Internal Report 2012–2013–13 August 2013

Universiteit Leiden
Opleiding Informatica

Coverability and Extended Petri Nets

Mathijs van de Nes

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Coverability and Extended Petri Nets

Mathijs van de Nes

August, 2013

Supervisors:
Dr. H.C.M. Kleijn
Dr. F.M. Spieksma

Leiden Institute of Advanced Computer Science
Mathematisch Instituut
Universiteit Leiden

Abstract

Coverability sets can be very useful in the analysis of Petri nets. For P/T nets, an algorithm for construction of these sets is known. For P/T nets extended with inhibitor arcs (PTI nets) no such algorithm exists. This thesis consists of three parts. The first part analyses the standard algorithm for coverability sets, and provides some methods to extract a more representative set from the result. In the second part, we prove that for every Petri net, a unique finite minimal coverability set exists. The last part focuses on PTI nets, and some ways to convert them back to P/T nets in such a way that coverability information is preserved.

Contents

1 Introduction

With Petri nets, one can model systems in which concurrency plays a role. Besides the intuitive graphical notation, P/T models also have a precise mathematical definition. This allows us to analyse the behaviour of the nets using all kinds of mathematical tools. Petri nets have been around since 1962 ([Pet62]). Lots of analysis has since then been done on them ([RR98], [Rei13] and [JvdAB+13]).

In this thesis, we will focus our attention on two types of Petri nets. The first is the Place/Transition net. Because of some modelling limitations, another class has risen. PTI nets – Place/Transition nets with Inhibitor arcs – are more powerful in terms of modelling power. This power comes at a price however. P/T nets have a property called monotonicity, where in a 'larger' state, every action which was possible in the smaller state is still possible. PTI nets do not have this property, and as a consequence some problems which are decidable for P/T nets are undecidable for PTI nets.

We will start with the analysis of the construction of *coverability sets* as described in the literature. We then add some ideas for choosing a representative of these sets. Next, we analyse the coverability set in a more abstract way, without the net structure. We step over to PTI nets in the last section. Here we first describe a method for checking a bound of the tokens in some of the places. We end with relating primitive PTI nets found in the literature to the P/T nets.

2 Preliminaries

2.1 Notations and Conventions

$\mathbb{N}$ is the set of all natural numbers, including zero. The set $\mathbb{N}_\omega := \mathbb{N} \cup \{\omega\}$ has the same operations as on $\mathbb{N}$, with in addition: for $n, k \in \mathbb{N}$ with $k \geq 1$: $\omega + \omega = \omega$, $\omega - \omega = \omega$, $0 \cdot \omega = 0$ and $\omega + n = \omega - n = k \cdot \omega = \omega$. The number ω can be viewed as an arbitrarily large positive integer. We also extend the ordering on $\mathbb{N}$ by defining $m < \omega$ for all $m \in \mathbb{N}$.

Let X be any finite set. A *string* s over X is a possibly empty finite sequence s of elements from X. Instead of writing $s = (s_1, s_2, \ldots, s_n)$ we may write $s = s_1 s_2 \ldots s_n$. The length of s is $|s| = n$.

2.2 Multisets

A *multiset* S over a finite set X is a function $S : X \to \mathbb{N}$, and an *extended multiset* is a function $S : X \to \mathbb{N}_\omega$. For $x \in X$ we say $x \in S$ iff $S(x) > 0$. For two (extended) multisets S, S' over X we say $S \leq S'$ or S' *covers* S if $\forall x \in X : S(x) \leq S'(x)$, we write $S < S'$ if $S \leq S'$ and $S \neq S'$.

The addition of two (extended) multisets S and S' over X is given by $(S + S')(x) := S(x) + S'(x)$ for all $x \in X$. Subtraction is given as $(S - S')(x) := \max\{0, S(x) - S'(x)\}$. Finally we define multiplication of a (extended) multiset with a constant $n \in \mathbb{N}$ as $(n \cdot S)(x) := n \cdot S(x)$.

When there is a natural enumeration given on X, i.e. X consists of the distinct elements $x_1, x_2, \ldots, x_n$, an (extended) multiset over X can also be described by a vector $(s(1), s(2), \ldots, s(n))$ where $s(i) = S(x_i)$.

2.3 Dickson's Lemma

Dickson's Lemma is a lemma which will be used frequently throughout this paper. The original version of the lemma from [Dic13] is quite general, so we provide a short proof for the version stated here.

Lemma 2.1 (Dickson)**.** *Every infinite sequence of vectors in $\mathbb{N}_\omega^n$ has an infinite non-decreasing subsequence.*

Proof. We use induction on n.

Let $n = 1$. Consider a sequence $(a_i)_{i=0}^{\infty}$, with $a_i \in \mathbb{N}$. If some a_i occurs infinitely many times in this sequence, we can take that number and we are done. Suppose there is no a_i that occurs infinitely many times. Thus in particular, ω occurs a finite number of times. Take any $a_i \neq \omega$. Then there is $j > i$ such that $a_j \geq a_i$, since there are only finitely many elements smaller than a_i. Thus, we can pick the first $a_i \neq \omega$, and then each time find larger elements of the sequence to construct our subsequence.

Let $n > 1$ and assume the statement holds for all $k < n$. Let $(a_i)_{i=0}^{\infty}$ with $a_i \in \mathbb{N}^n$ be an infinite sequence. We can write each element $a_i = (a(1), \ldots, a(n))$ with $b_i = a_i(1) \in \mathbb{N}$ and $c_i = (a_i(2), \ldots, a_i(n)) \in \mathbb{N}^{n-1}$. By the induction hypotheses, we can construct an infinite non-decreasing sequence on the c_i part. Thus we get a sequence $(a_{i_j})_{j=0}^{\infty}$ for which $c_{i_j} \leq c_{i_{j+1}}$ holds for all j. This sequence contains a non-decreasing subsequence on the b_{i_j}. This third sequence is non-decreasing on all b_i, and all c_i, thus an infinite non-decreasing subsequence of $(a_i)_{i=0}^{\infty}$.

Combining these two facts, we have proven the lemma. $\square$

3 Place/Transition nets

3.1 P/T nets

A place/transition net, or P/T net for short, is a basic type Petri net that has been studied extensively. A lot about these nets is known, and as such it is a good starting point for further study.

A P/T net is a 4-tuple $\mathcal{PT} = (P, T, W, M_0)$. P and T are disjoint finite sets, W is a multiset over $(P \times T) \cup (T \times P)$, and M_0 a multiset over P. We assume that for every $t \in T$ there is at least one $p \in P$ with $W(p, t) > 0$ or $W(t, p) > 0$.

The elements of P are the *places* of the net, and T its *transitions*. W connects the places and transitions by giving the *weight* of an arc between them. A *marking* M is a multiset over the places, where $M(p)$ describes how many tokens place p contains in that marking. M_0 is the initial marking.

We will draw a P/T as in Figure 1. Here places are represented as circles and transitions are represented as squares. For $p \in P$ and $t \in T$, if $W(p, t) > 1$ then we draw an arc from p to t with the weight next to it, if $W(p, t) = 1$ we omit the weight, and if $W(p, t) = 0$ we omit the arc entirely. Lastly, the marking M_0 is represented by placing $M_0(p)$ black dots or the value of $M_0(p)$ in the circle corresponding to place p.

We define ${}^\bullet t$ and $t^\bullet$ as the multisets over P given by ${}^\bullet t(p) := W(p, t)$, and $t^\bullet(p) := W(t, p)$. These are the *pre-* and *postsets* of a transition, and determine when a transition is enabled, and what its effect will be. A transition t can *fire* in a marking M, or is *enabled* in M, if ${}^\bullet t \leq M$, meaning all places with arcs

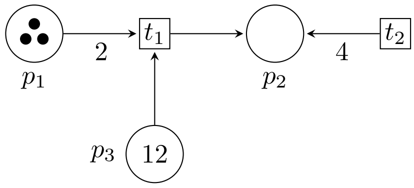


Figure 1: A P/T net

leading to t have at least as many tokens as the weights on the arcs. If t fires at M, then the resulting marking is a multiset $M' = M - {}^\bullet t + t^\bullet$. We write $M[t\rangle M'$.

