UNIVERSITÉ LIBRE DE BRUXELLES
Faculté des Sciences
Département d'Informatique

The Coverability problem for parametric Petri nets

Alexis Reynouard

Promotor : Prof. Gilles Geeraerts

Master Thesis in Computer Sciences

Academic year 2018 - 2019

Contents

Chapter 1

Introduction

Petri nets (PNs) are a mathematical and graphical model introduced by Carl Adam Petri in 1962 [17]. It was successfully used to analyse systems in a wide range of domains, especially for the formal verification of asynchronous systems.

In their standard definition, PNs are instantiated through many natural numbers which may represent, for example, the amount of resource needed for a given action to be carried out.

The introduction of parameters into the model to avoid the need to state these values explicitly[1] may have several benefits: it may allow to perform analysis of a whole family of PNs in an efficient way, like in [2]; or to model dynamic changes in the system, as introduced by [3] as a subclass of reconfigurable nets.

The use of parameters increases the modelling power of PNs but also make some basic coverability problems undecidable in the general case [6].

We adopt the parametric Petri net model introduced by [6], which seems the most general, and we study the existing results and algorithms for conventional Petri nets to find out wether or not they still hold or how to adapt them to the parametric model.

The rest of the document is as follows: We define the plain Petri net model (*i.e.* the classical one) and the parametric model. We then briefly motivate our study [TODO: and give concrete examples of applications]. Third, we recall, first, the results already obtained for the parameterized Petri net as we have defined them, second, the classical results that we will study on this new model. Then, we focus on the parametric Petri net model to establish wether the results related to the coverability problem in the plain Petri net model still hold or if the algorithms may be adapted to this new model.

1.1 Definitions

Definition 1 (Petri net)**.** A Petri net (PN) $\mathcal{N}$ is a weighted oriented bipartite graph, whose the two subsets of vertices define a tuple $\langle P, T \rangle$ where:

[1]One can find in the literature many other way to use parameters in PNs. For example, place and / or transitions may also be parameters in order to dynamically change the network structure, like in [5].

- P is a finite set of places,
- T is a finite set of transitions.

For each transition $t \in T$ are defined (exactly) these two functions:

- $I_t : P \mapsto \mathbb{N}$ associates to each place the weight of the edge to t *(input weight)*,
- $O_t : P \mapsto \mathbb{N}$ associates to each place the weight of the edge from t *(output weight)*.

It is denoted by $t = \langle I_t, O_t \rangle$. Because these functions define the edges of the graph, a PN is completely defined by the tuple $\langle P, T \rangle$ and so is denoted by $\mathcal{N} = \langle P, T \rangle$.

Definition 2 (marking)**.** Given a set of place P, a marking over P is a function $\mathbf{m} : P \mapsto \mathbb{N}$ that associates $\mathbf{m}(p)$ tokens to each place $p \in P$.

An order on the markings is essential for the analysis of PNs. The order we will define is a well quasi order and a partial order.

Definition 3 (quasi order)**.** A quasi order on a set $\mathcal{E}$ is a binary relation R that is:

$$\begin{array}{ll} \text{reflexive:} & \forall x \in \mathcal{E},\ x\ R\ x \\ \text{transitive:} & \forall (x, y, z) \in \mathcal{E}^3,\ (x\ R\ y \wedge y\ R\ z) \Rightarrow x\ R\ z \end{array}$$

Definition 4 (well quasi order)**.** A well quasi order $\sqsubseteq$ on a set $\mathcal{E}$ is a quasi order on $\mathcal{E}$ such that, for any infinite sequence $s = e_0, e_1, e_2, \ldots$ of elements from $\mathcal{E}$, there exist indices $i < j$ with $e_i \sqsubseteq e_j$. That is, there is no infinite antichain in $\mathcal{E}$ for this relation.

Definition 5 (partial order)**.** A partial order on a set $\mathcal{E}$ is a quasi order R that is

$$\text{antisymmetric:} \qquad \forall (x, y) \in \mathcal{E}^2,\ (x\ R\ y \wedge y\ R\ x) \Rightarrow x = y$$

Definition 6 (partial order $\preceq$ on the markings)**.** Given a set of places P, the partial order $\preceq \subseteq \mathbb{N}^{|P|} \times \mathbb{N}^{|P|}$ is such that for all pair of markings $(\mathbf{m}_1, \mathbf{m}_2) \in \mathbb{N}^{|P|} \times \mathbb{N}^{|P|}$ we have that $\mathbf{m}_1 \preceq \mathbf{m}_2$ if and only if for all place $p \in P : \mathbf{m}_1(p) \leq \mathbf{m}_2(p)$.

$\mathbf{m} \prec \mathbf{m'}$ denote that $\mathbf{m} \preceq \mathbf{m'}$ and $\mathbf{m'} \not\preceq \mathbf{m}$.

Lemma 1 ([7])**.** $\preceq$ *is a well quasi order.*

The following result will be useful in the sequel.

