Weighted Petri nets and polynomial dynamical systems

Mikhail Foursov, Christiane Hespel

I R I S A

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

PUBLICATION INTERNE
N° 1781

WEIGHTED PETRI NETS AND POLYNOMIAL DYNAMICAL SYSTEMS

MIKHAIL V. FOURSOV AND CHRISTIANE HESPEL

ISSN 1166-8687

IRISA
CAMPUS UNIVERSITAIRE DE BEAULIEU - 35042 RENNES CEDEX - FRANCE

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES
Campus de Beaulieu – 35042 Rennes Cedex – France
Tél. : (33) 02 99 84 71 00 – Fax : (33) 02 99 84 71 71
http://www.irisa.fr

Weighted Petri nets and polynomial dynamical systems

Mikhail V. Foursov * and Christiane Hespel **

Systèmes cognitifs

Publication interne n1781 — janvier 2006 — 17 pages

Abstract: In this article, we show that the generating series of polynomial dynamical systems are exactly the generating series of the subclass of weighted Petri nets where each transition has a single input place with arc weight 1. We propose furthermore an algorithm to check whether a given Petri net corresponds directly to a dynamical system. In many cases, different initial markings correspond to different dynamical systems. We finally prove that the place invariants for the Petri nets correspond to scaling Lie symmetries of the corresponding dynamical system, as well as that the invariants of the symmetry group of the dynamical system corresponds to implicit places in the corresponding Petri net.

Key-words: Formal power series, dynamical systems, generating series, weighted Petri nets.

(Résumé : tsvp)

* foursov@irisa.fr
** hespel@irisa.fr

Centre National de la Recherche Scientifique
(UMR 6074) Université de Rennes 1 – Insa de Rennes

Institut National de Recherche en Informatique
et en Automatique – unité de recherche de Rennes

Systèmes dynamiques polynomiaux et réseaux de Petri pondérés

Résumé : Dans cet article, nous montrons que les séries génératrices des systèmes dynamiques polynomiaux sont exactement les mêmes que les séries génératrices d'une sous–classe de réseaux de Petri pondérés, dans lesquels chaque transition a une seule place d'entrée avec le poids de l'arc égal à 1. Nous proposons ensuite un algorithme pour vérifier si un réseau de Petri donné correspond directement à un système dynamique. Dans de nombreux cas, des marquages initiaux différents correspondent à des systèmes dynamiques différents. Nous montrons enfin que les invariants de places dans les réseaux de Petri correspondent aux symétries de Lie de changement d'échelle du système dynamique correspondant, ainsi que les invariants du groupe de symétrie du système dynamique correspondent aux places implicites de réseau de Petri correspondant.

Mots clés : Séries formelles, systèmes dynamiques, séries génératrices, réseaux de Petri pondérés.

1 Introduction

The notion of formal power series in noncommutative variables was introduced by M.P. Schützenberger [14], in relation to automata and formal languages. Many problems from the theory of formal languages use formal power series : for example, arithmetic problems of the theory of formal languages, study of stochastic processes and of the context–free grammars. The formal power series also represent an interesting tool for solving combinatorial problems : enumeration of planar graphs, permutations and rearrangements in monoids.

Two principal families of formal power series have been studied : rational series and a subfamily of them formed by the recognizable and algebraic series. The rational series were also introduced by M.P. Schützenberger who showed that certain properties of rational series in one variable have a good generalization in noncommutative variables. He established the equivalence between the recognizability and the rationality of proper formal power series.

Another application of formal power series lies in the treatment of dynamical systems. M. Fliess [5] developed the idea that the generating series of a system can be used to code the input/output behavior of the system. Rational formal power series are generating series for bilinear dynamical systems and are recognized by weighted finite state automata.

Petri nets were introduced by Petri in 1962 and are widely–used graphical and mathematical tools which can be applied to many discrete distributed systems [9, 11, 12, 13]. They are used to model systems with concurrency and resource sharing, to describe synchronous and asynchronous behavior of systems, as well as to study the mathematical models which govern the behavior of systems. Moreover, they can be also used as a visual–communication aid similar to flow charts, block diagrams, and networks.

In a recent article [8], the class of polynomial dynamical systems was considered. It was shown that their generating series are generated by weighted multi–set grammars and accepted by weighted multi–set automata. This article is a continuation of [8] as it deals with polynomial dynamical systems and shows how their generating series can also be interpreted as generating series of a certain type of Petri nets.

In the section 2, we introduce dynamical systems and their generating series. In the section 3, we introduce weighted Petri nets and their generating series. In the section 4, we describe the main result : the relationship between the weighted Petri nets and polynomial dynamical systems. In the sections 5 and 6, we study when there is a dynamical system corresponding to a given Petri net structure, at least for some initial markings. Finally, in the section 7, we show that there are Petri net properties which have counterparts in the corresponding dynamical system.