If M is an *extended marking*, i.e. a marking where some places may contain ω number of tokens, the same definition for enabledness and effect is applied, keeping in mind the arithmetic rules of Section 2.1.

3.2 PTI nets

A PTI net is an extension to the P/T net defined before. The I in PTI comes from the term *inhibitor arc* that will be used in this model.

A PTI net is a 5-tuple $\mathcal{PTI} = (P, T, W, M_0, I)$ where (P, T, W, M_0) is a P/T net and I an *extended* multiset over $P \times T$. If $I(p,t) \in \mathbb{N}$ then p is an *inhibitor place* of $\mathcal{PTI}$. The intuitively meaning is that t cannot fire (is inhibited) if p contains more than $I(p,t)$ tokens. If $I(p,t) = \omega$, then p does not inhibit t. In a graphical representation, an inhibitor arc is drawn with a small circle as its head and a weight next to it. If $I(p,t) = 0$ then we put no weight next to the arc, if $I(p,t) = \omega$ we omit the arc. See also Figure 2. Note that if for all p and t, $I(p,t) = \omega$, then $\mathcal{PTI}$ is a plain P/T net.

h

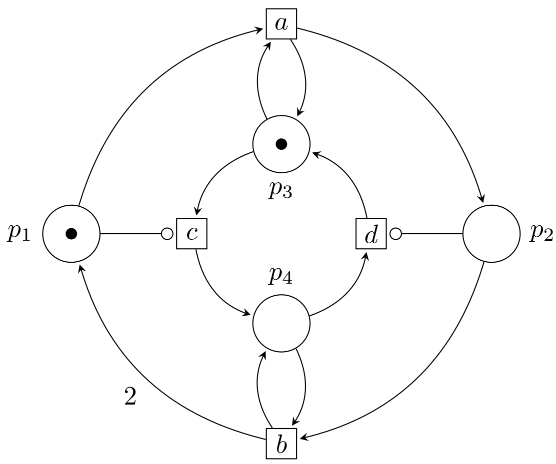


Figure 2: PTI net with two inhibitor arcs, from [vdVKK11].

For every $t \in T$, we define ${}^\circ t$ as the extended multiset over P where ${}^\circ t(p) = I(p,t)$. In a PTI net, t is enabled in a marking M when ${}^\bullet t \leq M \leq {}^\circ t$. If t is enabled and fires at M leading to a new marking M', this is written as $M[t\rangle M'$

where $M' = M - {}^{\bullet}t + t^{\bullet}$. This firing rule corresponds to the rule of P/T nets, where ${}^{\circ}t(p) = \omega$ for all $p \in P$.

3.3 Monotonicity

Monotonicity is an important aspect of the firing in P/T nets. If M is some marking of $\mathcal{PT}$, and $M[t\rangle M'$, then for all multisets E over P, it will hold that $(M + E)[t\rangle(M' + E)$. Increasing the number of tokens in a marking can only increase the number of enabled transitions, which is why we call this behaviour monotonic.

PTI nets do not have this property. It may be possible to disable some transition by adding tokens. It could also be possible to enable some transition by removing tokens. In the case of Figure 2, removing the token from p_1 would enable transition c.

4 Coverability in nets

4.1 Reachability

Definition 4.1. Let $\mathcal{PTI} = (P, T, W, M_0, I)$ be a PTI net. A marking M' is *reachable* from a marking M if there is a (possibly empty) sequence $(t_1, t_2, \ldots, t_n)$ of transitions such that $M[t_1\rangle M_1[t_2\rangle \cdots [t_n\rangle M$. A marking M' is reachable in $\mathcal{PTI}$ if it is reachable from M_0.

The set R of all reachable markings of $\mathcal{PTI}$, also called its *reachability set*, holds a lot of information. This set need not be finite. In Figure 3 we have drawn a fragment of the reachability set of the net from Figure 2 in the form of a labelled directed graph (R, A), with initial node M_0, set of nodes R and labelled arcs (M, t, M') iff $M[t\rangle M'$ in $\mathcal{PTI}$.

$$\begin{array}{c}
(1,0,1,0) \\
\big\downarrow a \\
(0,1,1,0) \\
\big\downarrow c \\
(0,1,0,1) \\
\big\downarrow b \\
(2,0,0,1) \\
\big\downarrow d \\
(2,0,1,0) \\
\big\downarrow a \\
(1,1,1,0) \\
\big\downarrow a \\
(0,2,1,0) \\
\Big\downarrow \\
\cdots
\end{array}$$

Figure 3: Part of the infinite reachability graph for the PTI net of Figure 2

The *reachability problem* is deciding for a given marking M, whether it is reachable in a specific net. [May81] shows that this is decidable for P/T nets, i.e., there exists an algorithm which gives a correct answer to this question in a finite ammount of time.

Definition 4.2. Let $\mathcal{PTI} = (P, T, W, M_0, I)$ be a PTI net. A place $p \in P$ is bounded by $b \in \mathbb{N}$ if for all markings M reachable from M_0, it holds that $M(p) \leq b$. We say that $\mathcal{PTI}$ is bounded if all places in P are bounded.

An immediate result is that the reachable set R of a net $\mathcal{PTI}$ is finite iff $\mathcal{PTI}$ is bounded.

4.2 Coverability Set

A coverability set is a way to represent the reachability set.

Definition 4.3. Let $\mathcal{PTI}$ be a PTI net and R its reachability set. A set V is a *coverability set* of $\mathcal{PTI}$ if it is a set of extended markings for which the following two properties hold:

- For any $M \in R$, there is a $v \in V$ such that $M \leq v$;
- $\forall v \in V$, $\forall b \in \mathbb{N}$ there exists an $M \in R$ such that for all places p of $\mathcal{PTI}$, either $M(p) = v(p)$, or both $M(p) \geq b$ and $v(p) = \omega$.

The first property is to make sure that it actually covers R, as each element of the reachability set must be dominated by an element from V. The second property states that all elements of V correspond to reachable markings with ω entries representing simultaneously unbounded places.

4.3 Coverability Trees for P/T nets

The standard approach ([KM69]) to construct a coverability set for P/T nets is to create a coverability tree (CT).

Definition 4.4. A CT is a 4-tuple (V, A, μ, v_0). V is the set of vertices, with $v_0 \in V$ being the root. A is the set of labelled arrows. μ is a function that assigns an extended marking to every node as a label.

To create such a tree, Algorithm 1 can be used.

In this algorithm, the way in which v is chosen on line 7 is not specified. Breadth-first search or depth-first search are two of the possibilities. Different choice functions may end up creating different trees (see Example 4.5), but it will always ([KK12]) have the following properties:

- The coverability tree is finite;
- The labels of the nodes form a coverability set of $\mathcal{PT}$;
- All firing sequences of $\mathcal{PT}$ are represented in CT. For any sequence of transitions such that $M_0[t_1\rangle M_1 \cdots M_{n-1}[t_n\rangle M_n$, there exists arcs $v_{i-1} \xrightarrow{t_i} w_i$ such that for $i \in \{1, \ldots, n-1\}$, $\mu[w_i] = \mu[v_i]$ and $M_i \leq \mu[v_i]$, and also $M_n \leq \mu[w_n]$.

