^*_{\mathcal{N}}(\uparrow m_{final})$ is non-empty. Since $Init \subseteq Cov_{\mathcal{N}}$, we get that $m_{final} \in Cov_{\mathcal{N}}$ if, and only if, $Init \cap \bigcup_k \uparrow B^I_k$ is non-empty. This last condition is equivalent to $\downarrow Init \cap \bigcup_k B^I_k$ is non-empty. □

We show the correctness of the algorithm by first providing a description of B^I_{k+1} with respect to B^I_k that matches the way it is computed in our algorithm. In fact, by observing that $\mathrm{Min}(X \cup Y) = \mathrm{Min}((X \setminus \uparrow Y) \cup Y)$ for any set X, Y of configurations, we derive the following equalities for every $k \in \mathbb{N}$ where N^I_k and P^I_k are some finite sets:

$$\begin{aligned} N^I_k &= cpre_{\mathcal{N}}(B^I_k) \setminus \uparrow B^I_k \\ P^I_k &= N^I_k \cap I \\ B^I_{k+1} &= \mathrm{Min}(B^I_k \cup P^I_k) \end{aligned}$$

We observe that the sets N, B, P computed by the algorithm during the kth execution of the main loop are the sets N^I_k, B^I_k, P^I_k.

The termination and the correctness of the algorithm are proved thanks to the following lemma:

`ICover`$(\mathcal{N}, m_{final}, I)$

Input: A Petri Net $\mathcal{N} = (P, T, F, Init)$, a target marking $m_{final} \in \mathbb{N}^P$ and a set I such that $Cov_{\mathcal{N}} \subseteq I$

Output: Whether there exist two markings $m_{init} \in Init$ and $m \in \mathbb{N}^P$ such that $m_{init} \xrightarrow{*} m$ and $m \geq m_{final}$.

```
1  begin
2      if m_final ∈ I then
3          B ← {m_final}
4      else
5          B ← ∅
6      while ↓Init ∩ B = ∅ do
7          N ← cpre_N(B) \ ↑B                    /* new predecessors */
8          P ← N ∩ I                    /* prune uncoverable markings */
9          if P = ∅ then
10             return False
11         B ← Min(B ∪ P)
12     return True
```

Lemma 4.5. *$B^I_k = B^I_{k+1}$ if, and only if, $P^I_k = \emptyset$.*

Proof. Assume first that $B^I_k = B^I_{k+1}$. Since $B^I_{k+1} = \mathrm{Min}(B^I_k \cup P^I_k)$, it follows that $P^I_k \subseteq \uparrow B^I_{k+1}$. From $B^I_k = B^I_{k+1}$, we get $P^I_k \subseteq \uparrow B^I_k$. Moreover, from $P^I_k \subseteq N^I_k$ and $N^I_k \cap \uparrow B^I_k = \emptyset$, we get $P^I_k \cap \uparrow B^I_k = \emptyset$. It follows that $P^I_k = \emptyset$. Conversely, if $P^I_k = \emptyset$ we get $B^I_{k+1} = \min(B^I_k) = B^I_k$ since any B set is defined as a set of minimal elements. □

Theorem 4.6. *The procedure* `ICover` *terminates on every input and is correct.*

Proof. For the termination, Lemma 4.3 shows that the sequence $(B^I_k)_{k\in\mathbb{N}}$ is ultimately stationary. It follows that there exists $k \in \mathbb{N}$ such that $B^I_{k+1} = B^I_k$. Lemma 4.5 shows that $P^I_k = \emptyset$. Therefore the algorithm cannot execute the main loop more than k times. Hence, the termination is guaranteed. The correctness directly follows from Lemmas 4.4 and 4.5. □

Remark 4.7. The algorithm `ICover` can be implemented just by assuming that the membership problem for I and $\downarrow Init$ are decidable. In fact, line 7 can be implemented by removing from $cpre_{\mathcal{N}}(B)$ the markings m such that $m \geq b$ for some $b \in B$. □

Remark 4.8. In the sequel, we assume for effectivity reasons that sets of initial markings are denoted by conjunctions of linear inequalities of the form $m_{init}(p) \leq n$ where $p \in P$ and $n \in \mathbb{N}$. In practice, in the file format `spec` used for the benchmarks of Section 7, the set of initial markings is given by a sequence of constraints, for each place p, of the type p *op* n, where *op* $\in \{=, \leq, \geq\}$ and

$n \geq 0$. For the coverability problem, considering $\downarrow Init$ instead of $Init$ does not change the result. This is what we did for the experiments. To extract $Init$ from the format `spec`, we proceed as follows: if for a place p the constraint is of the form $p \leq n$ or $p = n$, then we have the inequalities $m_{init}(p) \leq n$ and if the constraint is of the form $p \geq n$, then we do not have any constraint on it. Observe that lines 2, 6 and 8 are know decidable. □

Remark 4.9. Petri nets obtained by translation from high-level concurrent programs often contain transitions that cannot be fired from any reachable marking. An over-approximation of $Cov_{\mathcal{N}}$ can be used in a pre-processing algorithm to filter out some of them. Basically, if a transition t is not enabled in any marking of the set I, then it can be safely removed without modifying the coverability set. Algorithmically, when I is downward-closed, detecting such a property just reduces to a membership problem in I. In fact, a transition t is enabled in a downward-closed set of markings D if, and only if, D contains the marking m_t defined by $m_t(p) = F(p,t)$ for every place p. □

The algorithm `ICover` is parametrized by a set of markings I that over-approximates the coverability set. This set must be carefully selected. On the one hand, it must be precise enough to discard markings (at line 8) and accelerate the main loop. On the other hand, the membership problem in I must be very efficient to avoid slowing down the main loop. The next two sections show how to generate such sets with a good tradeoff. Since the over-approximations that we will define are downward-closed, it will be possible to easily apply Remark 4.9.

We focus, in the remainder of the paper, on so-called downward-closed invariants. Formally, an *invariant* for a Petri net $\mathcal{N} = (P, T, F, Init)$ is any subset $I \subseteq \mathbb{N}^P$ that contains every reachable marking, i.e., every marking m such that $m_{init} \xrightarrow{*} m$ for some $m_{init} \in Init$. Notice that downward-closed invariants are exactly the downward-closed over-approximations of the coverability set.

Remark 4.10. We can combine different over-approximations $I_1, I_2, \ldots$ of the coverability set by taking their intersection. □

5. State Inequation for Downward-Closed Invariants

The state equation [10] provides a simple over-approximation of Petri net reachability relations that was successfully used in two recent algorithms for deciding the coverability problems [6, 7]. This equation is obtained by introducing the total function $\Delta(t)$ in $\mathbb{Z}^P$ called the *displacement* of a transition t and defined for every place p by $\Delta(t)(p) = F(t,p) - F(p,t)$. Let us assume that a marking m_{final} is in the coverability set of a Petri net $\mathcal{N}$. It follows that there exist an initial marking m_{init}, a word $t_1 \ldots t_k$ of transitions, and a marking $m \geq m_{final}$ such that $m_{init} \xrightarrow{t_1} \cdots \xrightarrow{t_k} m$. We derive the following relation:

$$m_{init} + \Delta(t_1) + \cdots + \Delta(t_k) = m \geq m_{final}$$

By reordering the sum $\Delta(t_1) + \cdots + \Delta(t_k)$, we can group together the displacements $\Delta(t)$ corresponding to the same transition t. Denoting by $\lambda(t)$ the number of occurrences of t in the word $t_1 \dots t_k$, we get:

$$m_{init} + \sum_{t \in T} \lambda(t)\Delta(t) \geq m_{final} \tag{2}$$

The relation (2) is called the *state inequation* for the coverability problem. Notice that a similar equation can be derived for the reachability problem by replacing the inequality by an equality. We do not consider this equality in the sequel since we restrict our attention to the coverability problem. We introduce the following set I_S where $\mathbb{Q}_{\geq 0}$ is the set of non-negative rational numbers.