2 Preliminaries

Definition 2.1. *An affine dynamical system is a system of ordinary differential equations of the form*

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{v}_0(\mathbf{x}) + \sum_{j=1}^{n} \mathbf{v}_j(\mathbf{x}) u_j(t), \\ s(t) = h(\mathbf{x}(t)), \end{cases} \tag{2.1}$$

where

1. $\mathbf{u}(t) = (u_1(t), \ldots, u_n(t)) \in \mathbb{R}^n$ *is the input vector,*
2. $\mathbf{x}(t) \in \mathcal{M}$ *is the current state, where* $\mathcal{M}$ *is a real differential manifold, often* $\mathbb{R}^m$*,*
3. $\{\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n\}$ *is a family of smooth vector fields on* $\mathcal{M}$*,*
4. $h : \mathcal{M} \to \mathbb{R}$ *is a smooth function called the* ***observation map****,*
5. $s(t) \in \mathbb{R}$ *is the output function.*

We will be working with the *causal functional* that associates to the set of n input functions (also called commands) $\mathbf{u}(t)$ the corresponding output function $y(t)$. To the commands $u_1(t), u_2(t), \ldots, u_n(t)$ we associate the *alphabet* $\mathcal{Z} = \{z_0, z_1, \ldots, z_n\}$ of $(n+1)$ letters, z_0 being associated to the drift (which we will represent as an additional constant input function $u_0(t) \equiv 1$). To every multi–index $I = (i_1, i_2, \ldots, i_k)$ we associate the word $w = z_I = z_{i_1} z_{i_2} \cdots z_{i_k}$. These words form $\mathcal{Z}^*$, the free monoid over $\mathcal{Z}$. (The empty word is denoted by λ.)

The behavior of causal functionals is uniquely described by two noncommutative power series: the generating series and the Chen series.

The *generating series* $G = \sum_{w \in \mathcal{Z}^*} \langle G | z_I \rangle z_I$ of the system [5] is the geometric contribution and it is independent of the input. Its coefficients $\langle G | z_I \rangle$ are obtained by iteratively applying Lie derivatives with respect to the vector fields $\mathbf{v}_i$ to the observation map and evaluating the resulting expression at the initial state $\mathbf{x}_0$:

$$\langle G | z_I \rangle = \langle G | z_{i_1} z_{i_2} \cdots z_{i_k} \rangle = \mathbf{v}_{i_k} \circ \cdots \circ \mathbf{v}_{i_2} \circ \mathbf{v}_{i_q} \circ h \big|_{\mathbf{x}_0}.$$

(The Lie derivative of the function $f(x_1, \ldots, x_n)$ with respect to the vector field $\mathbf{v} = (v_1, \ldots, v_n)$ is defined by $\mathbf{v}(f) = \sum_i v_i \frac{\partial f}{\partial x_i}$.) The generating series completely describes the causal functional. More precisely, two formal power series define the same functional if and only if they are equal [6, 15].

The *Chen series* $\mathcal{C}_u(t) = \sum_{w \in \mathcal{Z}^*} \langle \mathcal{C}_u(t) | z_I \rangle z_I$ measures the input contribution [3, 4], and is independent of the system. The coefficients of the Chen series are calculated recursively by integration using the following two relations:

- $\langle \mathcal{C}_u(t) | \varepsilon \rangle = 1$,
- $\langle \mathcal{C}_u(t) | w \rangle = \int_0^t \langle \mathcal{C}_u(\tau) | v \rangle u_j(\tau) d\tau$ for a word $w = z_j v$.

The causal functional $y(t)$ is then obtained locally as the product of the generating series and the Chen series [7]:

$$y(t) = \langle G || \mathcal{C}_u(t) \rangle = \sum_{w \in \mathcal{Z}^*} \langle G|w \rangle \langle \mathcal{C}_u(t)|w \rangle \tag{2.2}$$

This formula is known as the *Peano–Baker formula*, as well as the *Fliess' fundamental formula.*

3 Weighted Petri nets and their generating series

Definition 3.1. *A weighted Petri net is a sextuple $WPN = (P, T, K, F, W, M_0)$, where*

- $P = \{p_1, \ldots, p_m\}$ *is a finite set of places,*
- $T = \{t_1, \ldots, t_n\}$ *is a finite set of transitions,*
- $K : T \to \mathbb{R}$ *is a transition weight function,*
- $F \subseteq (P \times T) \cup (T \times P)$ *is a set of arcs,*
- $W : F \to \{1, 2, 3, \ldots\}$ *is an arc weight function,*
- $M_0 : P \to \{0, 1, 2, \ldots\}$ *is the initial marking,*
- $P \cap T = \emptyset$ *and* $P \cup T \neq \emptyset$.

The firings in a weighted Petri net work the same way as for "ordinary" non–weighted Petri nets, except, of course, for the weights. Before the start of a computation, its weight is 1. During the firing of the transition t in the net, the current weight of the computation is multiplied by the weight of this transition and also, for each input place p of t, by C_n^k, where n is the number of tokens in the place p and k the weight of the arc from p to the fired transition t.