Algorithm 1: Construct a CT for a P/T net

Input: P/T net $\mathcal{PT} = (P, T, W, M_0)$
Result: Tree $CT = (V, A, \mu, v_0)$

$v_0 \leftarrow$ new node
$V \leftarrow \{v_0\}$
$A \leftarrow \emptyset$
$\mu[v_0] \leftarrow M_0$
UNPROCESSED $\leftarrow \{v_0\}$
while UNPROCESSED $\neq \emptyset$ **do**
  $v \leftarrow$ element of UNPROCESSED
  if $\forall u \in V \setminus$ UNPROCESSED, $\mu[u] \neq \mu[v]$ **then**
    foreach $t : \mu[v][t\rangle M'$ **do**
      $w \leftarrow$ new node
      $V \leftarrow V \cup \{w\}$
      $A \leftarrow A \cup \{v \xrightarrow{t} w\}$
      UNPROCESSED $\leftarrow$ UNPROCESSED $\cup \{w\}$
      `//` $x \rightsquigarrow_A y$ `means a path from` x `to` y `with arrows in` A
      if $\exists u \rightsquigarrow_A v$ *with* $\mu[u] < M'$ **then**
        foreach $p \in P$ **do**
          if $\mu[u](p) < M'(p)$ **then** $\mu[w](p) = \omega$
          else $\mu[w](p) = M'(p)$
        end
      else
        $\mu[w] = M'$
      end
    end
    UNPROCESSED $\leftarrow$ UNPROCESSED $\setminus \{v\}$
  end
end

The algorithm works by using the monotonicity property. In lines 14–18, the algorithm introduces an ω in markings when growth is found. If growth is found on a path $u \rightsquigarrow_A v$, we can repeat the transitions of that path arbitrarily many times to introduce as many tokens as we want on some places.

Example 4.5. Consider the P/T net in Figure 4a. In Figure 4b and 4c, a coverability tree is drawn, using respectively a breadth-first search and depth-first search in the algorithm. The depth-first search gives us the coverability set $\{(1,0,0),(0,1,0),\ (0,0,1),(0,\omega,0),(0,\omega,\omega)\}$. We can conclude that the second and third places are both (simultaneously) unbounded. The coverability set created by using a breadth-first search is different from the previous one, as it also includes the elements $(0,2,0)$ and $(0,\omega,1)$.

4.4 Coverability Graphs for P/T nets

In this section we explore constructing a graph rather than a tree.

Definition 4.6. A *Coverability Graph* (CG) is a 4-tuple (V, A, μ, v_0). V is the set of vertices, with $v_0 \in V$ being the initial node. A is the set of labelled arrows. μ is a function that assigns an extended marking to every node as a label.

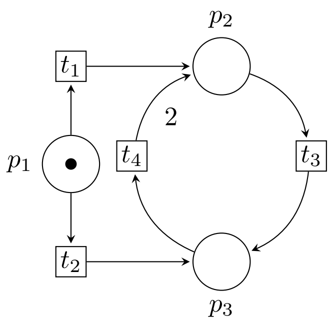
Algorithm 2 constructs such a graph for a given P/T net $\mathcal{PT}$. As with the coverability tree, line 7 of the algorithm does not specify the order on how nodes are expanded.

There are a few differences between Algorithms 1 and 2:

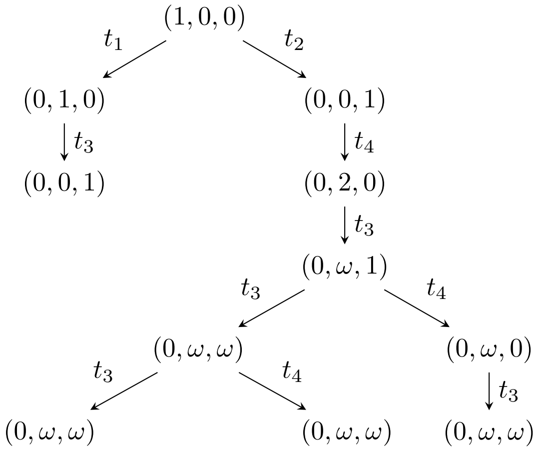
- The tree may have multiple nodes with the same labelling. The graph will not have this;
- When constructing the graph, each iteration you will need to search the graph whether a label already exists. In the tree, a similar search takes place in deciding whether or not there exists a processed node with the same label;
- In the tree, it is sufficient to keep track of a node's single parent, and walk backwards up the only path to the root of the tree to find its predecessors. In a graph, a node can have multiple parents. When implementing you could keep a list of parents in each node, and marking them as 'visited' when traversing in order to avoid duplicate work;
- Firing sequences are easier to find in the graph than in the tree, since we never have to jump from a node to a smaller one.

The advantages of the graph include it being a more intuitive representation, and consuming less space. The trade-off is, that the search for a smaller ancestor is more complex. Where in a tree a node only has one reverse path back to the root, the search in line 10 of Algorithm 2 could branch.

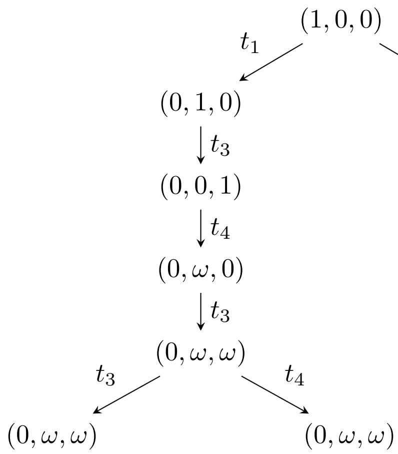
Example 4.7. Consider again the P/T net from Example 4.5 depicted in Figure 4a. The coverability graph as constructed by Algorithm 2 is shown in Figure 5. Nodes to expand were selected using a depth-first search. Here, all nodes with label $(0,\omega,\omega)$ are now merged into one single node. The coverability set is the same as the one extracted from Figure 4c.



(a) Net



(b) CT using BFS



(c) CT using DFS

Figure 4: A small unbounded P/T net and its coverability tree

Algorithm 2: Construct a CG for a P/T net

Input: P/T net $\mathcal{PT} = (P, T, W, M_0)$
Result: Graph $CG = (V, A, \mu, v_0)$

$v_0 \leftarrow$ new node
$V \leftarrow \{v_0\}$
$A \leftarrow \emptyset$
$\mu[v_0] \leftarrow M_0$
UNPROCESSED $\leftarrow \{v_0\}$
while UNPROCESSED $\neq \emptyset$ **do**
    $v \leftarrow$ element of UNPROCESSED
    foreach $t : \mu[v][t\rangle M'$ **do**
        $M \leftarrow$ new marking
        `//` $x \rightsquigarrow_A y$ `means a path from` x `to` y `with arrows in` A
        if $\exists u \rightsquigarrow_A v$ *with* $\mu[u] < M'$ **then**
            foreach $p \in P$ **do**
                if $\mu[u](p) < M'(p)$ **then** $\mu[w](p) = \omega$
                else $\mu[w](p) = M'(p)$
            end
        else
            $M = M'$
        end
        if $\nexists w \in V : \mu[w] = M$ **then**
            $w \leftarrow$ new node
            $V \leftarrow V \cup \{w\}$
            $\mu[w] = M$
            UNPROCESSED $\leftarrow$ UNPROCESSED $\cup \{w\}$
        else
            Select w with $\mu[w] = M$
        end
        $A \leftarrow A \cup \{v \xrightarrow{t} w\}$
    end
    UNPROCESSED $\leftarrow$ UNPROCESSED $\setminus \{v\}$
end

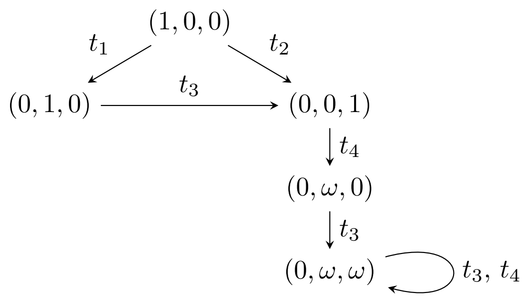


Figure 5: Coverability graph for the P/T net in Figure 4a

5 Coverability representatives

The coverability sets obtained from Algorithms 1 and 2 for a P/T net $\mathcal{PT}$ are not the only coverability set for $\mathcal{PT}$. Moreover, some of their elements may be superfluous. In this section we compare some transformations on the coverability set to get a smaller but more representative coverability set.