Lemma 2 ([4])**.** *The PN model is* strongly monotonic with regard to $\preceq$*. That is, for all PN $\mathcal{N} = \langle P, T, \mathbf{m}_0 \rangle$, for all transition $t \in T$ and for all markings $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$ of $\mathcal{N}$ such that $\mathbf{m}_1 \prec \mathbf{m}_2$ and $\mathbf{m}_1 \xrightarrow{t} \mathbf{m}_3$, there exists a marking $\mathbf{m}_4$ of $\mathcal{N}$ such that $\mathbf{m}_2 \xrightarrow{t} \mathbf{m}_4$ and $\mathbf{m}_3 \prec \mathbf{m}_4$.*

In this work we will focus on an extension of the PN model, the parametric Petri net (PPN) model, that is extended thanks to the use of parameters as input and output weights.

Definition 7 (parametric Petri net)**.** A parametric Petri net (PPN) $\mathcal{S} = \langle P, T, \mathbb{P} \rangle$ is a weighted oriented bipartite graph with a finite set $\mathbb{P}$ of parameters. The two subsets of vertices are:

- P: a finite set of places,
- T: a finite set of transitions,

For each transition $t \in T$ are defined the following functions:

- $I_t : P \mapsto \mathbb{N} \cup \mathbb{P}$ associates to each place the weight of the edge to t *(input weight)*,
- $O_t : P \mapsto \mathbb{N} \cup \mathbb{P}$ associates to each place the weight of the edge from t *(output weight)*.

As for plain PNs, this is denoted $t = \langle I_t, O_t \rangle$.

Definition 8 (parametric marking)**.** Given a set of place P, a parametric marking over P is a function $\mathbf{m} : P \mapsto \mathbb{N} \cup \mathbb{P}$ that associates $\mathbf{m}(p)$ tokens to each place $p \in P$.

A marking of a PN $\mathcal{N} = \langle P, T \rangle$ is a marking over P. A marking of a PPN $\mathcal{S} = \langle P, T, \mathbb{P} \rangle$ is a *parametric* marking over P. Note that a marking $\mathbf{m}$ is a parametric marking where $\mathbf{m}(p) \in \mathbb{N}$ for all $p \in P$.

Definition 9 (initialized (parametric) PN)**.** An initialized PN $\mathcal{N} = \langle P, T, \mathbf{m}_0 \rangle$ (resp. PPN $\mathcal{S} = \langle P, T, \mathbb{P}, \mathbf{m}_0 \rangle$) is a PN (resp. PPN) with an initial marking $\mathbf{m}_0$.

This is sometimes called a *marked (parametric) PN.* We will often refer to an initialized (parametric) PN loosely as a (parametric) PN.

The figure 1.1 shows an example of PPN whose $\mathbb{P} = \{a, b\}$ and with an initial marking $\mathbf{m}_0$ such that $\mathbf{m}_0(p_1) = 1$, $\mathbf{m}_0(p_2) = a$ and $\mathbf{m}_0(p_3) = 0$. The circles represent the places, the rectangles are the transitions and the dots are the tokens. If the number of token at a given place is parametric (*i.e.* depends on a parameter of $\mathbb{P}$), it is written inside the circle. An arrow from a place p and to a transition t denotes that $I_t(p) = 1$. If there is no arrow from p to t, $I_t(p) = 0$. If $I_t(p) \notin \{0, 1\}$, a label with the value of $I_t(p)$ is added to the arrow. Symmetrically, the arrows from the transitions to the places indicate the output weights.

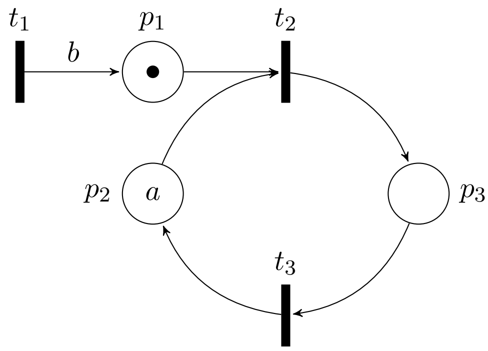


Figure 1.1: An initialized PPN

We usually set an order on the places. This allows to view the markings as vectors (here, $\mathbf{m}_0$ is the column vector $(1, a, 0)^T$, where $\cdot^T$ is the transpose operator) as well as the I and O functions. Likewise, we define an order on the transitions. Therefore, I_t and O_t denote respectively the I and O functions defined for the t^{th} transition (here, $I_1 = (0, 0, 0)^T$ and $O_1 = (b, 0, 0)^T$). Given a PPN $\mathcal{S} = \langle P, T, \mathbb{P} \rangle$, the backward and forward incidence matrices $\mathbf{I}_{\mathcal{S}} \in (\mathbb{N} \cup \mathbb{P})^{|P|\times|T|}$ and $\mathbf{O}_{\mathcal{S}} \in (\mathbb{N} \cup \mathbb{P})^{|P|\times|T|}$ are naturally defined by $\mathbf{I}_{\mathcal{S}}(p,t) = I_t(p)$ and $\mathbf{O}_{\mathcal{S}}(p,t) = O_t(p)$. ($\mathcal{S}$ is omitted when it is obvious from the context.) This allows to use linear algebra to analyse PNs.

