RISC-Linz
Research Institute for Symbolic Computation
Johannes Kepler University
A-4040 Linz, Austria, Europe

Reachability Test in Petri Nets by Gröbner Bases

Olga CAPROTTI, Alois FERSCHA, Hoon HONG

(27 January 1995)

RISC-Linz Report Series No. 95-03

Supported by: *Parallel Symbolic Computation*, No S5302-PHY, Austrian Science Foundation (FWF)

Reachability Test in Petri Nets by Gröbner Bases[1]

Olga Caprotti* Alois Ferscha‡ Hoon Hong*

*Research Institute for Symbolic Computation

Johannes Kepler Universitä, A-4020 Linz, AUSTRIA

{ocaprott,hhong}@risc.uni-linz.ac.at

‡ Institut für Angewandte Informatik

Universität Wien Lenaugasse 2/8, A-1080 Vienna, AUSTRIA

ferscha@ani.univie.ac.at

27 January 1995

Abstract

A decision procedure for the reachability problem in the class of reversible Petri nets is described. We introduce the polynomial translation of a Petri net and use it to convert the reachability problem to a problem that can be solved by the well known algebraic technique of Gröbner basis. The main results of the theory of Gröbner Bases are reviewed and then applied to some examples originated from Petri nets. The extension of the decision algorithm to the more general case of non-reversible Petri nets is discussed.

Keywords: Petri nets, Reachability Problem, Equality Problem, Canonical Algebraic Simplification, Gröbner Bases Algorithm.

1 Introduction

Petri nets are named after A.C. Petri who introduced them in 1962 as formal models for representing and analysing parallel processes, [11, 12]. Since then, they gained increasing importance because their expressive power allows to model, graphically, complex systems with concurrency: parallel, distributed, synchronous

[1] This work was done in the framework of the project *Parallel Symbolic Computation*, No S5302-PHY, supported by: Austrian Science Foundation (FWF)

or asynchronous, [1, 2, 13, 15, 14]. One additional reason is that, despite the large complexity, most of the interesting problems are still decidable, [7].

A central question in the theory of Petri nets is the question of reachability. It amounts to decide whether, in a given net, a certain state can be reached starting from the initial state. Its importance stems from the fact that many other issues in Petri nets are equivalent to reachability or to some of its variants and subproblems, [8]. In general, the reachability problem is known to be decidable [9], but it requires at least exponential time and space [10] in the general case. A lot of effort has been devoted by the Petri net researchers in order to obtain efficient algorithm for solve it.

This paper will treat the reachability problem as suggested in [5] by a well known algorithm used in the computer algebra community. The main goal of this work is to promote the interaction between the researchers working in computer modelling of parallel systems by Petri nets, and those working in symbolic algebraic methods. We will show how to translate a Petri net into a set of polynomials on which to employ computer algebra tools for solving reachability. The class of Petri nets addressed by the decision algorithm is the class of reversible cyclic Petri nets. For the more general Petri net, the algorithm is just a semi-decision procedure and can be used to check whether a state is not reachable from the initial one.

The paper is structured as follows. In the next section, we recall basic definitions in the theory of Petri nets. Section 3 will review major results relative to the Gröbner basis method. Section 4 will describe how to transform a reachability problem into a problem in polynomial ideal theory and will give a procedure to decide it. We will show an example of the effectivity of the method presented by using the computer algebra system MAPLE in Section 5. The final section is devoted to the discussion of possible extensions of the algorithm.

2 Reachability in Petri nets

A Petri net, $\mathcal{P} = (P, T, F, w, \mu^{(0)})$, is a tuple defined by:

> P, the set $\{p_1, p_2, \ldots p_n\}$ of *places*,
>
> T, the set $\{t_1, t_2, \ldots t_m\}$ of *transitions*,
>
> $F \subseteq (P \times T) \cup (T \times P)$, the *input-output relation*,
>
> $w : F \mapsto \mathbb{N}^+$, an assignment of weights $w((p,t))$ to arcs $(p,t) \in F$,
>
> $\mu^{(0)}$, an initial *marking*, (generated by the marking function $\mathcal{M} : P \mapsto \mathbb{N}^0$).

A transition $t \in T$ is *enabled* in some marking μ iff $\forall p \in I(t)$, $\mu(p) \geq w((p,t))$ in μ. ($I(t)$ denotes the set of input-places of $t \in T$). $E(\mu)$ is the set of all transitions enabled in μ. When a transition $t \in T$ fires in μ, it creates a new marking μ', such that $\forall p \in I(t) \cup O(t) \;\; \mu'(p) = \mu(p) - w((p,t)) + w((t,p))$. ($O(t)$ denotes the set of output-places of $t \in T$). The firing of t in the marking $\mu^{(i)}$ is denoted by $\mu^{(i)} \xrightarrow{t} \mu^{(i+1)}$.