5.1 Maximal elements

Definition 5.1. Let V be a coverability set of a P/T net. The set of *maximal elements* of V is

$$\lceil V \rceil := \{v \in V : \nexists s \in V \text{ with } v < s\}$$

As proven later in Section 6, it turns out that taking the maximal elements of a finite coverability set leaves us with a smaller set which still covers the reachable markings of the P/T net. This set is the smallest possible covering set. A disadvantage is however that it provides too little information, as can be seen in the following example.

Example 5.2. Consider the P/T net in Figure 6. It is easy to see that the reachability set for this net is $R = \{(1,0),(0,2),(0,3)\}$, which is also a finite coverability set for the net. The collection of maximal elements of this set is $\lceil R \rceil = \{(1,0),(0,3)\}$, which is also a coverability set. However, the marking $(0,2)$ is lost in the conversion. It is discarded since $(0,3)$ is larger, but this marking is not reachable from $(0,2)$.

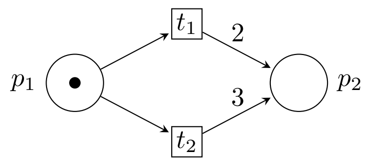


Figure 6

5.2 Almost maximal elements

Another way for creating a representative set is the following:

Definition 5.3. Let V be some coverability set. We define the set $\widetilde{V}$, the set of *almost maximal* elements of V, as:

$$\widetilde{V} := \Big\{ v \in V : \forall y \in V \text{ with } y > v, \forall p \in P : \Big(y(p) = \omega \Rightarrow v(p) = \omega \Big) \Big\}$$

With this definition we also keep smaller elements provided that they are only smaller on places that do not contain ω. Elements in $V \setminus \widetilde{V}$ are a special case of markings that can grow indefinitely.

Example 5.4. Consider again the P/T net from Figure 6. In this example, we have that $\widetilde{R} = R$.

Keeping all elements that are almost maximal is the most promising method for reducing a coverability set, as can be seen in the comparison later on.

5.3 Augmented marking

Another way to get a representative of a coverability set is via augmented markings. First, we define Parikh vectors:

Definition 5.5. Let w be a string over $T = \{t_1, \ldots, t_n\}$. For $t \in T$, let $\#(t, w)$ be the number of occurrences of t in w. We then define the *Parikh vector* of w as

$$\Psi(w) = (\#(t_1, w), \ldots, \#(t_n, w))$$

With this, we can define an augmented marking as follows.

Definition 5.6. Let $\mathcal{PT} = (P, T, W, M_0)$ be a P/T net. Let d be the sum of the sizes of P and T. For any marking sequence $X = M_0[t_1\rangle \cdots [t_m\rangle M$ we define the *augmented marking* $M_X \in \mathbb{N}_\omega^d$ denoted by

$$M_X = M \times \Psi(t_1 \cdots t_m).$$

For readability, we write the vectors of the form $(\cdots | \cdots)$, where the marking is in left part and the Parikh vector is on the right.

Example 5.7. Consider the net from Figure 6. The set of all augmented reachable markings is $R_a = \{(1, 0|0, 0), (0, 2|1, 0), (0, 3|0, 1)\}$. It holds that $\lceil R_a \rceil = R_a$, so we see that $(0, 2|1, 0)$ is preserved.

The reason for this method is that it retains information on how markings are reached from M_0. This way, if some marking M is smaller than some M', but M' is not reachable from M, this method will not discard M directly.

We can easily modify Algorithm 1 to keep track of the Parikh vector during the construction. However, the coverability graph of Algorithm 2 is not suited for modification. Some nodes can have multiple paths leading to it, which makes it difficult to assign a single Parikh vector to such nodes.

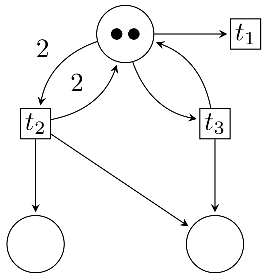
5.4 Comparison

Consider the P/T net of Figure 7a. A possible coverability tree is shown in Figure 7b. After calculating the Parikh vectors of all extended markings in the tree, we can find which elements are maximal in each of the proposed ways. This is depicted in Table 1.

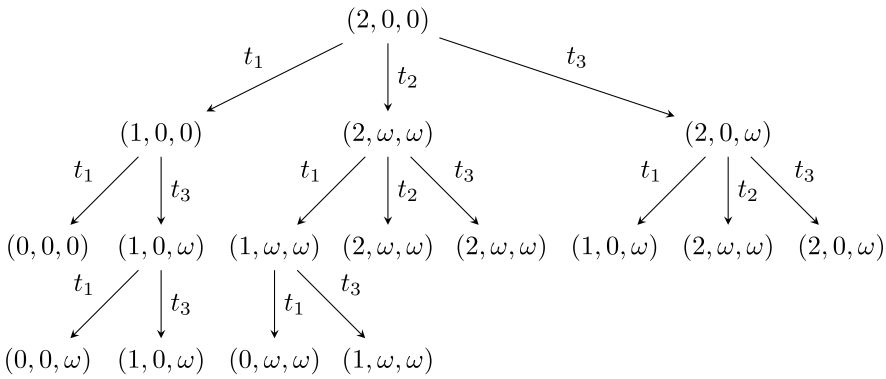
If we disregard the Parikh-vector, we find the following sets of maximal elements:

- $\lceil V \rceil = \{(2, \omega, \omega)\}$
- $\widetilde{V} = \{(0, \omega, \omega), (1, \omega, \omega), (2, \omega, \omega)\}$
- $\lceil V_a \rceil = \{(0, \omega, \omega), (1, \omega, \omega), (2, \omega, \omega), (0, 0, \omega), (1, 0, \omega), (2, 0, \omega)\}$

Notice that $\lceil V \rceil \subseteq \widetilde{V} \subseteq \lceil V_a \rceil$ holds for this case. Another thing to note is that the set $\lceil V_a \rceil$ is not uniquely determined by the net. It could be that Algorithm 1 would also explore the firing sequence $t_3 t_2 t_3$ when using some other search method, and thus obtain the marking $(2, \omega, \omega|0, 1, 2)$. This marking is larger than $(2, 0, \omega|0, 0, 2)$, which means $(2, 0, \omega)$ would not be maximal in the new V_a any more.



(a)



(b)

Figure 7: A P/T net and a coverability tree

M			$\Psi(t_1, t_2, t_3)$			element of		
p_1	p_2	p_3	t_1	t_2	t_3	$\lceil V_a \rceil$	$\widetilde{V}$	$\lceil V \rceil$
2	0	0	0	0	0			
1	0	0	1	0	0			
0	0	0	2	0	0			
1	0	ω	1	0	1			
0	0	ω	2	0	1	✓		
1	0	ω	1	0	2	✓		
2	ω	ω	0	1	0		✓	✓
1	ω	ω	1	1	0		✓	
0	ω	ω	2	1	0	✓	✓	
1	ω	ω	1	1	1	✓	✓	
2	ω	ω	0	2	0	✓	✓	✓
2	ω	ω	0	1	1	✓	✓	✓
2	0	ω	0	0	1			
1	0	ω	1	0	1			
2	0	ω	0	1	1			
2	0	ω	0	0	2	✓		

Table 1: Coverability sets of Figure 7a

In this example we discover that augmented markings and the almost maximal elements method both keep more markings than the maximal elements method. However, which elements are kept using augmented markings depends on the execution order of the algorithm used, which makes it a less suitable candidate. The almost maximal elements method provides a method to keep more relevant information than the maximal elements method, while also being more reliable than the augmented markings.

6 General coverability

As we will see in Section 7.2, coverability for PTI nets is not a trivial thing. Our goal in this section is to find an alternative method for constructing coverability sets, as Algorithms 1 and 2 only work for P/T nets. Abstracting away from the net structure, this section will focus on creating a coverability set from just a set of vectors.