$$\mathbf{I} = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{ccc} t_1 & t_2 & t_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \qquad \mathbf{O} = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{ccc} t_1 & t_2 & t_3 \end{array} \\ \begin{bmatrix} b & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array}$$

Figure 1.2: The incidence matrices of the PN from figure 1.1

1.1.1 Operational semantic of PNs

Given a PN $\mathcal{N} = \langle P, T \rangle$ and a marking $\mathbf{m}$ on $\mathcal{N}$, a transition $t \in T$ is said *enabled* by $\mathbf{m}$ if $\forall p \in P : \mathbf{m}(p) \geq I_t(p)$. An enabled transition can be *fired* to produce a new marking $\mathbf{m}'$ such that $\forall p \in P : \mathbf{m}'(p) = \mathbf{m}(p) - I_t(p) + O_t(p)$. This is denoted by $\mathbf{m} \xrightarrow{t} \mathbf{m}'$. It is important to note that the effect of a transition is to add or remove a constant number of tokens at each place and does not depend on the marking from which it is fired. A PN transition is said to have a *constant effect*. [TODO: Note difference when talking about ω-Petri nets]

Here are some additional notations:

- $\mathbf{m} \to \mathbf{m}'$ denotes that there exists $t \in T$ such that $\mathbf{m} \xrightarrow{t} \mathbf{m}'$.
- $\mathbf{m} \xrightarrow{\sigma} \mathbf{m}'$ where σ is a sequence of transitions $\sigma = (t_1, \ldots, t_{n-1}), t_i \in T, i \in \{1, \ldots, n-1\}$ denotes that there exists a sequence of markings $\mathbf{m}_1, \ldots, \mathbf{m}_n$ such that : $\mathbf{m} = \mathbf{m}_1 \xrightarrow{t_1} \cdots \xrightarrow{t_{n-1}} \mathbf{m}_n = \mathbf{m}'$.
- $\mathbf{m} \xrightarrow{*} \mathbf{m}'$ denotes that there exists a sequence of transition σ such that $\mathbf{m} \xrightarrow{\sigma} \mathbf{m}'$. Note that the $\xrightarrow{*}$ relation is the reflexive and transitive closure of the relation $\to$.

Definition 10. Given an PN $\mathcal{N} = \langle P, T \rangle$ and a marking $\mathbf{m}$ of $\mathcal{N}$:

- $\mathsf{Post}(\mathbf{m}) = \{\mathbf{m}' \mid \mathbf{m} \to \mathbf{m}'\}$ is the set of one-step successors of $\mathbf{m}$,
- $\mathsf{Pre}(\mathbf{m}) = \{\mathbf{m}' \mid \mathbf{m}' \to \mathbf{m}\}$ is the set of one-step predecessors of $\mathbf{m}$,
- $\mathsf{Post}^*(\mathbf{m}) = \{\mathbf{m}' \mid \mathbf{m} \xrightarrow{*} \mathbf{m}'\}$ is the set of successors of $\mathbf{m}$, in any number of step. With $\mathbf{m}_0$ the initial marking of $\mathcal{N}$, $\mathsf{Post}^*(\mathbf{m}_0)$ is the *reachability set* of $\mathcal{N}$.
- $\mathsf{Pre}^*(\mathbf{m}) = \{\mathbf{m}' \mid \mathbf{m}' \xrightarrow{*} \mathbf{m}\}$ is the set of predecessors of $\mathbf{m}$, in any number of step.

These operators are naturally extended to sets of markings as the union of the sets obtained by applying the operator on each marking of the sets. That is, with M a set of markings of $\mathcal{N} = \langle P, T \rangle$, $\mathsf{Post}(M) = \{\mathbf{m}' \mid \exists \mathbf{m} \in M : \mathbf{m} \to \mathbf{m}'\}$.

For example, regarding the PPN shown on figure 1.1, $\mathsf{Post}((0,1,0)) = \{(b,1,0)\}$ and $\mathsf{Post}^*((0,1,0)) = \{(i,1,0) \mid i \in \mathbb{N}\} \cup \{(i,0,1) \mid i \in \mathbb{N}\}$.