Example 3.2. An ordinary Petri net can be interpreted as a weighted net by assigning weight 1 to each transition.

Remark 3.3. In what follows, we will slightly relax this "classical" definition of Petri nets by allowing different transitions to have the same label, i.e. the word "set" is replaced by "multi–set" in the definition of the set of transitions T above. Another way to look at it is to equate the transitions the end of a firing sequence.

Definition 3.4. *Let WPN be a weighted Petri net. Then its generating series G is a formal power series defined in the following way :*

- *to the set of transitions $T = \{t_1, \ldots, t_n\}$ corresponds the alphabet of the generating series $\{z_1, \ldots, z_n\}$ (if there are several transitions labeled t_j, they all correspond to the same letter z_j),*

- *to a token in the place p_i corresponds a (real) constant x_i (which may be considered a formal variable),*
- *to the firing sequence firing the transitions $t_{i_1}, \ldots, t_{i_k}$ and ending at the marking $M(p) = (k_1, \ldots, k_m)$ corresponds the monomial $\big(\prod_{i=1}^{m} x_i^{k_i}\big)\big(\prod_{j=1}^{k} K(t_{i_j})\big) z_{i_1} \cdots z_{i_k}$,*
- *the generating series G is the formal sum of the monomials corresponding to all the possible transitions in the net.*

In particular, the support of the generating series is the language of traces of executions of the Petri net, whereas the coefficients code the markings. So the generating series is a generalization of this concept which take into account the different ways to obtain a given word.

Definition 3.5. *A weighted Petri net structure is a quintuple $WPNS = (P, T, K, F, W)$, where the elements are the same as above, except that no initial marking is specified.*

4 Weighted Petri nets and polynomial dynamical systems

Definition 4.1. *A dynamical system is called polynomial if its right–hand side is polynomial in the states.*

Theorem 4.2. *The generating series of polynomial dynamical systems are the same as the generating series of weighted Petri nets having the property that each transition has exactly one input place with arc weight 1. The correspondence is the following ;*

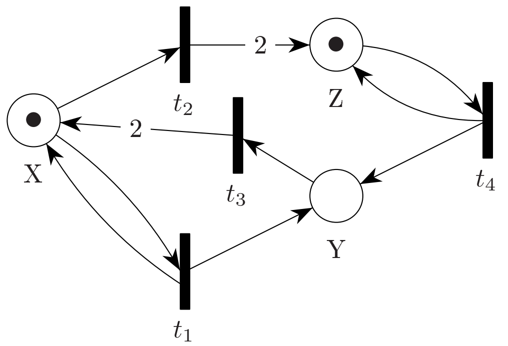
- *the places p_i of the Petri net correspond to the states x_i of the dynamical system,*
- *the transitions t_i correspond to the input functions $u_i(t)$ of the dynamical system (the same transition may appear several times in the net),*
- *the initial marking $(k_1, \ldots, k_m)$ corresponds to the output function $\prod x_j(t)^{k_j}$ of the dynamical system (note that any dynamical system can be rewritten, by adding new states, in such a way so that there is only one monomial in the output),*
- *the summand $Au_k(t) \prod x_j(t)^{n_j}$ on the right–hand side of the equation for the state x_l corresponds to the transition labeled t_k having the following properties : its weight is A, it has a single input place p_l with arc weight 1, its output places are the p_j for j with $n_j \neq 0$, and the weights of the arcs leading to p_j are exactly n_j,*
- *the token values x_i in the generating series of the net correspond to the initial conditions $x_i(0) = x_i$ of the dynamical system.*

The proof is done by recursion. We present the proof in the "classical" case when all the transitions have different labels. The more general case is proven in a similar manner, thus several possible transitions have to be considered at the same time.

The coefficients of the empty word λ are the same. Now let us take a firing sequence $t_{i_1}, \ldots, t_{i_k}$ of length k ending at the marking $M = (l_1, \ldots, l_m)$. Its contribution to the generating series of the Petri net is $C\big(\prod_{i=1}^m x_i^{l_i}\big) z_{i_1} \cdots z_{i_k}$ for some C. By the recursion hypothesis, the corresponding term in the dynamical system is the same, and it was obtained as $(\mathbf{v}_{t_k} \circ \cdots \circ \mathbf{v}_{t_1} \circ s(t)\big|_{t=0}) z_{i_1} \cdots z_{i_k}$.

Now we want to calculate the coefficient of the word $z_{i_1} \cdots z_{i_k} z_{i_{k+1}}$. There are two possibilities. If the transition $t_{i_{k+1}}$ is not firable, this coefficient is zero. The Lie derivative of $w(t) = C \prod_{i=1}^m x_i(t)^{l_i}$ with respect to $\mathbf{v}_{i_{k+1}}$ vanishes, too. If $t_{i_{k+1}}$ is firable leading to the marking $M = (p_1, \ldots, p_m)$, then this coefficient is $C \cdot K(t_{i_{i+1}}) \prod_{i=1}^m x_i^{p_i}$. Applying $\mathbf{v}_{i_{k+1}}$ to $w(t)$, we obtain $C \cdot K(t_{i_{i+1}}) \prod_{i=1}^m x_i(t)^{p_i}$, thus concluding the proof. □

Example 4.3.