Definition 1 (Reachable Marking) *A marking $\mu^{(j)}$ is reachable from a marking $\mu^{(i)}$ in the PN $\mathcal{P}$ iff there exists a sequence of transitions $\sigma = (t_k, t_l, \ldots, t_m)$ such that:*

$$\mu^{(i)} \xrightarrow{t_k} \mu^{(i+1)} \xrightarrow{t_l} \ldots \xrightarrow{t_m} \mu^{(j)}.$$

For short, we denote that the marking $\mu^{(j)}$ is reachable from a marking $\mu^{(i)}$ by $\mu^{(i)} \xrightarrow{\sigma} \mu^{(j)}$, where $\sigma = (t_k, t_l, \ldots)$ is also called a *firing sequence*. The set of all markings reachable from $\mu^{(0)}$ is denoted by $R(\mu^{(0)})$, the set of all possible firing sequences in $\mathcal{P}$is written as $\Sigma(\mathcal{P}, \mu^{(0)})$.

The problem of finding for a given $\mu^{(i)}$ whether $\mu^{(i)} \in R(\mu^{(0)})$ is known as the *reachability problem* for Petri nets.

In this paper we shall primarily look at the reachability problem for *reversible* Petri nets [2]:

A *reversible* Petri net is a Petri net, $\mathcal{P} = (P, T, F, w, \mu^{(0)})$ where $F = F^f \cup F^b$, F^f denoting the *forward flow relation*, and $F^b = (F^f)^{-1}$ denoting the *backward flow relation*, derived by reversing the direction of arcs in F^f. A transition $t \in T$ is *enabled* in some marking μ in *forward mode* iff $\forall p \in I(t)$, $(p,t) \in F^f$, $\mu(p) \geq w((p,t))$ in μ, and in *backward mode* iff $\forall p \in O(t)$, $(p,t) \in F^b$, $\mu(p) \geq w((p,t))$ in μ. An enabled transition can fire in the respective mode, creating a new marking in the accordingly.

3 Review of the Theory of Gröbner bases

In this section we give a self-contained review of the main results of the theory of Gröbner bases, see [5, 3] for a detailed exposition . We assume that the reader is familiar with the concept of polynomial over a field.

Let $\sim \subseteq T \times T$ be an equivalence relation on a decidable set T.

Definition 2 (Canonical Simplifier) *An algorithm S, with inputs and outputs in T, is a canonical simplifier for $\sim$ on T iff for all f, $g \in T$:*

(1) $S(f) \sim f$ and

(2) if $f \sim g$ then $S(f) = S(g)$.

Therefore, if a canonical simplifier S is given, the decision problem $f \sim g$, is reduced to checking the syntactical equivalence $S(f) = S(g)$. One can also show that if the equivalence relation $\sim$ is decidable then

[2] Note that this is not related to the classical definition of reversibility, which requires for each marking $\mu^{(i)} \in R(\mu^{(0)})$ there exists a firing sequence σ s.t. $\mu^{(i)} \xrightarrow{\sigma} \mu^{(0)}$.

there exists a canonical simplifier S for $\sim$. In this paper, we will reduce the reachability problem in the class of reversible and cyclic Petri nets to the problem of deciding congruence modulo a polynomial ideal. For this case, the canonical simplification algorithm employs the "Gröbner basis" algorithm described in [4].

We will first fix some notational conventions that we will adopt throughout.

$K[x_1, \ldots, x_n]$	ring of polynomials in n variables over K, a field.
f, g, h, k	polynomials in $K[x_1, \ldots, x_n]$.
$\mathcal{F}, \mathcal{G}$	finite subsets of $K[x_1, \ldots, x_n]$.
s, t, u	power products of the form $x_1^{e_1} \cdots x_n^{e_n}$, $e_1, \ldots, e_n \in \mathbb{N}$.
$\text{Ideal}(\mathcal{F})$	ideal generated by $\mathcal{F} = \{\sum h_i f_i \mid h_i \in K[x_1, \ldots, x_n], f_i \in \mathcal{F}\}$.

The goal of this section is to provide a canonical simplifier for the congruence modulo the polynomial ideal $\text{Ideal}(\mathcal{F})$, denoted by $\equiv_{\mathcal{F}}$ and defined by:

Definition 3 (Congruence $\equiv_{\mathcal{F}}$) *For all $\mathcal{F}$, f, g:*

$$f \equiv_{\mathcal{F}} g \qquad \textit{iff} \qquad f - g \in \text{Ideal}(\mathcal{F}).$$

Let $<_T$ be a total order on the power products satisfying the two properties:

(T1) $1 <_T t$ for all $t \neq 1$,

(T2) if $s <_T t$ then $s \cdot u <_T t \cdot u$.

Such an ordering is called "admissible'. One example of admissible ordering is the *lexicographical order* induced by the variable order $x < y$:

$$1 <_T x <_T x^2 <_T \ldots <_T y <_T xy <_T x^2y <_T \ldots <_T y^2 <_T xy^2 <_T \ldots$$

Let $<_T$ be an arbitrary but fixed admissible ordering. Then, with respect to the ordering $<_T$, it is possible to define:

$\text{cf}(g, t)$	coefficient of t in g.
$\text{lpp}(f)$	leading power product, maximal power product with nonzero coefficient in f.
$\text{lcf}(f)$	leading coefficient, coefficient of $\text{lpp}(f)$.