All of this applies to PPN through valuations of the parameters:

Definition 11 (Instantiation of PPNs)**.** Let $\mathcal{S} = \langle P, T, \mathbb{P}, \mathbf{m}_0 \rangle$ be a PPN and $v : \mathbb{P} \mapsto \mathbb{N}$ be a *valuation* on $\mathbb{P}$. Then $v(\mathcal{S})$ is defined as the PN obtained by replacing each parameter $a \in \mathbb{P}$ by $v(a)$. Thus, we have $v(\mathcal{S}) = \langle P, T, \mathbf{m'}_0 \rangle$ such that :

- $\mathbf{I}_{v(\mathcal{S})}(p,t) = \begin{cases} \mathbf{I}_\mathcal{S}(p,t) & \text{if } \mathbf{I}_\mathcal{S}(p,t) \in \mathbb{N} \\ v(\mathbf{I}_\mathcal{S}(p,t)) & \text{if } \mathbf{I}_\mathcal{S}(p,t) \in \mathbb{P} \end{cases}$
- $\mathbf{O}_{v(\mathcal{S})}(p,t) = \begin{cases} \mathbf{O}_\mathcal{S}(p,t) & \text{if } \mathbf{O}_\mathcal{S}(p,t) \in \mathbb{N} \\ v(\mathbf{O}_\mathcal{S}(p,t)) & \text{if } \mathbf{O}_\mathcal{S}(p,t) \in \mathbb{P} \end{cases}$
- $\mathbf{m'}_0(p) = \begin{cases} \mathbf{m}_0(p) & \text{if } \mathbf{m}_0(p) \in \mathbb{N} \\ v(\mathbf{m}_0(p)) & \text{if } \mathbf{m}_0(p) \in \mathbb{P} \end{cases}$

Given $\mathcal{S}$ a PPN and a valuation v, one can thus instantiate a PN $v(\mathcal{S})$ from $\mathcal{S}$ and apply the semantic described above. When the PPN under consideration is clear from the context, $\mathbf{I}_v$ is used to denote $\mathbf{I}_{v(\mathcal{S})}$ and $\mathbf{O}_v$ to denote $\mathbf{O}_{v(\mathcal{S})}$. We write $\xrightarrow{t}_v$, $\rightarrow_v$, $\xrightarrow{\sigma}_v$, $\xrightarrow{*}_v$, Post_v, Pre_v, Post^*_v and Pre^*_v to denote $\xrightarrow{t}$, $\rightarrow$, $\xrightarrow{\sigma}$, $\xrightarrow{*}$, Post, Pre, Post^* and Pre^* on the plain PN $v(\mathcal{S})$.

This makes it possible to formally represent a system and interactions between its components. We will now define some properties that the model may have and that are usually of interest to show that the modelled system meets some requirements.

1.1.2 Behavioural properties of PNs

The markings basically indicate the state of the system. Thus, knowing if an initialized PN may reach a given marking, that represents for example a bad state, is essential to check properties of the modelled system. This is the *reachability problem*.

Definition 12 (Reachability)**.** Given an initialized PN $\mathcal{N} = \langle P, T, \mathbf{m}_0 \rangle$ and a marking $\mathbf{m}$ of $\mathcal{N}$, $\mathbf{m}$ is said reachable if $\mathbf{m}_0 \xrightarrow{*} \mathbf{m}$.

However safety properties are more often analysed through the *coverability problem*, that is essentially asking if a PN can reach or exceed a given marking.

Definition 13 (Coverability)**.** Given an initialized PN $\mathcal{N} = \langle P, T, \mathbf{m}_0 \rangle$ and a marking $\mathbf{m}$ of $\mathcal{N}$, $\mathbf{m}$ is said coverable if there exists a marking $\mathbf{m'}$ such that $\mathbf{m} \preceq \mathbf{m'}$ and $\mathbf{m}_0 \xrightarrow{*} \mathbf{m'}$.

A set of markings is said coverable whenever one of them is coverable.

The behaviour of a PPN is defined by the behaviours of all the PNs that can be obtained by a valuation of its parameters. So, for an initialized PPN $\mathcal{S}$, the coverability problem may be declined in an existential and an universal form. The existential coverability problem ($\mathscr{E}$-cov) ask if there exists a valuation v such that $\mathbf{m}$ is coverable. The universal coverability problem ($\mathscr{U}$-cov) ask if $\mathbf{m}$ is coverable for all valuations v.

Definition 14 (Universal and existential coverability problems)**.** Given a PPN $\mathcal{S} = \langle P, T, \mathbb{P} \rangle$ and a non-parametric marking $\mathbf{m}$ of $\mathcal{S}$

- the *existential coverability problem* ask if there is a valuation v for $\mathbb{P}$ such that $\mathbf{m}$ is coverable,
- the *universal coverability problem* ask if $\mathbf{m}$ is coverable for all valuations of $\mathbb{P}$.

1.2 Motivations

1.2.1 Interests of PPNs

[TODO: sources and examples]

Today PNs are used in a wide range of areas. Most of the time, they are used either to design a safe system or to verify an existing one. This use requires that the system is complete. That is, for the design of a model, it must be entirely designed to be analyzable. On the other hand, when checking an existing system, if a desired property does not hold, the correction must be made "by hand".

With the introduction of parameters, some variables unknown at the design stage can be integrated into the model without having to set them arbitrarily. And, if during the verification a desired property turns out not to hold, it is possible to check if the change of parameters alone can solve the problem, or if the Petri net structure must be changed too. Going further, the use of parameters in the model can allow to determine the "safest values" for a system, or to synthesize the values that allow to respect a given strategy.