$$I_S = \{m \in \mathbb{N}^P \mid \exists m_{init} \in Init\ \exists \lambda \in \mathbb{Q}_{\geq 0}^T : m_{init} + \sum_{t \in T} \lambda(t)\Delta(t) \geq m\} \tag{3}$$

Proposition 5.1. *The set I_S is a downward-closed invariant with a polynomial-time membership problem.*

The downward-closed invariant I_S contains $Cov_{\mathcal{N}}$ so we may use it as parameter to the `ICover` algorithm of Section 4. A more precise downward-closed invariant can be obtained by requiring that $\lambda \in \mathbb{N}^T$. In particular, the pruned backward algorithm presented in Section 4 should produce smaller sets of markings with this more precise invariant. In practice, we do not observe any significant improvement on a large set of benchmarks. Moreover, whereas the membership problem of a marking m is decidable in polynomial time when λ ranges over $\mathbb{Q}_{\geq 0}^T$, the problem becomes NP-complete when λ is restricted to $\mathbb{N}^T$.

6. Sign Analysis for Downward-Closed Invariants

The invariant presented in the previous section over-approximates the behavior of Petri nets by allowing places to hold negative amounts of tokens. In this section, we explore another direction by ignoring the precise numbers of tokens in each place and only accounting for their sign (namely, zero or positive).

We call a place $p \in P$ a *zero place* when $m(p) = 0$ for every marking $m \in Cov_{\mathcal{N}}$. Given a set Z of zero places, we let

$$I_Z = \{m \in \mathbb{N}^P \mid \bigwedge_{p \in Z} m(p) = 0\} \tag{4}$$

The set I_Z is obviously a downward-closed invariant, hence, it contains $Cov_{\mathcal{N}}$. Moreover, I_Z has a linear-time membership problem. So we may use I_Z as parameter to the `ICover` algorithm of Section 4. The question remains how to compute a suitable set Z of zero places.

Ideally, we get the most precise over-approximation I_Z by taking Z to be the set of all zero places. However, the question whether a given place is not a zero place is equivalent to (i.e., logspace interreducible with) the coverability problem,

which is the problem that we want to solve in the first place. So we settle for the computation of a subset of zero places, using data-flow sign analysis [11].

Rephrased in the context of Petri nets, an invariant I is said to be *inductive* if $m \xrightarrow{t} m'$ and $m \in I$ implies $m' \in I$. Sign analysis can be formulated as the computation of the maximal (for inclusion) subset $Z \subseteq P$ such that the set I_Z defined in Equation (4) is an inductive invariant. The unicity of that set is immediate since the class of sets Z such that I_Z is an inductive invariant is clearly closed under union. In the sequel, Z denotes the maximal set satisfying this property, and this maximal set is shown to be computable in polynomial time thanks to a fixpoint propagation. We introduce the operator $\mathrm{prop}_t : 2^P \to 2^P$ associated to a transition t and defined for any set Q of places as follows:

$$\mathrm{prop}_t(Q) = \begin{cases} out(t) & \text{if } in(t) \subseteq Q \\ \emptyset & \text{otherwise} \end{cases}$$

It is understood that for a transition $t \in T$, $in(t) = \{p \in P \mid F(p,t) > 0\}$ and $out(t) = \{p \in P \mid F(t,p) > 0\}$. Intuitively, if t is a transition such that $in(t) \subseteq Q$ then from a marking with large number of tokens in each place of Q, it is possible to fire t. In particular places q in $out(t)$ cannot be in Z. This property is formally stated by the following lemma.

Lemma 6.1. *We have* $\mathrm{prop}_t(Q) \subseteq P \backslash Z$ *for every set* $Q \subseteq P \backslash Z$.

Proof. We can assume without loss of generality that $in(t) \subseteq Q$ since otherwise the set $\mathrm{prop}_t(Q)$ is empty. For the same reason, we can assume that there exists $q \in out(t)$. Let us prove that such a place q cannot be in Z. We introduce the markings m_t and m'_t defined by $m_t(p) = F(p,t)$ and $m'_t(p) = F(t,p)$ for every $p \in P$. Those markings are the minimal ones satisfying $m_t \xrightarrow{t} m'_t$. Observe that for every $p \in Z$, we have $p \in P \backslash Q$ since $Q \cap Z$ is empty. It follows that $F(p,t) = 0$ for every $p \in Z$. Hence m_t is in the inductive invariant I_Z. Since $m_t \xrightarrow{t} m'_t$, we deduce that $m'_t \in I_Z$. As $F(t,q) > 0$, we get $m'_t(q) > 0$. Hence $q \notin Z$. □

The set Z can be computed as a fixpoint by introducing the non-decreasing sequence $Q_0, Q_1, \ldots$ of places defined as follows:

$$Q_0 = \{q \in P \mid \exists m_{init} \in Init\ m_{init}(q) > 0\}$$
$$Q_{k+1} = Q_k \cup \bigcup_{t \in T} \mathrm{prop}_t(Q_k)$$

Let us notice that the set $Q = \bigcup_{k \geq 0} Q_k$ is computable in polynomial time. The following lemma shows that Q provides the set Z as a complement.

Lemma 6.2. *We have* $Z = P \backslash Q$.

Proof. Since $Q_0 \subseteq P \backslash Z$, Lemma 6.1 shows by induction that $Q_k \subseteq P \backslash Z$ for every k. It follows that $Q \subseteq P \backslash Z$. We derive $Z \subseteq P \backslash Q$. The converse inclusion

is obtained by proving that the set $M = \{m \in \mathbb{N}^P \mid \bigwedge_{p \in P \setminus Q} m(p) = 0\}$ is an inductive invariant. First of all, since $Q_0 \subseteq Q$, we deduce that there exists $m_{init} \in Init \cap M$. Now let us consider $m \in M$ and a transition t such that $m \xrightarrow{t} m'$ for some marking m'. Observe that $m(p) \geq F(p,t)$ for every $p \in P$. In particular, for $p \in P \setminus Q$, the equality $m(p) = 0$ implies $F(p,t) = 0$. Assume by contradiction that $m' \notin M$. In that case, there exists $q \in P \setminus Q$ such that $m'(q) > 0$. Since $m'(q) = m(q) + F(t,q) + F(q,t)$ and $m(q) = 0 = F(q,t)$, we deduce that $F(t,q) > 0$. Thus $q \in \mathrm{prop}_t(Q)$. By definition of Q, we get $q \in Q$ and we obtain a contradiction. Thus M is an inductive invariant. By maximality of Z, we get the inclusion $P \setminus Q \subseteq Z$. Thus $Z = P \setminus Q$. □

Corollary 6.3. *The set Z is computable in polynomial time.*

As mentioned above, we may use I_Z as parameter to the `ICover` algorithm of Section 4. In practice, though, we use it as a pre-processing step to simplify the Petri net under analysis. This was already hinted at in Remark 4.9. The remainder of this section presents this pre-processing in detail.