The primary notion of *reduction (modulo F)* of a polynomial g is a generalization of the polynomial division.

Definition 4 (Reduction modulo $\mathcal{F}$) *g reduces to h (modulo $\mathcal{F}$), denoted by $g \rightarrow_{\mathcal{F}} h$, iff there exist a polynomial $f \in \mathcal{F}$ and a power product u such that:*

$$\text{cf}(g, u \cdot \text{lpp}(f)) \neq 0 \quad \textit{and} \quad h = g - \frac{\text{cf}(g, u \cdot \text{lpp}(f))}{\text{lcf}(f)} \cdot u \cdot f.$$

Then the following lemma states the connection between congruence and reduction.

Lemma 1 *For all $\mathcal{F}$, f, g:*

$$f \equiv_{\mathcal{F}} g \qquad iff \qquad f \overset{*}{\leftrightarrow}_{\mathcal{F}} g$$

where $\overset{*}{\leftrightarrow}_{\mathcal{F}}$ be the reflexive, transitive and symmetric closure of $\rightarrow_{\mathcal{F}}$.

Definition 5 (Normal Form modulo $\mathcal{F}$) *h is a normal form of f (modulo $\mathcal{F}$), denoted by $f \overset{*}{\rightarrow}_{\mathcal{F}} h$, iff there exists a sequence of reductions of length ≥ 0 such that:*

$$f \rightarrow_{\mathcal{F}} \ldots \rightarrow_{\mathcal{F}} \; h$$

and h does not reduce (modulo $\mathcal{F}$) any further.

In general, normal forms modulo an arbitrary $\mathcal{F}$ are not unique. Sets of polynomials which have the property of ensuring the uniqueness of the normal forms have a special name.

Definition 6 (Gröbner basis) *$\mathcal{G}$ is a Gröbner basis iff for all g, h_1, h_2:*

if $g \overset{*}{\rightarrow}_{\mathcal{G}} h_1$ and $g \overset{*}{\rightarrow}_{\mathcal{G}} h_2$, then $h_1 = h_2$.

Gröbner bases are computationally interesting because many problems in polynomial ideal theory can be solved algorithmically by using the following property called *classical characterization* of Gröbner bases:

Lemma 2 *$\mathcal{G}$ is a Gröbner basis iff for all f, g:*

$$f \equiv_{\mathcal{G}} g \qquad \textit{iff} \qquad \textit{there exists } h \textit{ such that } \; f \overset{*}{\rightarrow}_{\mathcal{G}} h \quad \textit{and} \quad g \overset{*}{\rightarrow}_{\mathcal{G}} h$$

By the *GB* algorithm, known in the literature as "Buchberger algorithm" or "Gröbner bases" algorithm, we can find for any set $\mathcal{F}$, a Gröbner basis $\mathcal{G}$ such that $\mathrm{Ideal}(\mathcal{F}) = \mathrm{Ideal}(\mathcal{G})$. The algorithm is based on the notion of *S-polynomial* of two polynomials.

Definition 7 (S-polynomial of f_1 and f_2) *The S-polynomial corresponding to f_1 and f_2 is:*

$$SPol(f_1, f_2) := \frac{lcm(\mathrm{lpp}(f_1), \mathrm{lpp}(f_2))}{\mathrm{lpp}(f_1)} \cdot f_1 - \frac{\mathrm{lcf}(f_1)}{\mathrm{lcf}(f_2)} \cdot \frac{lcm(\mathrm{lpp}(f_1), \mathrm{lpp}(f_2))}{\mathrm{lpp}(f_2)} \cdot f_2.$$

The correctness of the GB algorithm relies on the *algorithmic characterization* of Gröbner bases.

Theorem 1 *$\mathcal{F}$ is a Gröbner basis iff for all f_1, $f_2 \in \mathcal{F}$, $SPol(f_1, f_2) \overset{*}{\rightarrow}_{\mathcal{F}} 0$.*

The Gröbner basis algorithm takes as input a set of polynomials and an admissible term ordering and returns a set of polynomials which is a Gröbner basis for the input set, namely it generates the same ideal. The algorithm can be modified in various ways in order to achieve better performances and a lot of research has been done for finding the "best" ordering and the best reduction strategy. Anyway, in its most essential form, it has the following simple structure

---

$GB\ \ (\mathcal{F}, <_T; \mathcal{G})$

Input: $\mathcal{F}$ a set of polynomials.
$<_T$ an admissible term ordering.

Output: $\mathcal{G}$ a Gröbner basis such that $\mathrm{Ideal}(\mathcal{F}) = \mathrm{Ideal}(\mathcal{G})$.