We can therefore say that parameters can simplify the *design* of a system. Indeed, since it is possible to keep unknown values, modelling can be done step by step, with the possibility to check the model at each step. In addition, the design can be partially automated by parameter synthesis. This approach gives a new interest for this model in fields as varied as chemistry, construction processes, financial loans...

There are also many advantages of using parameters when it comes to *verification*: For example, they allow to verify some properties simultaneously on many systems that differs only by parameters values.

1.2.2 Interest of the coverability problem in PPNs

[TODO: sources and examples]

Coverability problem is of primary interest for system design and verification, as it provide evidence of safety properties on the studied systems. Therefore, for the reasons given in the previous section, it is worth being able to solve it efficiently on PPNs.

To give a more concrete intuition on the interest, consider a system that execute a *task* for others systems. At each instant (whatever an instant is), the system may receive requests to perform the task from many other systems. We say that each request creates a *job*. We would like to have a system that is not too expensive to implement, but also capable of completing the tasks quickly enough. For this we make our system capable of performing a jobs at the same time, keeping a as low as possible to reduce costs. This system may be modeled as shown on the figure 1.1 with $\mathbf{m}_0 = (0, a, 0)$ as initial marking. p_1 represents the job queue and p_3 the execution unit.
We can now formally verify that, whatever the parameter values, the execution unit will not receive more than a jobs to perform at the same time, that is an instance of the $\mathscr{E}$-cov for the marking $(0, 0, a+1)$. Indeed, it is easy to see that $\mathsf{Post}^*((0, a, 0)) = \{(i, j, k) \mid i, j \text{ and } k \in \mathbb{N}, j+k = a\}$. [TODO: maybe remove part on a *as low as possible*]

Before recalling the known result on PPN and plain PN that will be useful to our study, let us give a brief overview of some work already done on PPNs.

1.3 Previous works on parametrization of PNs

The use of parameters in formal verification systems is a well developed topic in the literature.

With regard to PN, parameters have been introduced with many different roles. Some works, like [5] use parameters as places or transitions, for example to make it possible to change a place into a more complex subnet and thus allow different levels of abstractions to be considered. In [15], parameters are used on the markings to obtain concise parametrised reachability trees, but not to realize formal verifications on these parametric systems.

[3] introduces parameters as the weight of arcs to model changes in a system. The parameters have a finite valuation domain and verifications are performed on these parametrized systems. Systems with quantitative parameters with infinite valuation domains are analysed in [2]. Similarly, [16] study PNs with parametric initial markings which represent sets of possible initial markings.

Close to the PPN model, ω-PNs ([12]) allow input and output wheights to be ω. In this case, the transition consumes or produces a non-deterministic number of tokens.

Our work is in the line with [6] which use discrete parameters as arc weights as well as in the markings. [6] provide also a proof for the non decidability of $\mathscr{U}$-cov and $\mathscr{E}$-cov, and define several subclasses of PPN for which these problems are decidable.

Chapter 2

Preliminary results

2.1 Known results for coverability on plain PNs

We now present the results related to the coverability problem on the plain PNs which seem to us the most interesting.

To introduce these results, we need some additional definitions. They will be given for plain PNs, but most of them are naturally extended to PPNs.

The *coverability set* of an initialized PN is an over-approximation of the reachability set that is precise enough to solve the coverability problem, and is, therefore, interesting for our study. In order to define it formally, we need to know about the upward and downward closure of a (set of) marking(s).

Definition 15 (Upward- and downward-closure on markings)**.** Let $S \subseteq \mathbb{N}^{|P|}$ be a set of markings on the places P:

- The *upward-closure* of S, noted $\uparrow^{\preceq}(S)$, is the set $\{\mathbf{m} \in \mathbb{N}^{|P|} \mid \exists \mathbf{m'} \in S : \mathbf{m'} \preceq \mathbf{m}\}$,
- The *downward-closure* of S, noted $\downarrow^{\preceq}(S)$, is the set $\{\mathbf{m} \in \mathbb{N}^{|P|} \mid \exists \mathbf{m'} \in S : \mathbf{m} \preceq \mathbf{m'}\}$.

The closure of a marking $\mathbf{m}$ is the closure of the singleton $\{\mathbf{m}\}$.

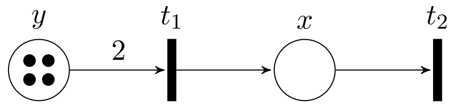
For instance, with $\mathbf{m} = (1, 2, 3)$, we have that its upward-closure is $\uparrow^{\preceq}(\mathbf{m}) = \{(i, j, k) \mid i \geq 1, j \geq 2, k \geq 3\}$.

Definition 16 (Upward- and downward-closed set of markings)**.** A set S of markings is said *upward-closed* if $S = \uparrow^{\preceq}(S)$. It is said *downward-closed* if $S = \downarrow^{\preceq}(S)$.