Consider a Petri net $\mathcal{N} = (P, T, F, Init)$ and two subsets $Q \subseteq P$ and $U \subseteq T$. Given a marking $m \in \mathbb{N}^P$, we let $m|_Q$ denote the *restriction* of m to Q, i.e., the marking in $\mathbb{N}^Q$ defined by $m|_Q(p) = m(p)$ for every $p \in Q$. The *restriction* of the flow function F to Q and U is the function $F|_{Q,U} : (Q \times U) \cup (U \times Q) \to \mathbb{N}$ defined by $F|_{Q,U}(p,t) = F(p,t)$ and $F|_{Q,U}(t,p) = F(t,p)$ for every $p \in Q$ and every $t \in U$. Finally, the *restriction* of the Petri net $\mathcal{N}$ to Q and U is the Petri net $\mathcal{N}|_{Q,U} = (Q, U, F|_{Q,U}, Init|_Q)$ where $Init|_Q = \{m_{init}|_Q \mid m_{init} \in Init\}$.

Theorem 6.4. *Let Q be a set of places of a Petri net $\mathcal{N}$ such that all places that are not in Q are zero places. Let U denote the set of transitions t of $\mathcal{N}$ such that $in(t) \subseteq Q$. A marking m is coverable in $\mathcal{N}$ if, and only if, $m|_Q$ is coverable in $\mathcal{N}|_{Q,U}$ and $m(p) = 0$ for every place $p \notin Q$.*

Proof. Assume that m is coverable in $\mathcal{N}$. There exists $m_0 \xrightarrow{t_1} m_1 \cdots \xrightarrow{t_k} m_k$ in $\mathcal{N}$ such that $m_0 \in Init$ and $m_k \geq m$. Observe that each $m_i \in Cov_{\mathcal{N}}$, hence, $m_i(p) = 0$ for every place $p \notin Q$. It follows that each t_i satisfies $in(t_i) \subseteq Q$, hence, $t_i \in U$. The above definition of restriction immediately entails that $m_0|_Q \xrightarrow{t_1} m_1|_Q \cdots \xrightarrow{t_k} m_k|_Q$ in $\mathcal{N}|_{Q,U}$. Moreover, $m_0|_Q \in Init|_Q$ and $m_k|_Q \geq m|_Q$. We have thus shown that $m|_Q$ is coverable in $\mathcal{N}|_{Q,U}$. Since $m_k \geq m$ and $m_k(p) = 0$ for every place $p \notin Q$, we also get that $m(p) = 0$ for every place $p \notin Q$.

Conversely, assume that $m|_Q$ is coverable in $\mathcal{N}|_{Q,U}$ and $m(p) = 0$ for every place $p \notin Q$. There exists $x_0 \xrightarrow{t_1} x_1 \cdots \xrightarrow{t_k} x_k$ in $\mathcal{N}|_{Q,U}$ such that $x_0 \in Init|_Q$ and $x_k \geq m|_Q$. To show that m is coverable in $\mathcal{N}$, we lift this sequence of steps into a sequence of steps in $\mathcal{N}$. Let us define the markings $m_0, \ldots, m_k$ of $\mathcal{N}$ by

$$m_i(p) = \begin{cases} x_i(p) & \text{if } p \in Q \\ F(t_1,p) + \cdots + F(t_i,p) & \text{otherwise} \end{cases}$$

It is understood that F denotes the flow function of $\mathcal{N}$. Since each t_i is a transition in U, it holds that $F(p, t_i) = 0$ for every place $p \notin Q$. It follows that $m_0 \xrightarrow{t_1} m_1 \cdots \xrightarrow{t_k} m_k$ in $\mathcal{N}$. Since $x_k \geq m|_Q$ and $m(p) = 0$ for every place $p \notin Q$, we get that $m_k \geq m$. It remains to show that $m_0 \in Init$. Recall that $x_0 \in Init|_Q$. This means that $x_0 = m_{init}|_Q$ for some $m_{init} \in Init$. Since $m_{init} \in Cov_{\mathcal{N}}$, $m_{init}(p) = 0$ for every place $p \notin Q$. It follows that $m_{init} = m_0$, which concludes the proof. □

Remark 6.5. The above theorem allows us to simplify the input to the coverability problem as follows. Given a Petri net $\mathcal{N} = (P, T, F, Init)$ and a target marking $m_{final} \in \mathbb{N}^P$, we compute the maximal subset $Z \subseteq P$ such that I_Z is an inductive invariant. This set is computable in polynomial time by Corollary 6.3. If $m_{final}(p) \neq 0$ for some place $p \in Z$ then we answer that m_{final} is not coverable in $\mathcal{N}$. Otherwise, we compute the restrictions $\mathcal{N}|_{Q,U}$ and $m_{final}|_Q$, where $Q = P \setminus Z$ and $U = \{t \in T \mid in(t) \subseteq Q\}$, and we solve coverability for $\mathcal{N}|_{Q,U}$ and $m_{final}|_Q$. Theorem 6.4 guarantees the correctness of this approach. □

7. Experimental Evaluation

We implemented our approach using the `QCover` [7] tool as a starting point. This tool, which implements a backward coverability algorithm for Petri nets, is written in Python and relies on the SMT-solver $z3$ [16]. `QCover` also uses some other heuristics that we kept unchanged. `QCover` was competitive with others tools especially for uncoverable Petri net. Only the `BFC` tool performs significantly better on coverable Petri net. We have made two modifications to `QCover`. First, we have added a pre-processing step based on sign analysis (see Remark 6.5). Second, we have replaced their pruning technique, which is based on coverability in continuous Petri nets, by the state inequation presented in Section 5. `ICover` is available on GitHub as a fork [13] of `QCover` [12].

To test our implementation, we used the same benchmark as `Petrinizer` [6] and `QCover` [7]. It comprises models from various sources: `Mist` [17], `BFC` [5], `Erlang` programs abstracted into Petri nets [18], as well as so-called `medical` and `bug_tracking` examples [6]. We let each tool work for 2000 seconds in a machine on Ubuntu Linux 14.04 with Intel(R) Core(TM) i7-4770 CPU at 3.40 GHz with 16 GB of memory for each benchmark. The computation times are the sum of the `system` and `user` times. Overall `QCover` solved 106 uncoverable instances on 115 Petri net and 37 coverable problems on 61 Petri nets. `ICover` was able to find one more coverable instance. In fact calling `QCover` on the Petri net computed by the pre-processing, that we will call `QCover/Pp`, can even solve one more uncoverable instance than `ICover`. On the 143 instances that `QCover` solved, the tool took 9729 seconds, `QCover/Pp` used 6827 seconds, and `ICover` used only 5213 seconds.

Figure 2(a) shows the comparison between `ICover` and `QCover` in time. The straight line represents when the two tools took the same time. Each dot represents a coverability question. When the dot is under the line, it means