6.1 Covering sets

Definition 6.1. (Generalisation of Definition 4.3.) Let $R \subseteq \mathbb{N}_\omega^n$. A set $V \subseteq \mathbb{N}_\omega^n$ is a *covering set* for R if it has the following two properties:

- For any $r \in R$, there is a $v \in V$ such that $r \leq v$;
- $\forall v \in V$, $\forall b \in \mathbb{N}$ there exists an $r \in R$ such that $\forall k \in \{1, \dots, n\}$, either $v(k) = r(k)$, or both $v(k) = \omega$ and $r(k) \geq b$.

We write: $R \sqsubseteq V$ and say that V covers R.

The first lemma shows a fairly trivial property, but it is important to observe.

Lemma 6.2. *For any $R \subseteq \mathbb{N}_\omega^n$, $R \sqsubseteq R$.*

Proof. For every $r \in R$, the element covers itself ($r \leq r$) and is an 'approximation' of itself ($\forall k, r(k) = r(k)$). ☐

The previous shows us that '$\sqsubseteq$' is reflexive. By the next lemma it is also transitive, showing that it actually is a pre-order.

Lemma 6.3. *Let $R_1, R_2, R_3 \subseteq \mathbb{N}_\omega^n$, such that $R_1 \sqsubseteq R_2$ and $R_2 \sqsubseteq R_3$. Then $R_1 \sqsubseteq R_3$.*

Proof. Let $r_1 \in R_1$. Since $R_1 \sqsubseteq R_2$, there exists an $r_2 \in R_2$ such that $r_1 \leq r_2$. We also have $R_2 \sqsubseteq R_3$, and as such there also exists a $r_3 \in R_3$ with $r_2 \leq r_3$. So we have that $r_1 \leq r_2 \leq r_3$, and thus that the first property of Definition 6.1 holds.

Let $r_3 \in R_3$, and $b \in \mathbb{N}$ be arbitrary. Then there exists a $r_2 \in R_2$ such that, $\forall k \in \{1, \dots, n\}$, either $r_2(k) = r_3(k)$ or both $r_3(k) = \omega$ and $r_2(k) \geq b$. For this r_2 and for all b', we can find a $r_1 \in R$ in the same way, so let $b' = b$. Let $k \in \{1, \dots, n\}$. If $r_3(k) < \omega$, then $r_3(k) = r_2(k) = r_1(k)$. If $r_3(k) = \omega$, then $r_2(k) \geq b$. This means that $r_1(k) \geq b$ will also hold. Repeating this for all k shows that the second property of Definition 6.1 holds. ☐

Our goal is to construct a finite covering set from a set of vectors. To help us with this, we take a look at the maximal elements.

Definition 6.4. (Generalisation of Definition 5.1.) Let $R \subseteq \mathbb{N}_\omega^n$. The set of *maximal elements* of R is

$$\lceil R \rceil := \{r \in R : \nexists s \in R \text{ with } r < s\}$$

Lemma 6.5. *Let $R \subseteq \mathbb{N}_\omega^n$. Then $\lceil R \rceil$ is finite.*

Proof. Suppose $\lceil R \rceil$ is infinite. There exists a sequence $(r_i)_{i=0}^{\infty}$ with $r_i \in \lceil R \rceil$, such that $r_i \neq r_j$ for $i \neq j$. Dickson's Lemma tells us that there is an infinite non-decreasing subsequence. Since all elements are pairwise different, this subsequence is strictly increasing, in contradiction with the definition of $\lceil R \rceil$. $\square$

The following lemma shows that for finite covering sets, taking just the maximal elements property of covering.

Lemma 6.6. *Let $V \subseteq \mathbb{N}_\omega^n$ be a finite covering set of $R \subseteq \mathbb{N}_\omega^n$. Then $R \sqsubseteq \lceil V \rceil$.*

Proof. First we consider $R = \emptyset$. V is a covering set of R, and due to the second property of Definition 6.1, $V = \emptyset$. Hence $\lceil V \rceil = \emptyset$, and $R \sqsubseteq \lceil V \rceil$.

Now consider the case where $R \neq \emptyset$.

Let $r \in R$. Since V is finite, the set $X_r = \{x \in V : x \geq r\}$ is finite and contains at least one element. Hence there will be at least one maximal element of V in X_r, i.e., there exists an $x \in X_r$, for which $x \in \lceil V \rceil$. Combining this with the fact that $x \geq r$ shows us that for r there is an element covering it in $\lceil V \rceil$.

Let $v \in \lceil V \rceil$. We have $\lceil V \rceil \subseteq V$, thus $v \in V$. Since the second criterion of Definition 6.1 holds for any individual element of V, it especially holds for v. $\square$

Example 6.7. Consider $R = \{(0, m) \mid m \in \mathbb{N}\} \subseteq \mathbb{N}^2$. An example of a finite covering set of R is $\{(0, \omega)\}$. However, $\lceil R \rceil = \emptyset$ does not cover R.

The fact that in the example, $\lceil R \rceil$ did not cover R is explained by the following lemma.

Lemma 6.8. *Let $R \subseteq \mathbb{N}^n$. R is finite $\Leftrightarrow R \sqsubseteq \lceil R \rceil$.*

Proof. "$\Rightarrow$": This follows directly by combining Lemma 6.2 and 6.6.

"$\Leftarrow$": Each $r \in R$ is covered by some element in $\lceil R \rceil$. By Lemma 6.5 we know that $\lceil R \rceil$ is finite. For each $s \in \lceil R \rceil$ we define $R_s = \{r \in R : r \leq s\}$. R_s is a finite set. More importantly, $R \subseteq \bigcup_{s \in \lceil R \rceil} R_s$. Since the latter set is a finite union of finite sets, R is thus finite as well. $\square$

If $\lceil R \rceil$ does not cover R then it must be the case that there are increasing sequences in R, without a maximum in R. We take a look at the limit of such a sequence. The work on limits is similar to that from [VH12], except that we consider here general vectors over $\mathbb{N}_\omega$ instead of markings.

6.2 Limits and closure

In this section we use limits of sequences, to extract more information from a set.

Definition 6.9. Let $(M_i)_{i=0}^{\infty}$ be an infinite non-decreasing sequence over $\mathbb{N}_\omega^n$. We define the *limit* $M = \lim_{i\to\infty} M_i$ as

$$M(k) = \begin{cases} m \in \mathbb{N} & \text{if } \exists I \text{ s.t. } \forall i \geq I,\ M_i(k) = m \\ \omega & \text{otherwise} \end{cases}$$

Note that the M_i do not have to be distinct. The sequence with a tail consisting of $M, M, M, \ldots$ has a limit equal to M. Also note that the elements in the sequence may contain ω components. Defining the limit this way makes sense. Since the elements of a sequence are all non-decreasing, the ordering makes sure that this is also coordinate-wise the case. If the sequence contains a tail for which a coordinate never changes, that element will be chosen. If this is not the case, there will always be an element at some point in the sequence which is larger, in which case ω is used.

Definition 6.10. Let $R \subseteq \mathbb{N}_\omega^n$. We define the *closure* of R as

$$\overline{R} = \{r \in \mathbb{N}_\omega^n \mid \exists r_0 \leq r_1 \leq r_2 \leq \ldots,\ \text{with } r_i \in R \text{ such that } r = \lim_{i\to\infty} r_i\}.$$

Calling $\overline{R}$ a closure must be justified. The next two lemmas take care of this.

Lemma 6.11. *Let $R \subseteq \mathbb{N}_\omega^n$. Then $R \subseteq \overline{R}$. Furthermore, if R is finite, then $R = \overline{R}$.*

Proof. Let $r \in R$. Then $r, r, r, \ldots$ is a non-decreasing infinite sequence over R with limit r. Thus $r \in \overline{R}$.