$\Longleftrightarrow$

$$\begin{cases} x' = u_1 x y + u_2 z^2 \\ y' = u_3 x^2 \\ z' = u_4 y z \\ s = x z \end{cases}$$

Remark 4.4. Omitting the initial marking for the Petri nets and the output function for the corresponding dynamical systems, we obtain a relationship between the Petri net structures and a class of dynamical systems where the output function is not specified.

5 Weighted Petri net structures and polynomial dynamical systems

If a transition in a Petri net structure has more than one input place, or if an input arc weight at a transition is different from 1, we can study whether there is a polynomial dynamical system in which the states correspond exactly to the places of this net. If this property does not hold in general, it may at least hold for certains initial markings. The following algorithm can be used for a given marking.

Algorithm 5.1.

1. describe all the markings reachable from the initial marking,

2. construct the reachability graph with arc labels kz_i (where k is the weight of the fired transition), the summits are labeled by products of tokens,

3. construct the corresponding dynamical system (which may be infinite if the net is not bounded), where the state s_n corresponds to the summit S_n,

4. correspond to the summit $S_n = \prod X_j^{n_j}$ the product $s_n(t) = \prod x_j(t)^{n_j}$,

5. replace all the $s_n(t)$ in the above dynamical system by these expressions,

6. if there are places such that all the output transitions have only one input place with arc weight 1, add the differential equations corresponding to these places,

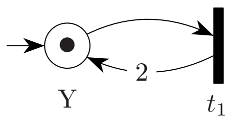
7. using differential algebra, reduce this system (assuming that all $x_j(t) \not\equiv 0$),

8. if the reduced form contains only polynomial differential equations in the states, we obtain the corresponding polynomial dynamical system, otherwise such a polynomial system does not exist.

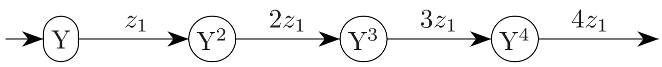
Remark 5.2. To study the same problem for a Petri net structure, this algorithm should be at first applied to several initial markings in order to see whether the behavior changes with the initial marking. If it does not change, the formal general behavior has to be proven by other means.

In the remainder of this section we present 5 examples which are typical of the relationship between Petri net structures and polynomial dynamical systems. The transition weights are all equal to 1.

Example 5.3. Consider the following simple Petri net :



Its reachability graph is

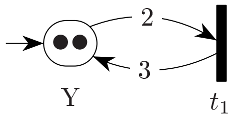


The corresponding infinite dynamical system is

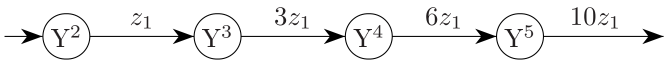
$$(y(t)^n)' = nu_1(t)y(t)^{n+1}, \qquad n \geq 1$$

which reduces to a single polynomial equation $y(t)' = u_1(t)y(t)^2$ with the output condition $s = y(t)$. Since all markings are reachable from this initial marking, this diffential equation is the dynamical system corresponding to the above Petri net structure. □

Example 5.4. Consider the following Petri net :



Its reachability graph is

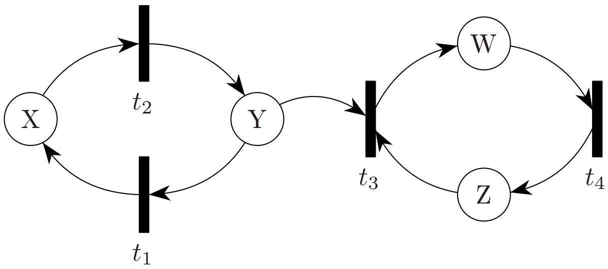


The corresponding infinite dynamical system is

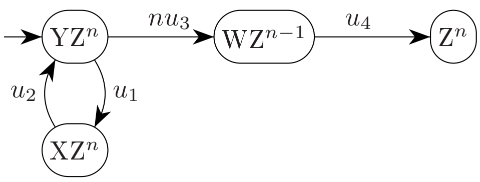
$$(y(t)^n)' = n(n-1)u_1(t)y(t)^{n+1}/2, \qquad n \geq 2$$

which reduces to the system $\{y(t)' = 0, u_1(t)y(t) = 0\}$. There is thus no polynomial dynamical system corresponding to this net. The same is true for any other initial marking. □

Example 5.5. (*the correspondence works with an equivalent Petri net, but only for certain initial markings*) Now, consider the following Petri net structure :