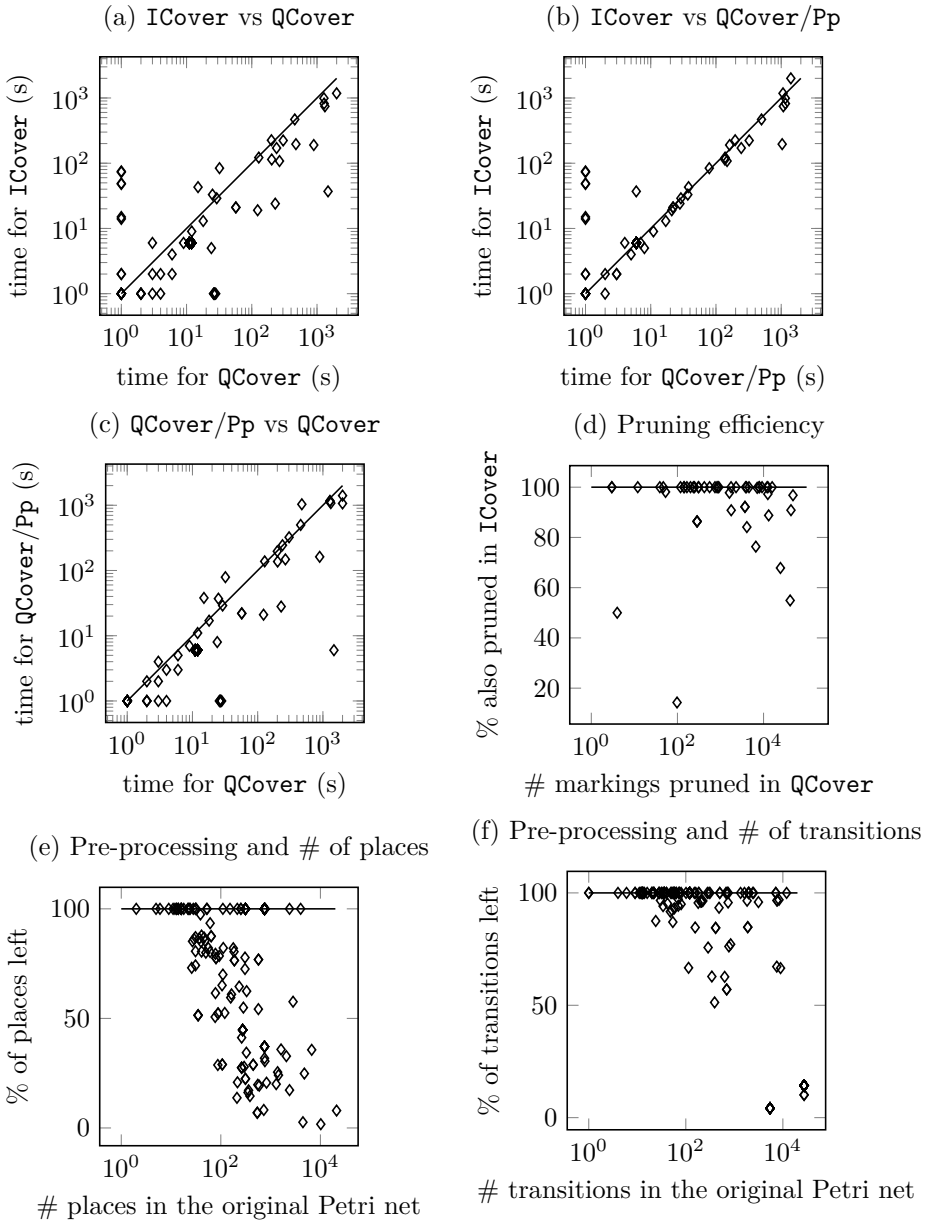

Figure 2: Experimental results for `ICover`, `QCover` and `QCover/Pp`

that `ICover` was faster than `QCover` and conversely. There are three instances where `QCover` performs very well, under a second, and where `ICover` took a few tens of seconds to answer. For the three cases, the formula used by `QCover` for

coverability in $\mathbb{Q}$ was enough to discard the target as uncoverable and it did not have to enter in the while loop. But `ICover` was not able to discard the target and had to enter the while loop in the three cases. We also see two dots above the line at the middle of the figure. The pre-processing took respectively 12 and 45 seconds while the initial Petri net was solved by `QCover` in respectively 16 and 33 seconds.

Figure 2(b) and (c) show the intermediate comparisons: `ICover` versus `QCover/Pp` and `QCover/Pp` versus `QCover`. We can observe that the pre-processing has a major impact on the good performance of `ICover` compared to `QCover`.

Figure 2(e) and Figure 2(f) aims to show the effect of the pre-processing on the size of Petri nets. The former show the percentage of places left after pre-processing. Some Petri nets kept all their places but others were left with only 2.5% of their initial places. And most of Petri nets lost a significant number of places. The latter shows the percentages of transitions left after the pre-processing. Overall less transitions were cut than places. Half of the Petri nets kept all their transitions, but some were left with only 4% of their initial transitions.

Figure 2(d) compares the efficiency of pruning between `ICover` and `QCover`. Again, each dot represents a coverability question. As discussed in Section 8, `QCover` always prunes at least as many markings as `ICover` (but at the expense of more complex pruning tests). A value of 100% means that `ICover` was able to prune the same markings as `QCover`. It turns out that on most instances, this perfect value of 100% is obtained. This is rather surprising at first sight, and warrants an investigation, which is the focus of the next section.

8. Comparison with Continuous Petri Net

Continuous Petri nets are defined like Petri nets except that transitions can be fired a non-negative rational number of times. The firing of such a transition produces markings with non-negative rational numbers of tokens. Under such a semantics, called the *continuous semantics*, the reachability problem was recently proved to be decidable in polynomial time [8]. Based on this observation, the tool `QCover` implements the pruning backward coverability algorithm presented in Section 4 with a downward-closed invariant derived from the continuous semantics. Whereas this invariant is more precise than the downward-closed invariant obtained from the state inequation introduced in Section 5, we have seen in Section 7 that such an improvement is overall not useful in practice for pruning in the backward coverability algorithm. In this section, we provide a simple structural condition on Petri nets in such a way the two kinds of downward-closed invariants derived respectively from the continuous semantics and the state inequation are "almost" equal. This structural condition is shown to be natural since it is fulfilled by the Petri nets obtained after the pre-processing introduced in Remark 6.5.

A *continuous marking* is a mapping $m \in \mathbb{Q}_{\geq 0}^P$ where $\mathbb{Q}_{\geq 0}$ denotes the set of non-negative rational numbers, and P the set of places. Given $r \in \mathbb{Q}_{\geq 0}$ and a

transition t, the continuous rt-step binary relation $\overset{rt}{\dashrightarrow}$ over the continuous markings is defined by

$$m \overset{rt}{\dashrightarrow} m' \;\Leftrightarrow\; \forall p \in P : m(p) \geq r.F(p,t) \wedge m'(p) = m(p) - r.F(p,t) + r.F(t,p)$$

The one-step continuous binary relation $\dashrightarrow$ is the union of these rt-step relations. Formally, $m \dashrightarrow m'$ if there exists $r \in \mathbb{Q}_{\geq 0}$ and $t \in T$ such that $m \overset{rt}{\dashrightarrow} m'$. The many-step continuous binary relation $\overset{*}{\dashrightarrow}$ is the reflexive-transitive closure of $\dashrightarrow$. We also introduce the binary relation $\overset{\infty}{\dashrightarrow}$ defined over the continuous markings by $m \overset{\infty}{\dashrightarrow} m'$ if there exists a sequence $(m_k)_{k\geq 0}$ of continuous markings that *converges* towards m' with the classical topology on $\mathbb{Q}_{\geq 0}^P$ and such that $m \overset{*}{\dashrightarrow} m_k$ for every k.

Example 8.1. Let us look back at the simple Petri net $\mathcal{N}$ depicted in Figure 1. For every positive natural number k, we have:

$$(1,0,0) \overset{\frac{1}{k}t_1}{\dashrightarrow} (1-\frac{1}{k},\frac{1}{k},0) \overset{\frac{1}{k}t_2\frac{1}{k}t_3}{\dashrightarrow} (1-\frac{1}{k},\frac{2}{k},\frac{1}{k}) \cdots \overset{\frac{1}{k}t_2\frac{1}{k}t_3}{\dashrightarrow} (1-\frac{1}{k},1+\frac{1}{k},1)$$

It follows that $(1,0,0) \overset{\infty}{\dashrightarrow} (1,1,1)$. Notice that the relation $(1,0,0) \overset{*}{\dashrightarrow} (1,1,1)$ does not hold. □

The downward-closed invariant used in the tool `QCover` for implementing the pruning backward algorithm is defined as follows:

$$I_C = \{m \in \mathbb{N}^P \mid \exists m_{init} \in Init\ \exists m' \in \mathbb{Q}_{\geq 0}^P : m_{init} \overset{*}{\dashrightarrow} m' \geq m\} \tag{5}$$

Recall that in Section 5 we introduced the set I_S for denoting the downward-closed invariant derived from the state inequation. The following result provides a characterization of that invariant when the Petri net satisfies a structural condition.