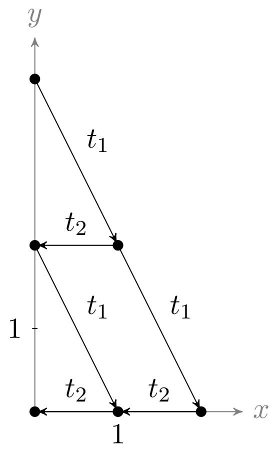
Definition 17 (Coverability set [8, 9])**.** Given an initialized PN $\mathcal{N} = \langle P, T, \mathbf{m}_0 \rangle$, a *coverability set* S of $\mathcal{N}$ is a set of markings such that $\downarrow^{\preceq}(S) = \downarrow^{\preceq}(\mathsf{Post}^*(\mathbf{m}_0))$.

Obviously, the minimal coverability set is thus $\downarrow^{\preceq}(\mathsf{Post}^*(\mathbf{m}_0))$ and is noted $\mathsf{Cover}(\mathcal{N})$.

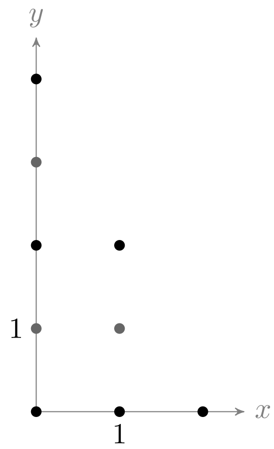
The figure 2.1a shows a marked PN with two places. One can therefore represents the markings as points on a plane. The figure 2.1b shows the reachable markings in the form of an accessibility graph. In 2.1c we see the minimal coverability set.



(a) A PN ($|P| = 2$)



(b) The reachable markings



(c) The minimal coverability set

Figure 2.1: Reachability and minimal coverability sets

Sometimes the number of token in a place is unbounded (*c.f.* the place boundedness problem). Therefore, the reachability and coverability sets are infinite. In a plain PN, this is due to the existence of an increasing self-covering sequence.

Definition 18 (Self-covering sequence)**.** Given an initialized PN $\mathcal{N} = \langle P, T, \mathbf{m}_0 \rangle$, a self-covering sequence is a sequence of the form:

$$\mathbf{m}_0 \xrightarrow{\rho} \mathbf{m}_i \xrightarrow{\sigma} \mathbf{m}_j$$

with ρ and σ two sequences of transitions of T and with $\mathbf{m}_i \preceq \mathbf{m}_j$.

Note that, since $\mathbf{m}_i \preceq \mathbf{m}_j$, σ is firable from $\mathbf{m}_j$. In addition, the monotonicity of PNs ensure that, with $\mathbf{m'}_j$ given by $\mathbf{m}_j \xrightarrow{\sigma} \mathbf{m'}_j$, we have $\mathbf{m}_j \preceq \mathbf{m'}_j$. Thus, we see that it is a sufficient condition for the *non-termination* of the system (the system may be able to fire transitions infinitely often). In fact, because $\preceq$ is a well quasi order (Lemma 1), one can find in any infinite sequence $\mathbf{m}_0 \to \mathbf{m}_1 \to \dots$ two markings $\mathbf{m}_i$ and $\mathbf{m}_j$ such that $\mathbf{m}_i \preceq \mathbf{m}_j$. And so, any infinite sequence is self-covering, and the existence of such a sequence is also necessary for the non-termination of the system.

Definition 19 (Increasing self-covering sequence)**.** Given an initialized PN $\mathcal{N} = \langle P, T, \mathbf{m}_0 \rangle$, an increasing self-covering sequence is a sequence of the form:

$$\mathbf{m}_0 \xrightarrow{\rho} \mathbf{m}_i \xrightarrow{\sigma} \mathbf{m}_j$$

with ρ and σ two sequences of transitions of T and with $\mathbf{m}_i \prec \mathbf{m}_j$.

Let $Q \subseteq P$ be the set of places $Q = \{q \in P \mid \mathbf{m}_i(q) < \mathbf{m}_j(q)\}$. $Q \neq \emptyset$ since $\mathbf{m}_i \prec \mathbf{m}_j$.

With a similar reasoning to the above, we see that having such a sequence ensure that one can reach a marking $\mathbf{m'}_j$ given by $\mathbf{m}_j \xrightarrow{\sigma} \mathbf{m'}_j$ such that $\mathbf{m}_j \prec \mathbf{m'}_j$. Because of the constant effect of transitions, we know that $\forall q \in Q : \mathbf{m}_j(q) < \mathbf{m'}_j(q)$. The unboundedness of the places in Q follows. [TODO: A more formal proof that the existence of an increasing self-covering sequence is a necessary and sufficient condition for unboundedness on places is either to be done here or to reference.]

An ω-marking is a way to represent a set of markings which have the same number of tokens in some places, and may have any number of tokens, potentially an infinity, in the other places. They are useful to effectively represent potentially infinite downward-closed set, like a coverability set or the sets of markings of an increasing self-covering sequence.