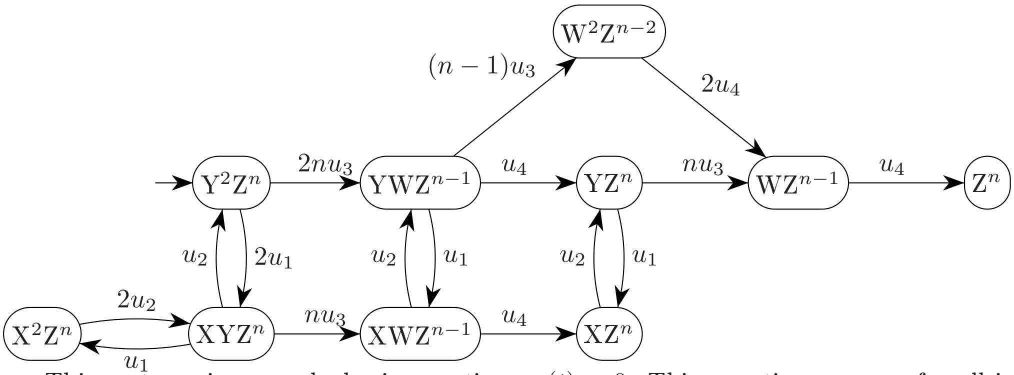
The reachability graph for the initial markings YZ^n is



It corresponds to the system

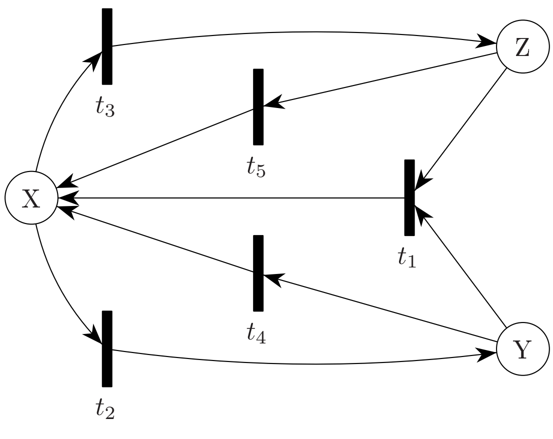
$$\begin{cases} x'(t) &= u_2(t)y(t) \\ z(t)y'(t) &= u_1(t)x(t)z(t) + nu_3(t)w(t) \\ z'(t) &= 0 \\ w'(t) &= u_4(t)z(t) \\ s &= y(t)z(t)^n \end{cases}$$

which is not polynomial, but rational. However, since $z(t)$ is a constant, this is not a real problem for this initial marking, as an equivalent Petri net can be easily found. However, it gives a hint that the writing may be impossible for other initial markings and that it is impossible to devise an equivalent Petri net in which each transition has only one input place. Indeed, for the initial marking Y^2Z^n we have :

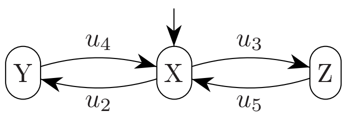


This system gives an algebraic equation $u_3(t) = 0$. This equation appears for all initial markings where $|X|+|Y| > 1$. Therefore, this Petri net structure does not have an equivalent dynamical system. □

Example 5.6. (*Petri net with dead transitions*) Now, consider the following Petri net structure :



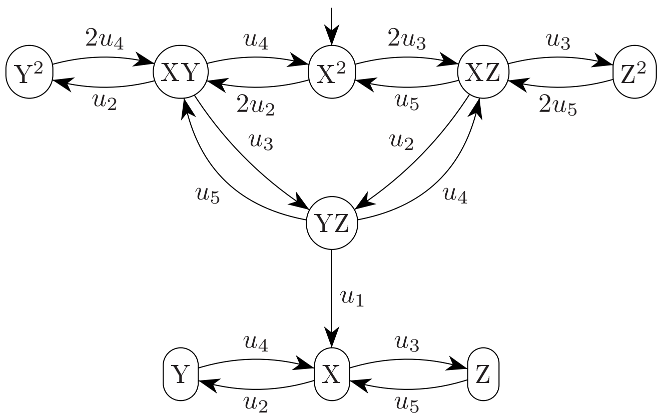
For the initial marking X the reachability graph is



So the corresponding system is

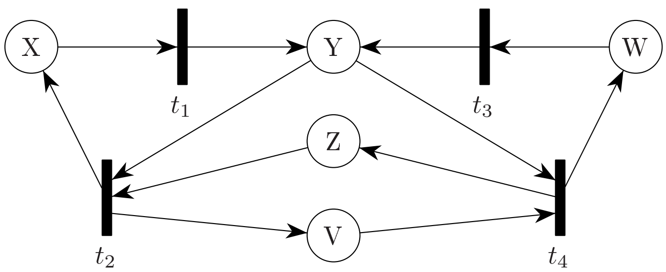
$$\begin{cases} x(t)' &= u_2(t)y(t) + u_3(t)z(t) \\ y(t)' &= u_4(t)x(t) \\ z(t)' &= u_5(t)z(t) \\ \quad s &= x(t), \end{cases}$$

which corresponds to the net above without the transition t_1. This transition is actually dead for this initial marking, so it can be simply ignored. However, this is not the case for other initial markings. For the initial marking XX, the reachability graph is