Theorem 8.2 ([14, Theorem 7])**.** *Assume that $Init$ is a set of markings of the following form where $Q \subseteq P$ and $x \in \mathbb{N}^Q$:*

$$Init = \{m_{init} \in \mathbb{N}^P \mid \bigwedge_{q \in Q} m_{init}(q) \leq x(q)\}$$

And assume that every transition is fireable from the downward-closed invariant I_Z introduced in Section 6. We have:

$$I_S = \{m \in \mathbb{N}^P \mid \exists m_{init} \in Init\ \exists m' \in \mathbb{Q}_{\geq 0}^P : m_{init} \overset{\infty}{\dashrightarrow} m' \geq m\} \tag{6}$$

Proof. The statement of Theorem 7 in [14] is wrong since it is based on a too strong definition of limit-reachability. Rephrased with our notations, that theorem claims that I_S is the set of markings m such that there exists a sequence $(m_k)_{k\geq 0}$ of continuous markings that *converges* towards a marking $m' \geq m$ and such that $m_0 \in Init$ and $m_k \overset{*}{\dashrightarrow} m_{k+1}$ for every k. Example 8.1 shows that this claim is wrong. However, the proof becomes correct with our definitions

and notations. The proof is obtained as follows. First of all, if m is a marking such that there exists $m_{init} \xrightarrow{\infty} m' \geq m$ for some $m_{init} \in Init$ and $m' \in \mathbb{Q}_{\geq 0}^P$, we derive that there exists a sequence $(m_k)_{k\geq 0}$ of continuous markings that *converges* towards m' such that $m_{init} \xrightarrow{*} m_k$ for every k. It follows that m_k satisfies the system of linear equations of I_S for every k. As a limit, it follows that m' satisfies the same system. In particular $m \in I_S$.

Conversely, let us assume that m is a marking in I_S, and assume that every transition is fireable from the downward-closed invariant I_Z introduced in Section 6. Given a word $\sigma = (r_1 t_1) \dots (r_k t_k)$ with $(r_j, t_j) \in \mathbb{Q}_{\geq 0} \times T$, we denote by $\xrightarrow{\sigma}$ the binary relation over the continuous markings defined as the concatenation $\xrightarrow{r_1 t_1} \cdots \xrightarrow{r_k t_k}$. Moreover, we say that σ is fireable from a continuous marking m if there exists a continuous marking y such that $x \xrightarrow{\sigma} y$. Given $r \in \mathbb{Q}_{\geq 0}$, we define $r\sigma$ as the word $(rr_1 t_1) \dots (rr_k t_k)$. Observe that if $x \xrightarrow{\sigma} y$ then $x \xrightarrow{\varepsilon\sigma} x + \varepsilon(y - x)$ for every ε satisfying $0 \leq \varepsilon \leq 1$. Moreover, if $x \xrightarrow{\sigma} y$ then $rx \xrightarrow{r\sigma} ry$ for every $r \geq 0$.

Let $(Q_k)_{k\in\mathbb{N}}$ be the sequence introduced in Section 6 and let us show by induction on k that there exists a sequence of continuous markings $(x_k)_{k\in\mathbb{N}}$ such that $x_0 \in Init$, $x_k \xrightarrow{*} x_{k+1}$, and $x_k(q) > 0$ for every $q \in Q_k$. The rank k is immediate thanks to the special form of $Init$. Assume the sequence $x_0, \dots, x_k$ built. Notice that there exists a sequence $t_1, \dots, t_n$ of transitions in T such that $F(p, t_j) = 0$ for every $p \in P \backslash Q_k$ and for every $1 \leq j \leq n$, and such that $Q_{k+1} \backslash Q_k \subseteq \{q \mid F(t, q) > 0\}$. Observe that for a rational number $\epsilon > 0$ small enough, we have $x_k \xrightarrow{\varepsilon t_1 \dots \varepsilon t_n} x_{k+1}$ where x_{k+1} satisfies $x_{k+1}(q) > 0$ for every $q \in Q_{k+1}$.

It follows that there exists $x_0 \in Init$, a continuous marking x, an a word w such that $x_0 \xrightarrow{w} x$ with $x(q) > 0$ for every $q \in Q$ where $Q = \bigcup_k Q_k$. Now, let us consider a marking $m \in I_S$. There exist $m_{init} \in Init$ and a word $t_1 \dots t_k$ of transitions such that $m_{init} + \Delta(t_1) + \cdots + \Delta(t_k) = y \geq m$. Thanks to the form of $Init$, we can assume without loss of generality that $m_{init} \geq x_0$. Let $\sigma = (1t_1) \dots (1t_k)$ and let ε such that $0 < \varepsilon < 1$. From $x_0 \xrightarrow{w} x$ and $m_{init} \geq x_0$, we derive $m_{init} \xrightarrow{\varepsilon w} (1 - \varepsilon)[m_{init} + z_\varepsilon]$ where $z_\varepsilon = \frac{\varepsilon}{1-\varepsilon}(x + m_{init} - x_0)$. Since $z_\varepsilon(q) > 0$ for every $q \in Q$, and every transition is fireable from I_Z, we deduce that there exists $n \geq 1$ large enough such that the word $\frac{1}{n}\sigma$ is fireable from z_ε. Since $\frac{1}{n}\sigma$ is fireable from z_ε, it is in particular fireable from $\frac{k}{n} m_{init} + \frac{n-k}{n} y + z_\varepsilon$ for every $0 \leq k < n$. From $m_{init} + \Delta(t_1) + \cdots + \Delta(t_k) = y$, we get the following relation:

$$\frac{n-k}{n} m_{init} + \frac{k}{n} y + z_\varepsilon \xrightarrow{\frac{1}{n}\sigma} \frac{n-(k+1)}{n} m_{init} + \frac{k+1}{n} y + z_\varepsilon$$

By concatenating the previous relations, we deduce that $m_{init} + z_\varepsilon \xrightarrow{*} y + z_\varepsilon$. With $m_{init} \xrightarrow{*} (1 - \varepsilon)[m_{init} + z_\varepsilon]$, we deduce the relation $m_{init} \xrightarrow{*} (1 - \varepsilon)[y + z_\varepsilon] = (1 - \varepsilon)y + \varepsilon(x + m_{init} - x_0)$. It follows, when ε tends to 0, that $m_{init} \xrightarrow{\infty} y$. □

The two equalities Equation (5) and Equation (6) show that I_S and I_C are very similar for Petri nets satisfying the structural condition stated in

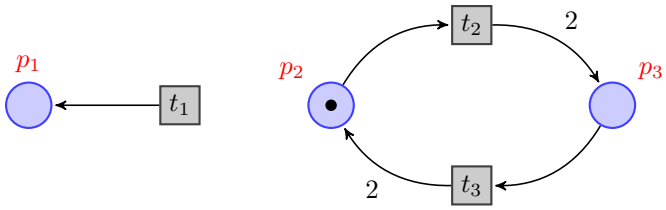

Figure 3: A Petri net with omega places

Theorem 8.2. This condition is fulfilled by the Petri nets produced by the pre-processing algorithm introduced in Remark 6.5. Notice that even if the membership problem in I_S and I_C are both decidable in polynomial time, the extra computational cost for deciding the membership problem for the invariant I_C, even for efficient SMT solvers like `Z3`, is not negligible. Naturally, if a marking is in I_C then it is also in I_S, and the converse property is false in general as shown by Example 8.1. However, in practice, we observed that markings that are in I_S are very often also in I_C (see Figure 2(d)), as already mentioned in Section 7.