(1) [Initialization.]
$\mathcal{G} := \mathcal{F}$;
$B := \{(f_1, f_2) \mid f_1, f_2 \in \mathcal{G}, f_1 \neq f_2\}$;

(2) [Loop.]
While $B \neq \emptyset$ do

(2.1) $(f_1, f_2) :=$ a pair in B;
(2.2) $B := B \setminus \{(f_1, f_2)\}$;
(2.3) $h :=$ NORMAL_FORM$(\mathcal{G}, SPol(f_1, f_2))$;
(2.4) if $h \neq 0$ then

(2.a) $B := B \cup \{(g, h) \mid g \in \mathcal{G}\}$;
(2.b) $G := \mathcal{G} \cup \{h\}$;

(3) [Return.]
Return $\mathcal{G}$.

Now, consider the problem of deciding congruence modulo a polynomial ideal:

GIVEN: $\mathcal{F}$, f and g,
DECIDE: $f \equiv_{\mathcal{F}} g$ modulo $\mathrm{Ideal}(\mathcal{F})$.

Suppose that $\mathcal{G}$ is the Gröbner basis of the ideal generated by $\mathcal{F}$, then instead of deciding $f \equiv_{\mathcal{F}} g$, we can decide $f \equiv_{\mathcal{G}} g$. By Lemma 2, this is equivalent to testing whether the normal forms of $\mathcal{F}$ and g modulo $\mathcal{G}$ are syntactically identical. Hence, the algorithm that computes the normal form of $\mathcal{F}$ (modulo $\mathcal{G}$) is a canonical simplifier for the congruence $\equiv_{\mathcal{F}}$ modulo $\mathrm{Ideal}(\mathcal{F})$, if $\mathcal{G}$ is the Gröbner basis of $\mathrm{Ideal}(\mathcal{F})$.

In the next section, we will show that the reachability problem can, in some cases, be transformed into the problem of deciding congruence modulo a polynomial ideal.

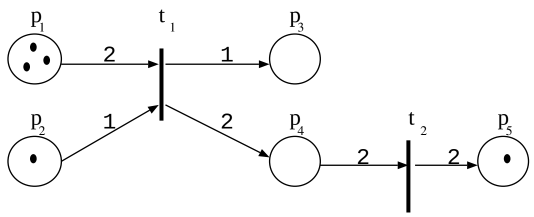
4 Translation of Reachability

The decision method for verifying reachability is based on the fact that there exists a way of translating places and transitions into power products (more generally, one may consider generators of a commutative semigroup) such that the reachability problem becomes a problem in polynomial ideal theory.

Informally, if one assigns a variable (generator) to each place of a reversible Petri net, then a transition can be viewed as a polynomial. If the net is not reversible, then we must keep track of the direction of the transitions and we should emply rewrite rules instead of polynomials. The power products of this polynomial involve the variables corresponding to the input/output places with degree defined by the respective arc's weight. Markings can also be interpreted as power products, namely by taking the marking vector as degree vector of the power product describing the set of all places.

Once this transformation is done, the reachability problem in a Petri net PN between a marking $\mu^{(i)}$ and a marking $\mu^{(j)}$ is the problem of deciding whether the power product of $\mu^{(i)}$ rewrites, using the rules defined by the transitions in PN, to the power product of $\mu^{(j)}$.

In order to show the main idea, let us apply the algebraic transformation to a very simple example. Consider the Petri net:



Firstly, we have to assign variables to the places $p_1, p_2, \ldots, p_5$ in the Petri net: say, $x_1, x_2, \ldots, x_5$ respectively. The transitions correspond to the polynomials:

(f_1) $x_1^2 x_2 - x_3 x_4^2$

(f_2) $x_4^2 - x_5^2$

The initial marking $\mu^{(i)}$ of the net is described by the power product: $x_1^3 x_2 x_5$. A marking that corresponds to a power product obtained by a sequence of reductions modulo f_1 and f_2 from $x_1^3 x_2 x_5$, is reachable from the initial marking. For instance, $x_1^3 x_2 x_5 - (x_1 x_5) f_1 = x_1 x_3 x_4^2 x_5$, hence the marking corresponding to $x_1 x_3 x_4^2 x_5$ is reachable from the initial marking by transition t_1.

Now we formally define how to transform the problem of reachability in a reversible Petri net, to a problem in polynomial ideal theory that can be decided by an application of the Gröbner basis algorithm.

Definition 8 (Polynomial Translation of PN) *Given a reversible PN, $\mathcal{P} = (P, T, F, w, \mu^0)$ with n places in P and m transitions in T, we associate, for $1 \le i \le n$ and $1 \le j \le m$:*

(1) to each place $p_i \in P$, a variable x_i;

(2) to each transition $t_j \in T$, a polynomial $f_j : x_1^{e_1} x_2^{e_2} \dots x_n^{e_n} - x_1^{d_1} x_2^{d_2} \dots x_n^{d_n}$, where

$$e_i = \begin{cases} w((p_i, t_j)) & \textit{if } (p_i, t_j) \in F \\ 0 & \textit{otherwise} \end{cases} \qquad \textit{(input degree vector)}$$

$$d_i = \begin{cases} w((t_j, p_i)) & \textit{if } (t_j, p_i) \in F \\ 0 & \textit{otherwise} \end{cases} \qquad \textit{(output degree vector).}$$

The algorithm that performs this translation is straightforward. We may do better than simply translating the PN by computing the Gröbner basis of the set of polynomial translations of the transitions. This has the advantage that the set of polynomials associated with the PN has a "nice" behavior with respect to the reachability problem, in fact, it is a canonical simplifier for the induced equivalence.