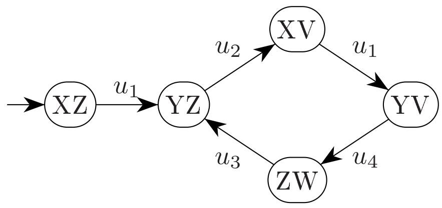


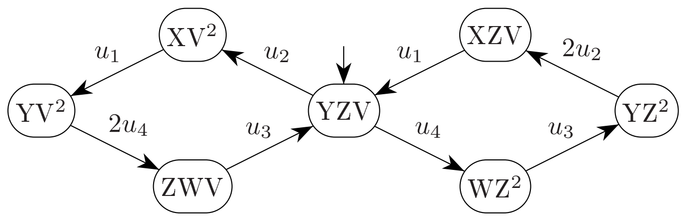
The corresponding system contains the above system, but also a non–differential equation $u_1(t) = 0$.

Example 5.7. (*same compatibility condition for any initial marking*) Finally, let us consider the following Petri net structure with the initial markings XZ and YZV.



The reachability graphs are respectively





The obtained systems are not equivalent, but the reduced system contains the same non–differential constraint

$$u_4(t)z(t)^2w(t) = u_2(t)v(t)^2x(t).$$

in both cases. This compatibility condition is moreover obtained for any initial (live) marking. □

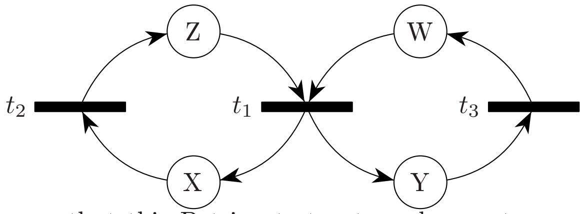
Remark 5.8. However, as it could be noted in the cases above, even when there is no polynomial dynamical system which can be directly obtained from the Petri net structure, the generating series may still be one of a polynomial dynamical system. For example, the generating series of any bounded Petri net is rational. It is obvious from the fact that the reachability graph is in fact a finite weighted automaton and thus the formal power series recognized by it is rational.

6 Classes of Petri net structures directly corresponding to polynomial dynamical systems

Definition 6.1. *A Petri net structure belongs to the class PDS if each transition has only one input place with arc weight 1.*

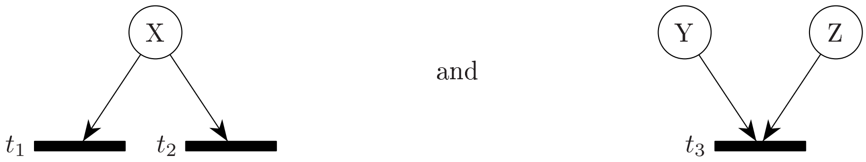
Remark 6.2. The states machines clearly belongs to the class PDS.

Remark 6.3. The class of marked graphs is not a subset of PDS. The following net is the simplest counterexample.

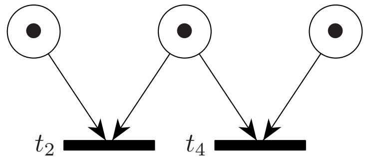


It is not hard to see that this Petri net structure does not correspond to a dynamical system, even though there are (different) dynamical systems corresponding to any initial marking.

The examples show that a Petri net structure may have a corresponding dynamical system if it involves only the transitions of the form

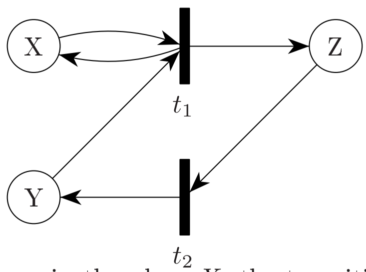


where t_1 and t_2 have no other input places, whereas Y and Z have no other output transitions. The non–differential equations in the examples above come from the transitions where at least one of these properties is violated. For example, the example 5.7, there is the following conflict :

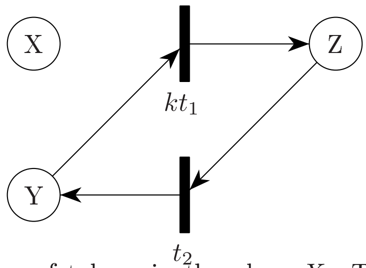


even though this conflict does not happen for the initial marking XZ.