9. Forward Analysis of Omega Places

We have presented in Sections 5 and 6 two downward-closed invariants, which are over-approximations of $Cov_{\mathcal{N}}$, to be used as parameter to the algorithm `ICover`. In practice, the invariant derived from sign analysis is used as a pre-processing as seen in Remark 6.5. The idea is to eliminate places that we know remain forever empty starting from $Init$. In this section, we exhibit some places that can receive an arbitrary high number of tokens. We show that those places can be safely eliminated for solving coverability problems.

We introduce the unit vector $\mathbf{e}_p$ in $\mathbb{N}^P$ defined as zero everywhere except for the index p where it is one. A place $p \in P$ is called an *omega place* if the marking $m + n\mathbf{e}_p$ is coverable for arbitrarily large natural numbers n and for every coverable marking m. A simple induction on n shows that p is an omega place if, and only if, $Cov_{\mathcal{N}} + \mathbf{e}_p \subseteq Cov_{\mathcal{N}}$. Here, the sum $Cov_{\mathcal{N}} + \mathbf{e}_p$ stands for the set $\{m + \mathbf{e}_p \mid m \in Cov_{\mathcal{N}}\}$.

Example 9.1. Figure 3 depicts a Petri net such that every marking is coverable. It follows that p_1, p_2, p_3 are omega places. □

The following theorem shows that omega places can be safely eliminated without changing results of coverability problems. This reduction is implemented as a pre-processing in our tool `ICover` (see Section 9.1 for experimental evaluations).

Theorem 9.2. *Let Q be a set of places of a Petri net $\mathcal{N}$ such that places that are not in Q are omega places. A marking m is coverable in $\mathcal{N}$ if, and only if, $m|_Q$ is coverable in the restriction $\mathcal{N}|_Q$ of $\mathcal{N}$ to the set of places Q.*

Proof. In order to simplify notations, the step relation $\xrightarrow{t}$ of $\mathcal{N}|_Q$ is denoted by $\xrightarrow{t}_Q$. Observe that if $x \xrightarrow{t} y$ for some markings x, y then $x|_Q \xrightarrow{t}_Q y|_Q$. It follows that if a marking m is coverable in $\mathcal{N}$ then $m|_Q$ is coverable in $\mathcal{N}|_Q$. Let us prove the converse property.

We first prove by induction over k that for every $x_0 \xrightarrow{t_1}_Q x_1 \cdots \xrightarrow{t_k}_Q x_k$ with $x_0 \in Init|_Q$, there exists a marking $m_k \in Cov_{\mathcal{N}}$ such that $x_k \leq m_k|_Q$. The base case $k = 0$ is immediate. Assume the property proved for some k, and assume that $x_k \xrightarrow{t_{k+1}}_Q x_{k+1}$. By induction, there exists a marking $m_k \in Cov_{\mathcal{N}}$ such that $x_k \leq m_k|_Q$. Notice that places of m_k indexed by $P\backslash Q$ can be replaced by arbitrarily large numbers since $P\backslash Q$ is a set of omega places. So, we can assume without loss of generality that $m_k(p) \geq F(p, t_{k+1})$ for every $p \in P\backslash Q$. It follows that there exists m_{k+1} such that $m_k \xrightarrow{t_{k+1}} m_{k+1}$. Since the coverability set is an inductive invariant, it follows that $m_{k+1} \in Cov_{\mathcal{N}}$. Moreover, notice that for every $p \in Q$ we have $m_{k+1}(p) = m_k(p) - F(p, t_{k+1}) + F(t_{k+1}, p) \geq x_k(p) - F(p, t_{k+1}) + F(t_{k+1}, p) = x_{k+1}(p)$. It follows that $x_{k+1} \leq m_{k+1}|_Q$. We have proved the induction.

Now, assume that m is a marking of $\mathcal{N}$ such that $m|_Q$ is coverable in $\mathcal{N}|_Q$. There exists $x_0 \xrightarrow{t_1}_Q x_1 \cdots \xrightarrow{t_k}_Q x_k$ with $x_0 \in Init|_Q$ such that $m|_Q \leq x_k$. From the previous paragraph, there exists $m_k \in Cov_{\mathcal{N}}$ such that $x_k \leq m_k|_Q$. Notice that places of m_k indexed by $P\backslash Q$ can be replaced by arbitrarily large numbers since $P\backslash Q$ is a set of omega places. So, we can assume without loss of generality that $m_k(p) \geq m(p)$ for every $p \in P\backslash Q$. It follows that $m \leq m_k$. Thus m is coverable in $\mathcal{N}$. □

To apply the pre-processing induced by Theorem 9.2, we need to compute omega places. It would be too costly in time to compute them all. In fact, as shown in Lemma 9.3, the detection of omega places is at least as hard as the coverability problem itself. More formally, we introduce the *omega place problem* that asks, given a Petri net $\mathcal{N} = (P, T, F, Init)$ and a place $p \in P$, whether p is an omega place.

Lemma 9.3. *The coverability problem is logspace reducible to the omega place problem.*

Proof. Let $\mathcal{N} = (P, T, F, Init)$ a Petri net and m_{final} a marking. We define $\mathcal{N}_\Omega = (P_\Omega, T_\Omega, F_\Omega, Init_\Omega)$ such that $P_\Omega = P \cup \{p_\Omega\}$ with $p_\Omega \notin P$ a new place, $T_\Omega = T \cup \{t_{final}, t_{add}\}$ is a set of new transitions, $Init_\Omega$ has all the same constraints of $Init$ and a new constraint: $m_{init}(p_\Omega) \leq 0$. And the flow function $F_\Omega : (P_\Omega \times T_\Omega) \cup (T_\Omega \times P_\Omega) \to \mathbb{N}$ is such that

- for all $t \in T$ the transition has the same effect as in $\mathcal{N}$,
- t_{add} is a transition that takes a token in p_Ω, puts a token in every place in $P_\Omega\backslash\{p_\Omega\}$ and puts two tokens in p_Ω, i.e., $F_\Omega(p_\Omega, t_{add}) = 1$, $F_\Omega(p, t_{add}) = 0$ for every place $p \neq p_\Omega$, $F_\Omega(t_{add}, p) = 1$ if $p \in P_\Omega\backslash\{p_\Omega\}$, $F_\Omega(t_{add}, p_\Omega) = 2$, and $F_\Omega(t_{add}, p) = 0$ for every place $p \notin P_\Omega$, and

- t_{final} is the transition such that $F_\Omega(q, t_{final}) = m_{final}(q)$ for all $q \in P$ and that put a token in p_Ω.

We will prove that m_{final} is coverable in $\mathcal{N}$ if, and only if, p_Ω is an omega place in $\mathcal{N}_\Omega$.

First, if m_{final} is coverable in $\mathcal{N}$. Let us prove that all markings in $\mathbb{N}^{P_\Omega}$ are coverable. Because m_{final} is coverable in $\mathcal{N}$, it is coverable in $\mathcal{N}_\Omega$ by definition of F_Ω. It would be then possible to fire t_{final} to cover $m_{final} + \mathbf{e}_{p_\Omega}$. But from $\mathbf{e}_{p_\Omega}$ we can put tokens in every places we want with t_{add}. Hence, all markings are coverable in $\mathcal{N}_\Omega$. We have then that p_Ω is an omega place in $\mathcal{N}_\Omega$.