Definition 20 (ω-marking)**.** We define ω to be such that: $\omega \notin \mathbb{N}$ and for all constant $c \in \mathbb{N}$:

- $c \leq \omega$
- $\omega + c = \omega$
- $\omega - c = \omega$

An ω-marking $\mathbf{m}$ over a set of places P is a function $\mathbf{m} : P \mapsto \mathbb{N} \cup \{\omega\}$ that associates $\mathbf{m}(p)$ tokens to each place $p \in P$.

With $\mathbb{P}$ a set of parameters, $\omega \notin \mathbb{P}$, *a parametric ω-marking* $\mathbf{m}$ over a set of places P is a function $\mathbf{m} : P \mapsto \mathbb{N} \cup \mathbb{P} \cup \{\omega\}$ that associates $\mathbf{m}(p)$ tokens to each place $p \in P$.

Note that an ω-marking $\mathbf{m}$ is a parametric ω-marking where $\mathbf{m}(p) \in \mathbb{N}\cup\{\omega\}$ for all places $p \in P$. Similarly, a parametric marking $\mathbf{m}$ is a parametric ω-marking where $\mathbf{m}(p) \neq \omega$ for all places $p \in P$. As for parametric markings, we often refer to a parametric ω-marking simply as ω-marking.

Having an ω-marking $\mathbf{m} \in \mathsf{Cover}(\mathcal{N})$ denotes that for all marking $\mathbf{m}_1$ such that $\mathbf{m}_1(p) = \mathbf{m}(p) \forall p \in \{p \mid \mathbf{m}(p) \neq \omega\}$, there exists an marking $\mathbf{m}_2$ in the reachability set of $\mathcal{N}$ such that $\mathbf{m}_1 \preceq \mathbf{m}_2$. Notice also that an ω-marking stands for one and only one downward closed set.

2.1.1 Karp and Miller procedure

The Karp and Miller algorithm [14] is a classical algorithm to compute a coverability set of an initialized PN. More precisely, it constructs a coverability tree and uses an acceleration function to systematically detect self-covering sequences, and thus ensure the termination.

Definition 21 (Coverability tree)**.** Given a PN $\mathcal{N} = \langle P, T, \mathbf{m}_0 \rangle$, a coverability tree $\mathcal{T}$ of $\mathcal{N}$ is a labelled tree $\mathcal{T} = \langle N, B, n_0, \Lambda \rangle$ where:

- N is the set of nodes of the tree.
- $n_0 \in N$ is the root of the tree, *i.e.* $\nexists n \in N$ such that $(n, n_0) \in B$.
- $\Lambda : N \mapsto (\mathbb{N}\cup\{\omega\})^{|P|}$ is a labelling function that associate to each node a ω-marking of $\mathcal{N}$.
- $B \subseteq N \times N$, the set of edges, is such that:

- with $(n_1, n_2) \in N^2$, if there exists an edge $(n_1, n_2) \in B$ then there exists a sequence σ of transitions of T such that $\Lambda(n_1) \xrightarrow{\sigma} \Lambda(n_2)$, and
- for all node $n \in N \setminus \{n_0\}$, there exists a path from the root to n, that is, there exists a sequence of edges of B of the form $((n_0, n_1), (n_1, n_2), \ldots, (n_i, n))$, $i > 1$, and
- there is no cycles, that is, there is no sequences of edges of B of the form $((n_1, n_2), (n_2, n_3), \ldots, (n_i, n_1))$.

and such that $\downarrow^{\preceq}(\{\Lambda(n) \mid n \in N\}) = \mathsf{Cover}(\mathcal{N})$.

To keep N finite, the Karp and Miller procedure exploits the monotonicity of PNs to introduce ω-markings through an *acceleration function* Acceleration. This function takes a marking $\mathbf{m}$ to accelerate and a set of markings S as a base [TODO: ?] for the acceleration and returns a marking $\mathbf{m}_\omega$ such that:

$$\mathbf{m}_\omega(p) = \begin{cases} \omega & \text{if } \exists \mathbf{m}' \in S : \mathbf{m}' \prec \mathbf{m} \text{ and } \mathbf{m}'(p) < \mathbf{m}(p) \\ \mathbf{m}(p) & \text{otherwise} \end{cases}$$

We denote by $\mathscr{T}_n$ the path in the tree from the root to n.

The algorithm constructs the tree $\mathcal{T}$ as follows: The root n_0 of the tree is labelled with $\mathbf{m}_0$. A frontier F is defined to be the set of unprocessed nodes of the tree and is initialised to $\{n_0\}$. Then, while F is non empty, a node n is chosen to be processed: it is removed from F and, if there is no node n' in $\mathscr{T}_n$ such that $\Lambda(n) = \Lambda(n')$, for all ω-marking in $\{\mathsf{Acceleration}(\mathbf{m}, \mathscr{T}_n) \mid \mathbf{m} \in \mathsf{Post}(\Lambda(n))\}$, a node labelled with $\mathsf{Acceleration}(\mathbf{m}, \mathscr{T}_n)$ is added to the frontier and to the tree as a child of n.