TRANSLATE $(P, T, F, w; \mathcal{G}_\mathcal{P})$

Input: P the set of places of $\mathcal{P}$. T the set of transitions of $\mathcal{P}$. F the input-output relation of $\mathcal{P}$. w the weight function of $\mathcal{P}$.

Output: $\mathcal{G}_\mathcal{P}$ is the Gröbner basis (wrt $>_\mathcal{P}$) of the polynomial translation $\mathcal{F}_\mathcal{P}$.

(1) [Translate the places.]
Assign variable x_i to each place p_i in P.

(2) [Translate the transitions.]
Assign polynomial f_j to each transition t_j in T.

(3) [Compute the Gröbner basis.]
$\mathcal{F}_\mathcal{P} := \{f_j \mid f_j \text{ is the pol. translation of } t_j \in T\}$
$GB(\mathcal{F}_\mathcal{P}, >_\mathcal{P}; \mathcal{G}_\mathcal{P})$;

(4) [Return.]
Return $\mathcal{G}_\mathcal{P}$.

Moreover, the polynomial translation after the Gröbner basis computation still corresponds to a PN because the generated polynomials are of the same form as the polynomials arising from the translation.

Lemma 3 *Let $\mathcal{F}$ be a set of polynomials of the form $f_i : m_i^{(1)} - m_i^{(2)}$, where $m_i^{(1)}$ and $m_i^{(2)}$ are power products. Then, also the polynomials in the Gröbner basis of $\mathcal{F}$ with respect to some admissible ordering, are of the same form.*

Proof. We have to show that the polynomials that arise from the computation of the Gröbner basis of $\mathcal{F}$ are of the special form $f_i : m_i^{(1)} - m_i^{(2)}$, where $m_i^{(1)}$ and $m_i^{(2)}$ are power products. Let us consider the S-polynomials of a pair f_i and f_j:

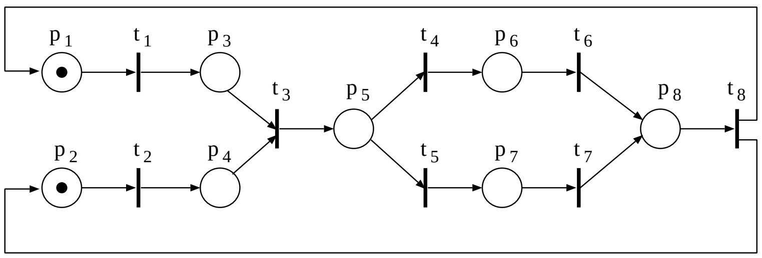
$$SPol(f_1, f_2) = u_1 \cdot (m_i^{(1)} - m_i^{(2)}) - u_2 \cdot (m_j^{(1)} - m_j^{(2)}),$$

where $u_1 \cdot m_i^{(1)} = u_2 \cdot m_j^{(1)}$, and u_1 u_2 are power products. This gives:

$$SPol(f_1, f_2) = -u_1 \cdot m_i^{(2)} + u_2 \cdot m_j^{(2)}$$

which is either the zero polynomial or again of the form we want. It remains to show that the reductions also preserve this form. This is straightforward if we consider that a reduction modulo polynomials of the above form, is basically like a S-polynomial computation where $u_1 = 1$.

As example, the polynomial translation for the conflict free net:



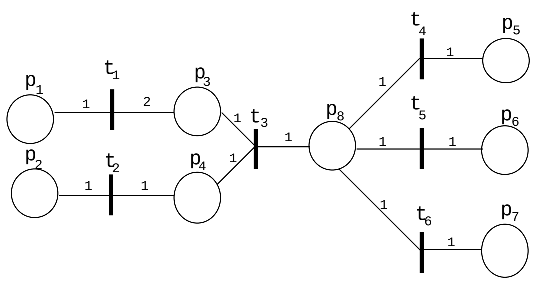
consists of the set of polynomials:

$$F = \{x_1 - x_3, x_2 - x_4, x_3x_4 - x_5, x_5 - x_6, x_5 - x_7, x_6 - x_8, x_7 - x_8, x_8 - x_1x_2\}.$$

Their Gröbner basis, wrt the lexicographical order, is the set:

$$G = \{x_1 - x_3, x_2 - x_4, x_3x_4 - x_8, x_5 - x_8, x_6 - x_8, x_7 - x_8\}$$

which corresponds to the PN:



In the same way as we have translated the PN, we may translate a marking.