Let R be finite and $r \in \overline{R}$. Then there is a non-decreasing infinite sequence $(r_i)_{i=0}^{\infty}$ over R such that $\lim_{i\to\infty} r_i = r$. Since R is finite, there are only finitely many different r_i in this sequence. There exists an $I \in \mathbb{N}$, such that $r_i = r_{i+1}$ for all $i \geq I$. Thus $r = r_I$, and thus $r \in R$. $\square$

Lemma 6.12. *Let $R \subseteq \mathbb{N}_\omega^n$. Then $\overline{\overline{R}} = \overline{R}$.*

Proof. "$\supseteq$": This follows directly from Lemma 6.11.

"$\subseteq$": Take an $r \in \overline{\overline{R}}$. Then there exists an non-decreasing infinite sequence $(r_i)_{i=0}^{\infty}$ over $\overline{R}$ such that $\lim_{i\to\infty} r_i = r$. For each r_i from this sequence, there exists an non-decreasing infinite sequence $(r_{i,j})_{j=0}^{\infty}$ over R such that $\lim_{j\to\infty} r_{i,j} = r_i$.

We are going to construct a non-decreasing infinite sequence $(r'_i)_{i=0}^{\infty}$ over R such that $\lim_{i\to\infty} r'_i = r$.

Let $k \in \{1, \ldots, n\}$ and i fixed. Then if $r_i(k) < \infty$, set $r'_i(k) = r_i(k)$, as there surely will be a j such that $r_{i,j}(k) = r_i(k)$. Otherwise, if $r_i(k) = \omega$, choose a j such that $r_{i,j}(k) > i$ and $r_{i,j}(k) > r'_{i-1}(k)$, and set $r'_i(k) = r_{i,j}(k)$. Doing this for all k and i, we have a new non-decreasing sequence $(r'_i)_{i=0}^{\infty}$.

Let $k \in \{1, \ldots, n\}$. If $r_i(k) < \omega$ for all i, then $r'_i(k) = r_i(k)$ for $i > 0$. If for some I, $r_I(k) = \omega$, then for each $i > I$, $r'_i(k) \geq i$ and will thus have the limit ω. This shows that $r = \lim_{i\to\infty} r_i = \lim_{i\to\infty} r'_i$, and thus that $r \in \overline{R}$. $\square$

6.3 Minimal unique covering set

The following theorem describes how we obtain a finite covering set.

Theorem 6.13. *Let $R \subseteq \mathbb{N}_\omega^n$. Then $\lceil \overline{R} \rceil$ is a finite covering set of R.*

Proof. The finiteness of $\lceil \overline{R} \rceil$ follows from Lemma 6.5.

Let $r \in \lceil \overline{R} \rceil$. Since r is the limit of an infinite non-decreasing sequence, it satisfies the properties of Definition 6.9. This property coincides with the second property of Definition 6.1, thus $\lceil \overline{R} \rceil$ also satisfies this property.

To prove that $\lceil \overline{R} \rceil$ satisfies the first property of Definition 6.1, we take an $r \in R$. Suppose there is no $x \in \lceil \overline{R} \rceil$ such that $r \leq x$. For every $y \in \overline{R}$ with $y > r$, we can find another $y' \in \overline{R}$ such that $y' > y$. We can now construct a strictly increasing infinite sequence $(y_i)_{i=0}^{\infty}$ over $\overline{R}$ with all $y_i > r$. Let $y = \lim_{i \to \infty} y_i$. Because of Lemma 6.12, we have that $y \in \overline{R}$, and since all elements are strictly increasing, we have that at least one new ω will be introduced. After at most n iterations of this process, the limit which we then have obtained will be the vector $z = (\omega, \ldots, \omega)$. Since $r \leq z$ it means that z is not an element of $\lceil \overline{R} \rceil$. But z is the largest element of $\mathbb{N}_\omega^n$, thus $z \in \lceil \overline{R} \rceil$, which leads to a contradiction. Thus for r there is an element that covers it in $\lceil \overline{R} \rceil$. □

The following theorem shows that taking the maximal elements of two finite covering sets have the same maximal elements.

Theorem 6.14. *Let V, W be finite covering sets of $R \subseteq \mathbb{N}_\omega^n$. Then $\lceil V \rceil = \lceil W \rceil$.*

Proof. For all $s \in \mathbb{N}_\omega^n$ we define R_s as $\{r \in R \mid r \leq s\}$.

Let $v \in \lceil V \rceil$, and consider R_v.

The first case we examine is where R_v is finite. This means that $v \in R$ and $\lceil R_v \rceil = \{v\}$. The former implies that there exists an $w \in \lceil W \rceil$, with $w \geq v$.

Suppose R_w is finite. Then w is an element of R, as above. This means there exists a $v' \in \lceil V \rceil$ such that $v' \geq w \geq v$. Since v is maximal in V, it holds that $v' = w = v$, and thus $v \in \lceil W \rceil$. This proves the case for both R_v and R_w finite.

Suppose R_w is infinite. Then w has an ω on at least one of its coordinates. Let $l = \max_{k \in \{1,\ldots,n\}}[v(k)+1]$. There exists an $M \in R_w$ such that $M(k) = w(k)$ or $M(k) \geq l$ if $w(k) = \omega$. Since $\forall k$, $v(k) \leq w(k)$, and $v(k) < l$, we have that $v(k) < M(k)$. Since $\lceil V \rceil$ is covering (Lemma 6.6), there exists a $v' \in \lceil V \rceil$ such that $v' \geq M$. Then we have that $v' \geq M > v$, which cannot be true since v is a maximal element of V. Thus R_w must be finite.

The second case is where R_v is infinite. In this case for at least one $k \in \{1, \ldots, n\}$, $v(k) = \omega$. For each $b \in \mathbb{N}$, we choose a corresponding M_b from R with $M_b(k) = v(k)$, or $M_b(k) \geq b$ if $v(k) = \omega$. By Dickson's Lemma, we can choose $i_0 < i_1 < \ldots$ such that (M_{i_j}) is a non-decreasing sub-sequence. All these M_{i_j} must be covered by $\lceil W \rceil$, and thus especially by a finite number of elements. In fact, any element that does not cover the entire set just covers a finite part due to the ordering, which means that this sequence (M_{i_j}) must be covered by just one $w \in \lceil W \rceil$. For each $k \in \{1, \ldots, n\}$, if $v(k) = \omega$, then $w(k) = \omega$, otherwise, if $w(k) = m \in \mathbb{N}$, then $w \not\geq N_{i_{m+1}}$. Also, if $v(k) \in \mathbb{N}$ then $w(k) \geq v(k)$ otherwise it would not cover any of the elements of (M_{i_j}). Thus we have that $w \geq v$, and by repeating the argument, we can find a $v' \in \lceil V \rceil$ which covers all elements of (the infinite) R_w, such that $v' \geq w$. Since $w \geq v$ and v is maximal, we have that $v' = w = v$. This shows that $v \in \lceil W \rceil$.

The two cases show that $\lceil V \rceil \subseteq \lceil W \rceil$, and by symmetry that $\lceil V \rceil = \lceil W \rceil$. $\square$

Theorems 6.13 and 6.14 yield the following result.

Theorem 6.15. *Let $R \subseteq \mathbb{N}_\omega^n$. Then $\left\lceil \overline{R} \right\rceil$ is finite and the minimal covering set of R.*

Proof. Theorem 6.13 states that $\left\lceil \overline{R} \right\rceil$ is a finite covering set of R.

Let V be any finite covering set of R which is of smaller or equal size to $\left\lceil \overline{R} \right\rceil$. Since we are trying to find a minimal set, we can assume that $\lceil V \rceil = V$. Theorem 6.14 states that $\lceil V \rceil = \left\lceil \left\lceil \overline{R} \right\rceil \right\rceil$. Since $\left\lceil \left\lceil \overline{R} \right\rceil \right\rceil = \left\lceil \overline{R} \right\rceil$, it follows that $V = \left\lceil \overline{R} \right\rceil$. $\square$

This theorem shows that we have achieved the goal of creating a finite covering set from the reachability set. By choosing the maximal elements of the closure, we not only have a finite set, but one that is the smallest possible.