For the converse, if p_Ω is an omega place in $\mathcal{N}_\Omega$ it means in particular that $\mathbf{e}_{p_\Omega}$ is coverable. Therefore there exist some markings $m_0, m_1, \ldots, m_k$ and some transitions $t_1, t_2, \ldots t_n$ in T_Ω such that $m_0 \in Init_\Omega$ (hence $m_0(p_\Omega) = 0$), for all $0 \leq k < n$, $m_k(p_\Omega) = 0$ and $m_0 \xrightarrow{t_1} m_1 \ldots \xrightarrow{*} m_{n-1} \xrightarrow{t_n} m_n \geq \mathbf{e}_{p_\Omega}$. Only the transition t_{final} can put a token in p_Ω when there is no token in it yet. Therefore $t_n = t_{add}$ and for all $1 \leq k < n$, $t_k \in T$. Furthermore that transition can be fire, by definition, only if there are at least $m_{final}(q)$ tokens for each place $q \in \mathbb{N}^P$. We have then $m_{n-1} \geq m_{final}$ and $m_0 \xrightarrow{t_1} m_1 \ldots \xrightarrow{*} m_{n-1}$ is a path in $\mathcal{N}$. This prove that m_{final} is coverable in $\mathcal{N}$. □

Rather than trying computing the whole set of omega places of a Petri net $\mathcal{N} = (P, T, F, Init)$, we will only use a simple forward fixpoint computation to find some of them. We define a sequence of sets of omega places such that:

$$\Omega_0 = \{p \in P \mid \forall m_{init} \in {\downarrow}Init, m_{init} + \mathbf{e}_p \in {\downarrow}Init\}$$
$$\Omega_{k+1} = \Omega_k \cup \bigcup_{\substack{t \in T \\ in(t) \subseteq \Omega_k}} out(t)$$

The set Ω_0 is exactly the set of places for which there is no constraint in $Init$. The sequence $\Omega_0, \Omega_1, \ldots$ is non-decreasing (for $\subseteq$) and furthermore this sequence is ultimately stationary because $\Omega_k \subseteq P$ for all $k > 0$ and P is finite. We note $\Omega = \bigcup_k \Omega_k$. We will show that Ω is a set of omega places.

We first need to show that we can put a token in any place in Ω from an initial marking that has tokens only in places in Ω_0. But first, we need the following technical lemma.

Lemma 9.4. *We have $x + y \xrightarrow{*} x' + y'$ for every $x \xrightarrow{*} x'$ and $y \xrightarrow{*} y'$.*

Proof. First we prove that if $x \xrightarrow{*} x'$ then for every $z \in \mathbb{N}^P$ we have $x+z \xrightarrow{*} x'+z$. Let $z \in \mathbb{N}^P$ and suppose that $x \xrightarrow{*} x'$. There exist $x_0, x_1, \ldots, x_n$ and $t_1, t_2, \ldots, t_n$ such that $x = x_0 \xrightarrow{t_1} x_1 \ldots \xrightarrow{t_n} x_n = x'$. It is readily seen that for every $0 \leq k < n$, $x_k + z \xrightarrow{t_k} x_{k+1} + z$. We have then $x + z \xrightarrow{*} x' + z$.

To conclude, we have $x + y \xrightarrow{*} x' + y \xrightarrow{*} x' + y'$ for every $x \xrightarrow{*} x'$ and $y \xrightarrow{*} y'$. □

Lemma 9.5. *For all $k > 0$ and for all $p \in \Omega_k$, there exists an initial marking $m_{init} \in \sum_{p \in \Omega_0} \mathbb{N} \cdot \mathbf{e}_p$ such that $m_{init} \xrightarrow{*} m \geq \mathbf{e}_p$ for some marking m.*

Proof. We will prove it by induction on $k > 0$. For the basis $k = 1$, let $p \in \Omega_1$, we have trivially $m_{init} = \mathbf{e}_p \geq \mathbf{e}_p$. Hence the basis $k = 1$ holds.

For the induction step, let $k > 0$. Let $q \in \Omega_{k+1} \backslash \Omega_k$. By construction of Ω_{k+1}, there exists $t \in T$ such that $q \in out(t)$ and $in(t) \subseteq \Omega_k$. By induction hypothesis, for all places $p \in in(t) \subseteq \Omega_k$, there exists $m_{init,p} \in \sum_{p_0 \in \Omega_0} \mathbb{N} \cdot \mathbf{e}_{p_0}$ and m_p such that $m_{init,p} \xrightarrow{*} m_p \geq \mathbf{e}_p$. According to Lemma 9.4 we have $m_{init} = \sum_{p \in in(t)} F(p,t) \cdot m_{init,p} \xrightarrow{*} \sum_{p \in in(t)} F(p,t) \cdot m_p \geq \sum_{p \in in(t)} F(p,t) \cdot \mathbf{e}_p$. Let $m = \sum_{p \in in(t)} F(p,t) \cdot m_p$. We can then fire t from m. We have $m \xrightarrow{t} m'$ with $m'(q) = F(t,q) - F(q,t)$. We have $F(q,t) = 0$ because $q \notin \Omega_k$ and $in(t) \subseteq \Omega_k$ hence $q \notin in(t)$, and $F(t,q) > 0$ because $q \in out(t)$. We have then $m'(q) > 0$ and therefore $m_{init} \xrightarrow{*} m' \geq \mathbf{e}_p$. Furthermore, it is readily seen that $m_{init} \in \sum_{p \in \Omega_0} \mathbb{N} \cdot \mathbf{e}_p$. □

We now show that all places in Ω are indeed omega places.

Lemma 9.6. *For all places $p \in \Omega$, $Cov_{\mathcal{N}} + \mathbf{e}_p \subseteq Cov_{\mathcal{N}}$.*

Proof. Let $p \in \Omega$. Let $m \in Cov_{\mathcal{N}}$ a coverable marking. There exists $m_{init} \in Init$ such that $m_{init} \xrightarrow{*} m' \geq m$. According to Lemma 9.5, there exists $m'_{init} \in \sum_{p \in \Omega_0} \mathbb{N} \cdot \mathbf{e}_p$ such that $m'_{init} \xrightarrow{*} m'' \geq \mathbf{e}_p$. By definition of Ω_0 and direct induction, we have $m_{init} + m'_{init} \in {\downarrow}Init$, and therefore there exists $\delta \in \mathbb{N}^P$ such that $m_{init} + m'_{init} + \delta \in Init$. According to Lemma 9.4 we have $m_{init} + m'_{init} + \delta \xrightarrow{*} m' + m'' + \delta \geq m + \mathbf{e}_p$. Hence $m + \mathbf{e}_p$ is coverable. Therefore $Cov_{\mathcal{N}} + \mathbf{e}_p \subseteq Cov_{\mathcal{N}}$. □

We can now eliminate all the places in Ω that are identified as omega places thanks to Theorem 9.2.

Example 9.7. Figure 3 depicts a Petri net such that every marking is coverable. It follows that p_1, p_2, p_3 are omega places. Observe that $\Omega_0 = \emptyset$ and $\Omega_k = \{p_1\}$ for every $k \geq 1$. It follows that our forward fixpoint computation detects that p_1 is an omega place but fails to detect that p_2 and p_3 are omega places as well. □

9.1. Experimental evaluation

We now continue the experiments seen in Section 7. The computation of the omega places as described before is embedded with the sign analysis. It is an option of `ICover`. We denote by `ICover` $+ \Omega$ the procedure that start with the pre-processing that eliminates some omega places and then proceeds the same way as described in Section 7. `ICover` $+ \Omega$ was able to solve as much uncoverable instance as `ICover` but six more coverable instance. For the total of 144 instances where both algorithm concluded, `ICover` took 6362 seconds and `ICover` $+ \Omega$ took 3134 seconds. Furthermore, `ICover` $+ \Omega$ used only 1964 seconds for the same instances where `QCover` took 9729 seconds.