Definition 9 (Polynomial Translation of a Marking) *Given a reversible and cyclic PN, $\mathcal{P} = (P, T, F, w, \mu^0)$, the polynomial translation of a marking μ (generated by a marking function $\mathcal{M} : P \rightarrow I\!N^0$), is a a power product $m : x_1^{\mathcal{M}(p_1)} x_2^{\mathcal{M}(p_2)} \ldots x_n^{\mathcal{M}(p_n)}$, where $x_1, \ldots, x_n$ is the polynomial translation of the places of $\mathcal{P}$.*

Since, in a symmetric and cyclic PN, reachability is reflexive and symmetric (if $\mu^{(j)}$ is reachable from $\mu^{(i)}$, then also $\mu^{(i)}$ is reachable from $\mu^{(j)}$), we may define an algebraic equivalence relation $\sim_{\mathcal{P}}$ on the set of markings, induced by reachability, by:

$$\mu^{(i)} \sim_{\mathcal{P}} \mu^{(j)} \qquad \text{iff} \qquad \mu^{(j)} \text{ is reachable from } \mu^{(i)}.$$

Let $\mathcal{P}$ be a symmetric and cyclic PN. Consider the set $\mathcal{F}_{\mathcal{P}}$ of polynomials f_j associated to the $\mathcal{P}$ as described above. If we fix a variable ordering, say $x_1 < x_2 < \ldots < x_n$, and an admissible term ordering, say $<_{\mathcal{P}}$, we can speak of reduction modulo $\mathcal{F}_{\mathcal{P}}$ and of Gröbner basis $\mathcal{G}_{\mathcal{P}}$ of $\mathcal{F}_{\mathcal{P}}$. Let m_i, m_j be the polynomial translations of markings $\mu^{(i)}$, $\mu^{(j)}$ in a PN described by polynomials $\mathcal{F}_{\mathcal{P}}$. Then, it holds:

$$\mu^{(i)} \sim_{\mathcal{P}} \mu^{(j)} \qquad \text{iff} \qquad m_i \overset{*}{\leftrightarrow}_{\mathcal{F}_{\mathcal{P}}} m_j\,.$$

By the results presented in Section 3, we obtain the following result.

Corollary 1 *Let $\mathcal{F}_{\mathcal{P}}$ be the polynomial translation of $\mathcal{P}$, $\mathcal{G}_{\mathcal{P}}$ be its Gröbner basis with respect to an admissible ordering, and $\mu^{(j)}$ $\mu^{(i)}$ markings in $\mathcal{P}$. Then:*

$$\mu^{(i)} \sim_{\mathcal{P}} \mu^{(j)} \qquad \textit{iff} \qquad \textit{there exists } m \textit{ such that} \quad m_i \overset{*}{\rightarrow}_{\mathcal{G}_{\mathcal{P}}} m \quad \textit{and} \quad m_i \overset{*}{\rightarrow}_{\mathcal{G}_{\mathcal{P}}} m,$$

where m_i and m_j are the polynomial translations of $\mu^{(j)}$, $\mu^{(i)}$.

In view of the corollary, we can formulate a decision algorithm for the reachability equivalence $\sim_{\mathcal{P}}$ in the PN $\mathcal{P}$ as described below. It takes as input a canonical translation of a PN, a pair of markings and returns "`yes`" when one marking is reachable from the other or returns "`no`" in the opposite case. The algorithm needs Buchberger's *GB* algorithm in order to transform the set of polynomials corresponding to the given Petri net, to a Gröbner basis, hence, it uses this Gröbner basis as canonical simplifier for deciding reachability. We observe that the Gröbner basis computation is carried out only one time for the PN under consideration, after that, any reachability question in this net can be answered by this decision algorithm.

REACHABLE $(\mathcal{G}_{\mathcal{P}}, \mu^{(i)}, \mu^{(j)}; u)$

Input: $\mathcal{G}_{\mathcal{P}}$ is the Gröbner basis of a polynomial translation $\mathcal{F}_{\mathcal{P}}$ wrt $>_{\mathcal{P}}$.
$\mu^{(i)}$, $\mu^{(j)}$ are markings of PN.

Output: u is '`yes`' if $\mu^{(i)}$ is reachable from $\mu^{(j)}$ in PN and is '`no`' otherwise.

(1) [Polynomial Translation.]

Translate $\mu^{(i)}$ and $\mu^{(j)}$ to polynomials p_i and p_j.

(2) [Canonical Simplification.]

Compute q_i and q_j, the normal form modulo $\mathcal{G}_{\mathcal{P}}$ of p_i and p_j.

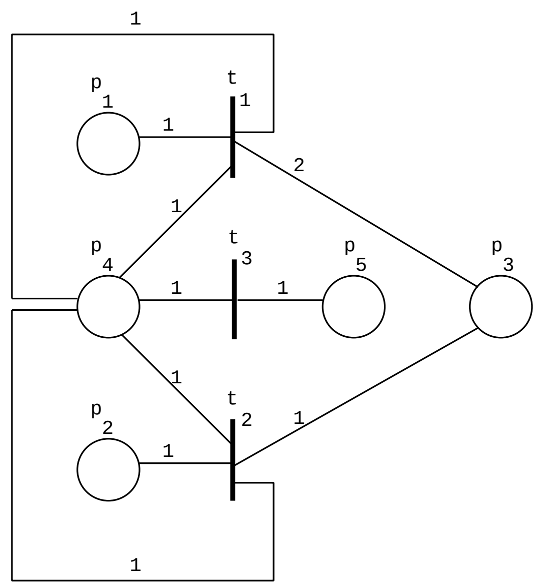
(3) [Return.]