7 Analysis of PTI nets

PTI nets are not just a simple extension from P/T nets. Inhibitor arcs add modelling power, at the cost of monotonicity as described in Section 3.3. In this section we will look at some properties of PTI nets, and some techniques to analyse them.

7.1 Turing complete

The PTI net model has significant modelling power. In fact, general PTI nets are even Turing complete, i.e., it can execute any general algorithm. One way, as described by [Age74], is by simulating two stacks that represent the tape of a Turing machine. These stacks are simulated by placing a special number of tokens in the places, and then using multiplication and division to modify the stacks. [Pet81] shows another way via register machines. With just two registers and the appropriate operations, a register machine can simulate a Turing machine. In turn, a register machine with two registers can be simulated by a PTI net with two inhibitor places.

An important corollary is that the reachability problem for general PTI nets is undecidable. Otherwise, we could construct a PTI net which corresponds to a Turing machine, and check whether it could reach a marking corresponding to the halting state of the Turing machine. It is well-known that this problem, the halting problem, is undecidable. See for example [Pul00] for a nice short proof.

Although reachability is not decidable for general PTI nets, [Rei08] proves that it is decidable for PTI nets with a single inhibitor arc.

7.2 Coverability

In [KK08] a modified version of the standard coverability tree algorithm for PTI nets is proposed. This algorithm correctly identifies unbounded places, but may fail to detect all of them. [vdVKK11] shows that there is a PTI net with two inhibitor arcs for which the algorithm does not terminate. For PTI nets with only one inhibitor arc this algorithm *does* terminate.

7.3 Bounded inhibitor places

Consider a PTI net $\mathcal{PTI} = (P, T, W, M_0, I)$ with just one inhibitor place p. That is, whenever for places p' and transitions t, if $I(p', t) < \omega$ then $p = p'$. Note that this inhibitor place may have more than one inhibitor arc. Assume that this place p is bounded by some $b \in \mathbb{N}$. Thus, for all reachable markings M, $M(p) \leq b$. With this we can apply the following transformation:

Transformation 7.1. Without loss of generality, assume $b \geq M_0(p)$. Remove all inhibitor arcs from p by complementing it, i.e., add a place p_b with arrows such that $W(p_b, t) = W(t, p)$ and $W(t, p_b) = W(p, t)$ for all t, such that $M(p_b) = b - M(p)$ for all reachable markings M. Finally we replace the inhibitor arc $I(p, t)$ by two arcs $W(t, p') = W(p', t) = b - I(p, t)$.

Figure 8 shows an example of this transformation.

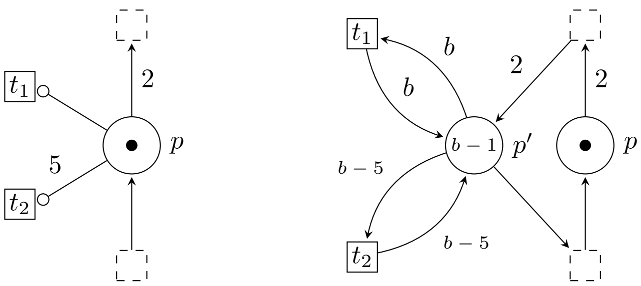


Figure 8: Example of Transformation 7.1

If b is a bound for p, then the reachability graphs of the PTI net and its transformation are isomorphic. If b is not a bound, markings only reachable from markings where p has more than b tokens are lost.

The following theorem and its proof show us that using Transformation 7.1 we can decide whether an assumed bound is correct.

Theorem 7.2. *Let $\mathcal{PTI} = (P, T, W, M_0, I)$ be a PTI net with one inhibitor place p. For any given $b \in \mathbb{N}$, it is possible to decide whether for all reachable markings M, $M(p) \leq b$.*

Proof. To decide this, consider the following algorithm.

1. Apply Transformation 7.1 to $\mathcal{PTI}$ to obtain $\mathcal{PTI}_b$. The newly created complement place will be called p_b.
2. Construct a coverability tree for $\mathcal{PTI}_b$ using Algorithm 1, but:

3. If some transition t is not enabled in the current marking M when executing line 9, check the following
   - If $M(p) > {}^{\circ}t(p)$ continue with the algorithm
   - If $\exists q \in P$ with $M(q) < {}^{\bullet}t(q)$, continue with the algorithm
   - Else output 'Incorrect bound' and halt
4. If the full coverability tree is created without halting, output 'Correct bound'.

Suppose the algorithm outputs 'Incorrect bound'. Then at some point, there is a transition t that is not enabled in the current marking M of $\mathcal{PTI}_b$. $M(p_b) < {}^{\bullet}t(p_b)$, while for all other $q \in P$, $M(q) \geq {}^{\bullet}t(q)$. If M contains an ω, the algorithm ensures that there is a reachable marking M' such that on all $q \in P$ with $M(q) = \omega$, $M'(q) \geq {}^{\bullet}t(q)$. Since p_b is bounded by construction, $M(p_b) < \omega$. If we would have used $b + {}^{\bullet}t(p_b)$ rather than b in the transformation, transitions t would be enabled, since only p_b is influenced by this and it was not inhibited by an inhibitor arc. Thus we can conclude that b is an incorrect bound, as the raising of the bound will extend the reachability graph.

Suppose the algorithm gives the answer 'Correct bound'. If b is incorrect, the reachability graphs of $\mathcal{PTI}$ and $\mathcal{PTI}_b$ will not be isomorphic. Since both start with the same marking, we can then find a reachable marking M in $\mathcal{PTI}$ where $M[t\rangle M'$, but where t is not enabled in the corresponding marking of $\mathcal{PTI}_b$. The generated coverability graph CT of $\mathcal{PTI}_b$ has a marking $N \geq M$. Since $N(p) + N(p_b) = b$, and $N \geq M$, we know that $N(p) = M(p)$. This shows that t is not enabled in N in $\mathcal{PTI}$, and it is enabled in N in $\mathcal{PTI}$ and thus it will not be that $M(p) > {}^{\circ}t(p)$, so the other continue case must be true. However, for all $q \in P$, the value of ${}^{\bullet}t(q)$ is the same in $\mathcal{PTI}$ and $\mathcal{PTI}_b$, so $N(q) \geq {}^{\bullet}t(q)$. This is a contradiction, and thus it will be that the algorithm would correctly identify the incorrect bound.

This shows that the algorithm provides a correct answer in all cases. It will also always halt, as Algorithm 1 always halts, and all other checks in the described algorithm run in finite time as well. This shows that it is decidable whether for all reachable markings M of $\mathcal{PTI}$, $M(p) \leq b$ for some fixed b. $\square$

The above is for PTI nets with just one inhibitor place. If we would transform a net with more inhibitor places, then we cannot judge the bound of individual places. Although we have not proven it here, we conjecture that the above proof is also valid to determine whether a bound is correct for *all* inhibitor places, since the only real testing is done on the newly created complement places.

7.4 Primitive PTI nets

Finally we consider a subclass of PTI nets where the nets are allowed to have more than one inhibitor place. We do however impose some other constraint on the net, such that we can still apply some analysis techniques on them.

Definition 7.3 ([Bus02, Definition 4.1])**.** Let $\mathcal{PTI}$ be a PTI net. Let p be an inhibitor place, i.e., a place for which $\exists t$, $I(p,t) < \omega$. The *emptiness limit* $EL(p)$ of p is the minimal number such that the following holds: For all reachable markings M, if $M(p) > EL(p)$, then for all markings M' reachable from M and $t \in T$, if t is enabled in M' then $I(p,t) = \omega$.

A *primitive* PTI net is a PTI net that has an emptiness limit for all inhibitor places.

Suppose a place p has an emptiness limit k. If p would contain more than k tokens in some marking, then all transitions inhibited by p will never be able to fire again. That means that p will be acting as a 'normal' place from then on, and we can ignore the transitions inhibited by p.

The main idea is, that in a normal P/T net, due to the monotonicity, smaller markings never allow for more behaviour than larger ones. In a PTI net, some large markings can behave differently after they become smaller again. In a primitive PTI net, when it has crossed a certain threshold, this special behaviour of PTI nets disappear for that place.