Example 6.4. The existence of several input places to a transition may or may not be an obstruction to rewriting it in polynomial form. Let us for example consider the following Petri net (a marked graph) :



As long as there are some tokens in the place X, the transition t_1 is always firable. So we can replace the above net by the following equivalent net :



where k is the (initial) number of tokens in the place X. The latter net behaves exactly in the same way as the former one, but it corresponds directly to a polynomial dynamical system. □

Conjecture 6.5. *A (weighted) Petri net corresponds to a polynomial dynamical system if and only if it can be transformed to a net of the class PDS by eliminating the self–loops at the nodes that have no other input and output arcs except for these self–loops.*

7 Symmetries of dynamical systems and invariants of Petri nets

Definition 7.1. *Let Σ be a system of ordinary differential equations $\mathbf{x}'(t) = \mathbf{F}(t)$. A Lie symmetry group of the system Σ is a local group of transformations G acting of the space of independent and dependent variables for the system with the property that whenever $\mathbf{x} = \mathbf{f}(t)$ is a solution of Σ, then $\mathbf{x} = g \cdot \mathbf{f}(t)$ is also a solution of the system (here $g \cdot \mathbf{f}(t)$ denoted the group action). In other words, a symmetry group sends solutions of Σ to (other) solutions of Σ.*

In what follows, we will use the term “Lie symmetry” or simply “symmetry” for the infinitesimal generators of the symmetry group. This point of view is equivalent to the above definition. The infinitesimal generators of the symmetry group are vector fields tangent to the hypersurface defined by the system in the jet space (for more details see [10, 2]). The symmetries are found by applying the prolonged vector field to the equations of the system and restricting the obtained formula to the hypersurface defined by the system [10, 2].

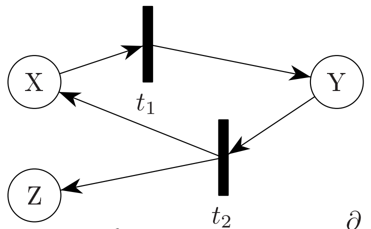
Definition 7.2. *An invariant of a group action is a real–valued function $h(t, \mathbf{x})$ such that $h \equiv g \cdot h$. In other words it does not change under the group action.*

The invariants vanish under the action of the infinitesimal generators of the group action. Therefore, a method of finding them is resolving the system $\mathbf{v}_i(h) = 0$ for all infinitesimal generators $\mathbf{v}_i$ of the group action.

Theorem 7.3. *Consider a weighted Petri net structure from the class PDS. Then $i = \sum k_i N(p_i)$ is an invariant of this net (where $N(p_i)$ is the number of tokens in the place p_i) if and only if $\mathbf{v} = \sum k_i x_i \dfrac{\partial}{\partial x_i}$ is a Lie symmetry of the corresponding dynamical system. Such symmetries are called scaling symmetries.*

The proof of the theorem is straightforward : it is easy to see that the system of linear equations used to find the place invariants for a Petri net is the same as the system used to find the scaling symmetries of the corresponding dynamical system.

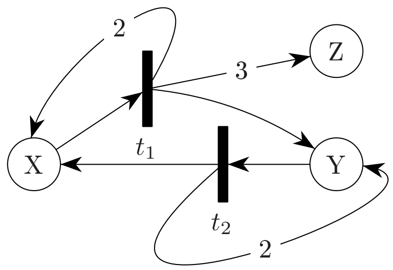
Example 7.4.



$$\Longleftrightarrow \quad \begin{cases} x' = u_1 y \\ y' = u_2 x z \\ z' = 0 \end{cases}$$

$\#X + \#Y$ is the invariant of the net, whereas $x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}$ is a Lie symmetry of the dynamical system. (Note that in the theory of Lie symmetries the vector fields are usually noted $a\frac{\partial}{\partial x} + b\frac{\partial}{\partial y}$ rather than (a, b)).

Example 7.5.



$$\Longleftrightarrow \quad \begin{cases} x' = u_1 x^2 y z^3 \\ y' = u_2 x y^2 \end{cases}$$

$\#X - \#Y$ is the invariant of the net, whereas $x\frac{\partial}{\partial x} - y\frac{\partial}{\partial y}$ is a Lie symmetry of the dynamical system.

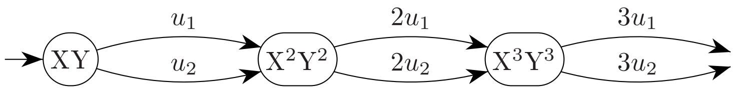
Proposition 7.6. *Consider a weighted Petri net structure from the class PDS. Then* $i = \sum k_i N(p_i) + k_t N(t)$ *is an invariant of this net (where* $N(p_i)$ *is the number of token in the place* p_i *and* $N(t)$ *the number of fired iterations) if and only if* $\mathbf{v} = k_t t \frac{\partial}{\partial t} + \sum k_i x_i \frac{\partial}{\partial x_i}$ *is a Lie symmetry of the corresponding simplified dynamical system in which all the inputs* $u_j(t)$ *are considered constant.*

Proposition 7.7. $\mathbf{v} = \frac{\partial}{\partial x}$ *is a symmetry of a dynamical system if and only if there are no input transitions at the place* X *in the corresponding Petri net.*