If q_i is syntactical identical to q_j, return "`yes`" otherwise return "`no`".

---

5 MAPLE Simulation

Consider the reversible Petri net:



The polynomial translations of the transitions are:

$$p_1 : x_1x_4 - x_3^2x_4; \qquad p_2 : x_2x_4 - x_3x_4; \qquad p_3 : x_4 - x_5.$$

We ask whether the marking corresponding to $x_1^5x_2x_3^3x_4^3x_5^2$ is reachable from the marking corresponding to $x_1^5x_2^2x_3^2x_4^5$. Here below is the simulation of the algorithm *REACHABLE* by the computer algebra system MAPLE, [6].

At first, we input the set of polynomials to the system.

```
> p1:= x1*x4 - x3^2*x4;
```

$$p1 := x1\ x4 - x3^2\ x4$$

```
> p2 := x2*x4 - x3*x4;
```

$$p2 := x2\ x4 - x3\ x4$$

```
> p3 := x4 - x5;
```

$$p3 := x4 - x5$$

We call F the set of these polynomials.

```
> F := [p1,p2,p3];
```

$$F := [\, x1\ x4 - x3^2\ x4,\ x2\ x4 - x3\ x4,\ x4 - x5\,]$$

Then, we must specify the variable ordering. This is done in MAPLE by a list (decreasing order):

```
> T := [x1,x2,x3,x4,x5];
```

$$T := [\, x1,\ x2,\ x3,\ x4,\ x5\,]$$

Using a MAPLE package called "**grobner**", it is possible to compute the Gröbner basis just by calling a function **gbasis**. **gbasis** requires as parameters the set of polynomials, the variable ordering and the term ordering. In this case, we use the purely lexicographical ordering, specified by **plex**.

```
> G := gbasis(F,T,plex);
```

$$G := [\, x5\ x1 - x5\ x3^2,\ x5\ x2 - x5\ x3,\ x4 - x5\,]$$

MAPLE answers with the Gröbner basis:

$$g_1 : x_5x_1 - x_5x_3^2,$$

$$g_2 : x_5x_2 - x_5x_3,$$

$$g_3 : x_3x_4 - x_5.$$

Using the same package, it is possible to compute the normal form modulo a Gröbner basis by invoking the function *normalf* with the polynomial, the Gröbner basis, and the orderings. In this case, let us check the markings to which the polynomials $x_1^5x_2x_3^3x_4^3x_5^2$ and $x_1^5x_2^2x_3^2x_4^5$ correspond:

```
> q1 := normalf(x1^5*x2*x3^3*x4^3*x5^2,G,T,plex);
```

$$q1 := x5^5\ x1^7$$

```
> q2 := normalf(x1^5*x2^2*x3^2*x4^5,G,T,plex);
```

$$q2 := x5^5\ x1^7$$

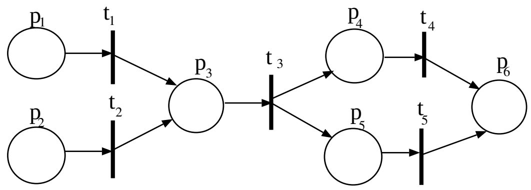
As the normal forms modulo the Gröbner basis are identical, one marking is reachable from the other.

6 Extensions

The decision method that we have described is complete for reversible Petri net. In such case, it will answer whether, in the petri net at hand, there is a firing sequence of transitions for the given pair of markings. In this section, we analyse the possibility of extending the method to a broader class of PNs, that includes also non-reversible nets.

The first observation is that the method presented is a semi-decision algorithm for the general net. Namely, given a PN and given two markings, if the algorithms returns "`no`", then there is no firing sequence connecting the markings. If the algorithm answers "`yes`", then we have to additionally check if the firing sequence exists in the original net (remember that the alg. works over the canonical translation of the net).

In order to exemplify the problem, let us consider this simple PN:



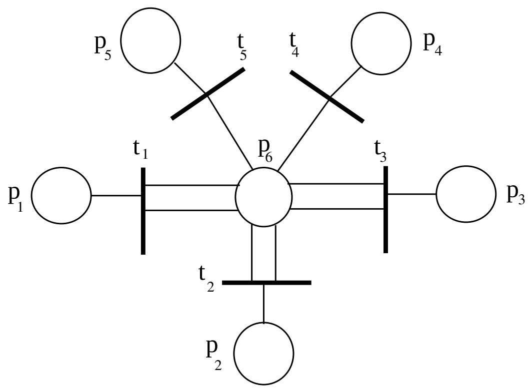
The set of polynomial translations is:

$$F = \{x_1 - x_2, x_2 - x_3, x_3 - x_4x_5, x_4 - x_6, x_5 - x_6\}.$$

Their Gröbner basis with respect to the purely lexicographical ordering given by the variable ordering: $x_6 < x_5 < x_4 < x_3 < x_2 < x_1$, is:

$$G = \{x_1 - x_6^2, x_2 - x_6^2, x_3 - x_6^2, x_4 - x_6, x_5 - x_6\}.$$

Choosing this ordering produces reductions which are compatible with the direction of transitions in the original net. This canonical net looks as follows.