Definition 7.4 ([Bus02, Definition 5.16])**.** Let $\mathcal{PTI} = (P, T, W, M_0, I)$ be a PTI net and $\mathcal{PT} = (P', T', W', M'_0)$ a P/T net. We say that $\mathcal{PT}$ *simulates* $\mathcal{PTI}$ iff there exists a mapping $\eta : T' \to T$ such that:

- If $t_1 \dots t_n$ is a firing sequence of $\mathcal{PTI}$ then there exists a firing sequence $t'_1 \dots t'_n$ of $\mathcal{PT}$ such that $\eta(t'_i) = t_i$ for $i \in \{1, \dots, n\}$;
- If $t'_1 \dots t'_n$ is a firing sequence of $\mathcal{PT}$ then $\eta(t'_1) \dots \eta(t'_n)$ is a firing sequence of $\mathcal{PTI}$.

Theorem 7.5. *The class of primitive nets is the largest one that can be simulated by a P/T net.*

This statement is proven in [Bus02, Theorem 5.17]. It shows us that every primitive PTI net can be simulated by a P/T net, and that every PTI net that can be simulated by a P/T net is essentially a primitive PTI net.

Example 7.6. There exists a PTI net with just one inhibitor arc than cannot be simulated by a P/T net.

Let $\mathcal{PTI}$ be the net in Figure 9. The place p_3 is unbounded, as the sequence $\sigma_n = (t_1 t_2 t_3)^n t_1^n t_2^n$ puts n tokens in that place for any $n \in \mathbb{N}$. If after σ_n we were to fire t_3 n times, the previously inhibited transition t_1 becomes enabled again. This shows that $\mathcal{PTI}$ has no emptiness limit for p_3, and thus it cannot be simulated by a P/T net as per Theorem 7.5.

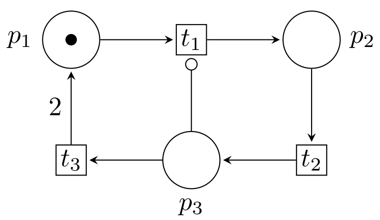


Figure 9: A non-primitive PTI net

Lemma 7.7. *Let $\mathcal{PTI}$ be a PTI net with one inhibitor place p. If p is bounded by $b \in \mathbb{N}$, then $\mathcal{PTI}$ can be simulated by a P/T net.*

Proof. We can set the emptiness limit of p to b. This is valid, as p will never have more than b tokens. Theorem 7.5 then states that $\mathcal{PTI}$ can be simulated. $\square$

Example 7.8. The converse is not true. Let $\mathcal{PTI}$ be the PTI net in Figure 10. Place p_1 has an emptiness limit of 0. Once it contains a token, t_2 will never be enabled again. However, p_1 is not bounded, as we can obtain any number of tokens in it by firing t_1 as many times as we want.

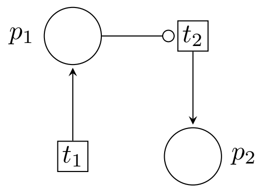


Figure 10: A primitive PTI net

To conclude, although primitive PTI nets are more limited than general PTI nets, they do offer more tools to analyse them.

8 Conclusion

This research started with the search for a coverability algorithm for a subclass of PTI nets. This goal was accomplished by discovering a semi-algorithm for determining bounds on single inhibitor places in PTI nets. Aside from this, some other new insights where also gained.

In the first part, we analysed the construction of a coverability set using trees or graphs. For P/T nets, these algorithms terminate and produce a finite set. The order of expanding nodes in the algorithms can be chosen freely, which may produce different coverability sets. That is why we provided some methods for choosing a subset of coverability sets, as a representative so to speak.

Following this, we did a mathematical analysis of the covering set. This was to gain insight in the construction of it, without using the information of the net structure. Using limits of non-decreasing sequences, a closure of the reachability set can be found. The maximal elements of this closure were found to be a covering set. This finite set was also proven to be the unique minimal covering set. There is not a clear bound on the size of this minimal set. It may depend on the number of (un)bounded places, but the number of tokens in the initial marking and the weight of the arcs also clearly play a role.

In the last part of the research, we analysed some PTI nets. The general class was found to be too complex to study, so we first switched to nets with just one inhibitor place. With this limitation, we found that we can check whether the inhibitor place is bounded by a specific number. We conjectured that a similar algorithm exists for more than one inhibitor place, since there the decision process is fairly simple. However, there could be subtle interactions between the inhibitor places that could complicate it. If we have a bound on the inhibitor place, we can transform it to a P/T net and perform the analysis there. After this, we looked at the class of primitive PTI nets. This was found out to be the largest class that could be transformed to P/T nets. And indeed, PTI nets with bounded inhibitor places are actually primitive, although the converse does not have to hold.

During the research, we also considered steps as described in [KK08]. This could be an interesting point for further research, as Transformation 7.1 does not preserve certain steps in the a priori semantics. Something else is that in this research, we could only check whether a bound on an inhibitor place is correct. It is currently open for further study whether the existence of such a bound is decidable. Lastly, both a P/T net and PTI net can have place-invariants, which can be obtained using Linear Algebra. In such an invariant, the number of tokens remains constant (up to some multiplicity). Combining this with other tools may also provide new insights.

In this paper, Sections 5, 6 and especially 7.3 described novel work from the author, while the other sections provide information found in the literature.

References

[Age74] Tilak Agerwala. Complete model for representing the coordination of asynchronous processes. Technical report, Johns Hopkins Univ., Baltimore, Md.(USA), 1974.

[Bus02] Nadia Busi. Analysis issues in petri nets with inhibitor arcs. *Theoretical Computer Science*, 275(1):127–177, 2002.

[Dic13] Leonard Eugene Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *American Journal of Mathematics*, 35(4):pp. 413–422, 1913.

[JvdAB+13] Kurt Jensen, Wil van der Aalst, Gianfranco Balbo, Maciej Koutny, and Karsten Wolf. *Transactions on Petri Nets and Other Models of Concurrency VII*. Springer, 2013.

[KK08] Jetty Kleijn and Maciej Koutny. *Steps and coverability in inhibitor nets*. University of Newcastle upon Tyne, Computing Science, 2008.

[KK12] Jetty Kleijn and Maciej Koutny. Step coverability algorithms for communicating systems. *Science of Computer Programming*, 77(78):955 – 967, 2012.

[KM69] Richard M. Karp and Raymond E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147 – 195, 1969.

[May81] Ernst W. Mayr. An algorithm for the general petri net reachability problem. In *Proceedings of the thirteenth annual ACM symposium on Theory of computing*, STOC '81, pages 238–246, New York, NY, USA, 1981. ACM.

[Pet62] Carl Adam Petri. *Kommunikation mit Automaten*. PhD thesis, Universität Hamburg, 1962.

[Pet81] James Lyle Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1981.

[Pul00] Geoffrey K. Pullum. Scooping the Loop Snooper. *Mathematics Magazine*, 73(4):319–320, October 2000.

[Rei08] Klaus Reinhardt. Reachability in petri nets with inhibitor arcs. *Electronic Notes in Theoretical Computer Science*, 223:239–264, 2008.

[Rei13] W. Reisig. *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*. Springer-Verlag New York Incorporated, 2013.

[RR98] Wolfgang Reisig and Grzegorz Rozenberg. *Lectures on Petri Nets I: Basic Models: Advances in Petri Nets*, volume 1. Springer, 1998.

[vdVKK11] Sander van der Vlugt, Jetty Kleijn, and Maciej Koutny. *Coverability and Inhibitor Arcs: an example*. Computing Science, Newcastle University, 2011.

[VH12] Antti Valmari and Henri Hansen. Old and new algorithms for minimal coverability sets. In Serge Haddad and Lucia Pomello, editors, *Application and Theory of Petri Nets*, volume 7347 of *Lecture Notes in Computer Science*, pages 208–227. Springer Berlin Heidelberg, 2012.