Theorem 7.8. *Suppose the monomial* $f(x_1, \ldots, x_n)$ *is the only invariant of the Lie symmetry group of a polynomial dynamical system with arbitrary inputs* $u_j(t)$. *Then* $f(X_1, \ldots, X_n)$ *is a token distribution in the corresponding Petri net which only repeats itself during any firing sequence (in other words it is a implicit place). In the case the only invariant* f *is polynomial, we need to consider the sum of tokens in several Petri nets with the same structure, but different initial markings. In the case of a rational invariant, an equivalent Petri net can be given such that this invariant becomes polynomial.*

The proof is based on a result from the Lie symmetry theory. For a first–order system of ODEs, the first derivative of an invariant can be expressed in terms of all the invariants of the system. As there is only one invariant in our case, and as the system is polynomial, it is straightforward to transpose this result to the Petri nets.

Example 7.9. Let us consider the example 7.5 without the place Z. xy is the only invariant under the symmetry $x\frac{\partial}{\partial x} - y\frac{\partial}{\partial y}$, since $x\frac{\partial(xy)}{\partial x} - y\frac{\partial(xy)}{\partial y} = 0$. The initial marking XY gives the following reachability graph :



Thus the only reachable markings are powers of XY.

Remark 7.10. In the case of several invariants, the situation becomes more complicated. In the Lie symmetry theory, one construct a minimal family of functionally–independent invariants. However, the derivatives of any invariant can be expressed as a function of the other invariants, but not necessarily a polynomial one. This leads us to the following definition.

Definition 7.11. *Let G a transformation group. We call a (finite) family of invariants polynomially independent, if any invariant of G can be expressed as a polynomial in the members of this family. Note that such a family may well be functionnally dependent or functionally independent.*

Theorem 7.12. *Let Σ be a dynamical system and let $\{\mathbf{v}_1, \ldots, \mathbf{v}_s\}$ be its Lie symmetry group. Suppose there exists a family of polynomially–independent polynomial invariants $\{f_1(x_1, \ldots, x_n), \ldots, f_r(x_1, \ldots, x_n)\}$ of this group. Then $f_1(X_1, \ldots, X_n), \ldots, f_r(X_1, \ldots, X_n)$ are implicit places for the corresponding Petri net structure.*

8 Conclusion

In this article we present a new relationship between a subclass of Petri nets and polynomial dynamical systems, via their generating series. It allows us to express several properties of Petri nets in terms of the properties of the corresponding dynamical systems.

A interesting direction to follow is to enlarge this relationship to a more general class of Petri nets and dynamical system. Other direction could be to describe Petri net structures possessing at least some initial markings whose generating series correspond to a polynomial dynamical system.

Acknowledgments

The authors are extremely thankful to Philippe Darondeau for introducing us to the wonderful world of Petri nets, for showing the relationship of the multi–set weighted grammars [8] and the Petri nets, as well as for giving us a large number of very useful hints and insights about the Petri nets.

References

[1] J. Berstel and C. Reutenauer, *Les séries rationnelles et leurs langages*, Masson, 1984.

[2] G.W. Bluman and S. Kumei, *Symmetries and differential equations*, Springer–Verlag, 1989.

[3] K.-T. Chen, *Algebras of iterated path integrals and fundamental groups*, Trans. Am. Math. Soc. **156** (1971), 359–379.

[4] K.-T. Chen, *Iterated path integrals*, Bull. Am. Math. Soc. **83** (1977), 831–879.

[5] M. Fliess, *Fonctionnelles causales non linéaires et indéterminées non commutatives*, Bull. Soc. Math. France **109** (1981), 3–40.

[6] M. Fliess, *On the concept of derivatives and Taylor expansions for nonlinear input/output systems*, in "IEEE Conference on Decision and Control" (San Antonio, Texas), 1983, 643–648.

[7] M. Fliess, *Réalisation locale des systèmes non linéaires, algèbres de Lie filtrées transitives et séries génératrices*, Invent. Math. **71** (1983), 521–537.

[8] M. Foursov and C. Hespel, *Formal power series and polynomial dynamical systems*, technical report IRISA 1691 (2005).

[9] T. Murata, *Petri nets : properties, analysis and applications*, Proceedings of the IEEE **77** (1989), 541–580.

[10] P. Olver, *Applications of Lie groups to differential equations*, Springer–Verlag, 1993.

[11] J. Peterson, *Petri net theory and the modeling of systems*, Prentice–Hall, 1981.

[12] W. Reisig, *Petri nets : an introduction*, Springer–Verlag, 1985.

[13] C. Reutenauer, *Aspects mathematiques des reseaux de Petri*, Masson, 1988.

[14] M.P. Schützenberger, *Un problème de la théorie des automates*, séminaire Dubreil-Pisot **3** (1959–1960).

[15] Y. Wang and E.D. Sontag, *On two definitions of observation spaces*, Syst. Contr. Lett. **13** (1988), 279–289.