Now, we may check reachability between p_2 and p_1. *REACHABLE* will answer affirmatively because the normal form of their polynomials translation, x_6^2 is the same for both. But, by looking at the original net, we can immediately see that there can be no firing sequence connecting p_1 and p_2. Moreover, *REACHABLE* answers positively also when queried about reachability among the polynomial markings x_2 and x_4x_5 although this is true only in one direction: x_4x_5 is reachable from x_2, but not vice-versa. On the other hand, if we ask whether the polynomial marking x_1x_6 is reachable from x_2, *REACHABLE* reports the correct negative answer.

More research needs to be done investigating the possibility of checking the positive answers given by *REACHABLE*. This involves a study on which admissible ordering to choose for the polynomial translation and on strategies for choosing reductions while computing the normal forms.

7 Concluding Discussion

The main property of a canonical simplifier is that, no matter which kind of reduction it applies, eventually the unique normal form will be reached. In this sense, compared to the indeterminism inherent in the reachability problem, the algorithm we have presented behaves as a deterministic one for the case of reversible PNs.

We should also point out that, although the computation of the Gröbner basis could be very time consuming, in the case of sets of polynomials arising from translations of PNs, the complexity is lower because the degrees are generally linear and the polynomials are very short.

Some work has also been done in extending this approach to the general case but more insight is still needed for devising a complete decision algorithm.

References

[1] J. L. Baer. Modelling architectural features with petri nets. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Lecture Notes in Computer Science: Petri Nets: Applications and Relationships to Other Models of Concurrency, Advances in Petri Nets 1986, Part II, Proceedings of an Advanced Course, Bad Honnef, September 1986*, volume 255, pages 258–277. Springer-Verlag, 1987. NewsletterInfo: 27.

[2] A. A. Bakaev and I. V. Zapatrina. *Problems of the Modelling of Systems Using the Theory of Petri Nets.* Akad. Nauk Ukrain., Inst. Kibernet., Kiev, USSR, Preprint 82-60, 1982. Russian.

[3] T. Becker and V. Weispfenning. *Gröbner Bases: a Computational Approach to Commutative Algebra.* Graduate Texts in Mathematics. Springer–Verlag, 1993.

[4] B. Buchberger. *An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal (German).* PhD thesis, Univ. Innsbruck, Dept. of Math., Innsbruck, Austria, 1965.

[5] B. Buchberger. Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory. In N.K. Bose, editor, *Multidimensional Systems Theory*, pages 184–232. D. Reidel Publishing Company, Dordrecht-Boston-Lancaster, 1985.

[6] B. W. Char, K. O. Geddes, G. H. Gonnet, and S. M. Watt. *Maple User's Guide.* WATCOM Publications Limited, 4th edition, 1985.

[7] M. Hack. *Decidability Questions for Petri Nets.* New York: Garland Publishing, Inc., 1979.

[8] R. M. Keller. Formal verification of parallel programs. *Comm. ACM 19*, (7):371–384, 1976.

[9] S. R. Kosaraju. Decidability of reachability in vector addition systems. In *Proc. of the 14th Annual ACM Symp. on Theory of Computing, San Francisco, May 5–7, 1982*, pages 267–281, 1982. NewsletterInfo: 12.

[10] E. W. Mayr. An algorithm for the general petri net reachability problem. *SIAM J. Comput.*, 13(3):441–460, August 1984. NewsletterInfo: 18,21.

[11] C. A. Petri. *Kommunikation mit Automaten.* Bonn: Institut f¨ur Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962.

[12] C. A. Petri. Fundamentals of a theory of asynchronous information flow. In *Proc. of IFIP Congress 62.*, pages 386–390, Amsterdam, 1963. North Holland Publ. Comp.

[13] G. R. Wheeler. Modelling concurrent systems using petri nets. In *Proceedings of the Australian Software Engineering Conference, Canberra*, pages 151–160, May 1987. NewsletterInfo: 29.

[14] W. M. Zuberek. Timed petri nets in modelling and evaluation of multiprocessor systems. In S. K. Sahni, editor, *Proceedings of the 1987 International Conference on Parallel Processing*, pages 695–698, USA, 1987. Pennsylvania University Press. NewsletterInfo: 29.

[15] R. Z. Zurawski. Modelling distributed real-time systems for performance evaluation using timed petri nets. In *Proceedings of the Conference on Computing Systems and Information Technology, 1989, Sydney, NSW, Australia; Preprints of Papers*, pages 60–63, Barton, ACT, Australia, 1989. Instn. Eng. Australia. NewsletterInfo: 39.