The correctness and the termination of the algorithm lies on the strict monotonicity of PNs, and was proved by Karp and Miller in their work [14].

The Karp and Miller tree has a lot of convenient properties, and allows to answer coverability and [TODO:] problems. Furthermore, the Karp and Miller procedure can easily be adapted to some parametric problems [6], as we will show in section [TODO:].

However, this tree, although finite, is often much larger than the minimal coverability set, and cannot be constructed in reasonable time. As a consequence, many improvement were proposed, as well as other algorithms with different approaches.

2.1.2 An efficient computation method of the coverability set of Petri nets

[11, 13] propose another approach to the computation of the coverability set. It is not based on the Karp and Miller algorithm and is not an alternative to the Karp and Miller procedure in the sense that it does not allow to answer the same set of questions that the Karp and Miller tree solves. However, this techniques solve coverability problems more efficiently in practice.

As in the Karp and Miller algorithm, an acceleration function exploits the strict monotonicity of PN to allow termination. But here, the acceleration of a marking is performed with only one marking as the base (instead of a set of marking).

To choose the base to use, the algorithm work on pair of markings. Thus we will need the following definitions:

2.1.3 Expand, Enlarge and Check (EEC) algorithm

[11, 10]

[REMOVED: A backward algorithm [9, 1]]

2.2 Known results on PPNs

[TODO:] [6] as shown that $\mathscr{U}$-cov and $\mathscr{E}$-cov are undecidable on PPN and define natural subclasses of PPNs are defined in which parametric coberability problems are decidable.

PreT-PPNs are PPNs where parameters are used only in **I**. PostT-PPNs are PPNs where parameters are used only in **O**.

[6] provides an adaptation of the Karp and Miller Algorithm to solve the $\mathscr{U}$-cov problem on PreT-PPNs, and another to solve the $\mathscr{E}$-cov problem on PostT-PPNs.

Chapter 3

Contributions

3.1 Adapting Geeraerts efficient computation method of the coverability set of Petri nets for PostT-PPNs

3.2 Adapting EEC for PreT-PPNs

In this section we provide an adaption of EEC to work with PPNs. It is inspired by the [TODO: KM^{+}] algorithm from [6].

The algorithm is the same as the one presented in the section 2.1.3, with the difference that the condition to add a marking to the over-approximation is more restrictive.

Conclusions

The conclusions are to be written with care, because it will be sometimes the part that could convince a potential reader to read the whole document.

Bibliography

[1] P. A. Abdulla, K. Cerans, B. Jonsson, and Yih-Kuen Tsay. General decidability theorems for infinite-state systems. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, LICS '96, pages 313 – 321, Washington, DC, USA, 1996. IEEE Computer Society.

[2] Parosh Abdulla, Frédéric Haziza, and Lukáš Holík. All for the price of few: (parameterized verification through view abstraction). volume 7737, pages 476–495, 2013.

[3] Eric Badouel and Javier Oliver. Dynamic changes in concurrent systems: Modelling and verification. 1999.

[4] G. W. Brams. *Réseaux de Petri: théorie et pratique*. Number 1. 1983.

[5] Søren Christensen and Kjeld Høyer Mortensen. Parametrisation of coloured petri nets. *DAIMI Report Series*, 26(521), March 1997.

[6] Nicolas David. *Discrete Parameters in Petri Nets*. PhD thesis, Université de Nantes, Augustus 2017.

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

[8] Alain Finkel. A generalization of the procedure of karp and miller to well structured transition systems. In T. Ottmann, editor, *ICALP '87 Proceedings of the 14th International Colloquium, on Automata, Languages and Programming*, volume 267, pages 499 – 508. Springer-Verlag, 1987.

[9] Alain Finkel. Reduction and covering of infinite reachability trees. *Information and Computation*, 89(2):144 – 179, 1990.

[10] G. Geeraerts, J.-F. Raskin, and L. Van Begin. Expand, enlarge and check: New algorithms for the coverability problem of wsts. *Journal of Computer and System Sciences*, 72(1):180 – 203, 2006.

[11] Gilles Geeraerts. *Coverability and Expressiveness Properties of Well-structured Transition Systems*. PhD thesis, Université Libre de Bruxelles, 2007.

[12] Gilles Geeraerts, Alexander Heussner, M Praveen, and Jean-François Raskin. ω-petri nets: Algorithms and complexity. 137:29–60, 01 2015.

[13] Gilles Geeraerts, Jean-François Raskin, and Laurent Van Begin. On the efficient computation of the minimal coverability set for petri nets. In *International Symposium on Automated Technology for Verification and Analysis*, pages 98–113. Springer Berlin Heidelberg, 2007.

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

[15] M. Lindqvist. *Parametrized Reachability Trees for Predicate / Transition Nets*, pages 351–372. Springer Berlin Heidelberg, Berlin, Heidelberg, 1991.

[16] Marco Ajmone Marsan, G. Balbo, Gianni Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

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