Remark 9.8. In a restricted Petri net, a transition t can be such that $out(t) = \emptyset$. We can safely remove this transition without changing the result of the computation. □

Now we analyze how many places where cut with the method described in this section. Over the 176 instances, 112 have omega places found. And 102 of these instances have only one omega place. Those last Petri nets were mostly extracted from `tts` systems where one place represents the number of threads at the initial state. And that number has no constraints. Those places are omega places. One instance, `kanban` from the tool `Mist` [17], our pre-processing had computed Ω which is the set of all places. The instance was therefore trivially solved, despite the fact that `QCover` and `ICover` both did not terminate.

As in Section 7, we presents comparison between `ICover` $+\,\Omega$ and `ICover` in Figure 4(a). We observe a good impact. In Figure 4(b) we compare `ICover`$+\Omega$ and `QCover` which summarize the effect of our modifications: state inequation, sign analysis and omega places. We can see a substantial improvement.

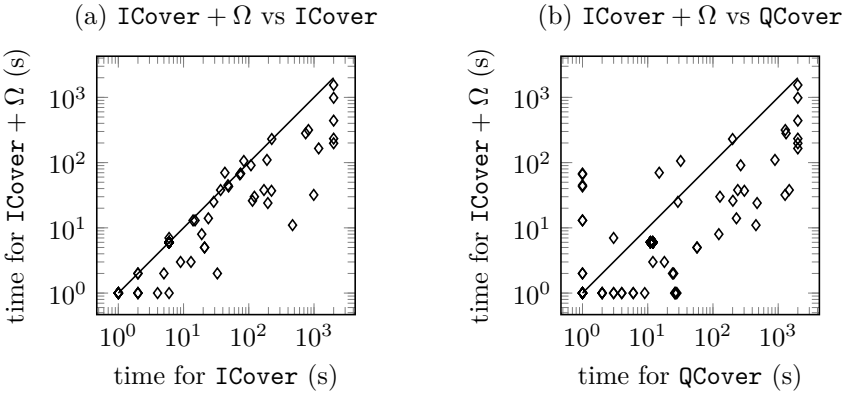

Figure 4: Experimental results for the effect of omega places

Even with only one place cut in a Petri net with a few tens of places, the effect can be important: the procedure can be accelerate by a factor of three in some cases. The Example 9.9 shows why we can have this effect.

Example 9.9. Consider the simple Petri net $\mathcal{N}$ in Figure 5 with the set of initial markings $Init = \{m_{init} \in \mathbb{N}^P \mid m_{init}(p_2) \leq 1\}$.

We want to know whether the target marking $m_{final} = 100\mathbf{e}_{p_2}$ is coverable. Without the pre-processing, the sequence of (B_k^I) will be for $0 < i < 100$, $B_i^I = \{(j\mathbf{e}_{p_1} + (100-j)\mathbf{e}_{p_2}) \in \mathbb{N}^P \mid j \leq i\}$. And finally $\downarrow Init \cap B_{99}^I \neq \emptyset$ will prove that m_{final} is coverable. The size of the last basis B_{99}^I is 99 despite a Petri net of a very limited size.

But our pre-processing will compute $\Omega = \Omega_0 = \{p_1\}$, and then the restricted Petri net will only contain one place p_2, and a simple transition that take one token in p_2 and put two tokens in that same place. The initial set will be

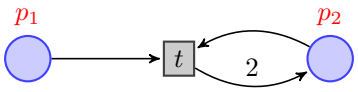

Figure 5: Simple Petri net that can be solved easily with the omega pre-processing

$Init|_Q = \downarrow \mathbf{e}_{p_2}$. And now the sequence will be for all $0 \leq i < 100$, $B_i^I|_Q = \{(i\mathbf{e}_{p_1} + (100 - i)\mathbf{e}_{p_2})\}$. All the bases will be of size one. □

This simple computation leads to a great effect despite being a simple forward analysis.

10. Conclusion

Petri nets have recently been used as low-level models for model-checking concurrent systems written in high-level programming languages [19, 18]. The original verification question on the concurrent program reduces to a coverability question on the resulting Petri net. We have proposed in this paper a family of simple coverability algorithms parametrized by downward-closed invariants. Additionally, we have presented two simple pre-processings that reduce the size of Petri nets while preserving the desired coverability information. As future work, we intend to look for classes of downward-closed invariants with a good tradeoff between precision and efficient membership.

References

[1] S. M. German, A. P. Sistla, Reasoning about systems with many processes, Journal of the Association for Computing Machinery 39 (3) (1992) 675–735.

[2] R. M. Karp, R. E. Miller, Parallel program schemata, Journal of Computer and System Sciences 3 (2) (1969) 147–195.

[3] R. J. Lipton, The reachability problem requires exponential space, Tech. Rep. 62, Yale University (1976).

[4] C. Rackoff, The covering and boundedness problems for vector addition systems, Theoretical Computer Science 6 (2) (1978) 223–231.

[5] A. Kaiser, D. Kroening, T. Wahl, A widening approach to multithreaded program verification, ACM Trans. Program. Lang. Syst. 36 (4) (2014) 14:1–14:29.

[6] J. Esparza, R. Ledesma-Garza, R. Majumdar, P. J. Meyer, F. Niksic, An SMT-based approach to coverability analysis, in: CAV, Springer, 2014, pp. 603–619.

[7] M. Blondin, A. Finkel, C. Haase, S. Haddad, Approaching the coverability problem continuously, in: TACAS, Springer, 2016, pp. 480–496.

[8] E. Fraca, S. Haddad, Complexity analysis of continuous Petri nets, Fundamenta Informaticae 137 (1) (2015) 1–28.

[9] P. A. Abdulla, K. Cerans, B. Jonsson, Y. Tsay, Algorithmic analysis of programs with well quasi-ordered domains, Information and Computation 160 (1-2) (2000) 109–127.

[10] T. Murata, State equation, controllability, and maximal matchings of Petri nets, IEEE Transactions on Automatic Control 22 (3) (1977) 412–416.

[11] P. Cousot, R. Cousot, Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints, in: POPL, ACM, 1977, pp. 238–252.

[12] M. Blondin, A. Finkel, C. Haase, S. Haddad, QCover: an efficient coverability verifier for discrete and continuous Petri nets, http://github.com/blondimi/qcover.

[13] T. Geffroy, J. Leroux, G. Sutre, ICover: Petri net coverability checker with invariant-based pruning, http://github.com/gsutre/icover.

[14] L. Recalde, E. Teruel, M. Silva, Autonomous continuous P/T systems, in: ICATPN, Springer, 1999, pp. 107–126.

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

[16] L. M. de Moura, N. Bjørner, Z3: an efficient SMT solver, in: TACAS, Springer, 2008, pp. 337–340.

[17] P. Ganty, Mist – A safety checker for petri nets and extensions, http://github.com/pierreganty/mist.

[18] E. D'Osualdo, J. Kochems, C. L. Ong, Automatic verification of Erlang-style concurrency, in: SAS, Springer, 2013, pp. 454–476.

[19] A. Donaldson, A. Kaiser, D. Kroening, T. Wahl, Symmetry-aware predicate abstraction for shared-variable concurrent programs, in: CAV, Springer, 2011, pp. 356–371.