26ième ECOLE DE PRINTEMPS D'INFORMATIQUE THEORIQUE,
NOIRMOUTIER. 4-7 MAI 1997

PETRI NETS, (MAX,+) ALGEBRA AND SCHEDULING

Jean Mairesse, LIAFA, CNRS - Université Paris VII

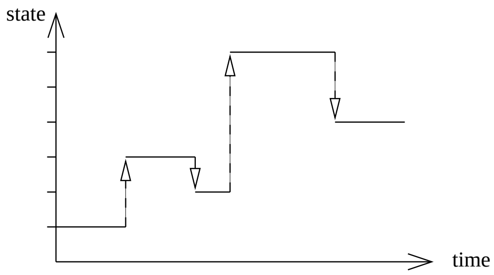
Contents

Introduction

This document was prepared for the *26ième École de Printemps d'Informatique Théorique*. It is largely inspired by the articles [19] and [12]. A short version of [12] is given as an Appendix.

1 Petri Nets

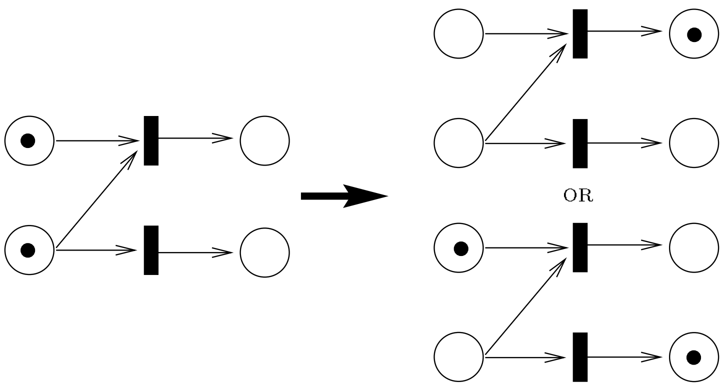
1.1 Discrete Event System



Discrete Event Systems (DES) are defined by opposition to classical systems whose evolution are driven by differential, or difference, equations.
In a SED, the transformations of the system are triggered by punctual 'events' such as: arrival of a customer, arrival of a signal or completion of a task. Natural domains of application are: manufacturing, computer architecture or communication networks.

Physical system	*Graphical representation*	*Algebraic representation*
DES	Queueing network Petri net	GSMP Dynamical system in exotic algebras

1.2 Petri net



Definition 1. A Petri net (PN) is a 4-tuple $\mathcal{G} = (\mathcal{T}, \mathcal{P}, \mathcal{F}, M)$, where :

- $\mathcal{T}$ is a finite set, whose elements are called *transitions*.
- $\mathcal{P}$ is a finite set, whose elements are called *places*.

- $\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ is a relation between places and transitions.

- M is a map $\mathcal{P} \to \mathbb{N}$. The integer $M(p)$ is called the *initial marking* of place p.

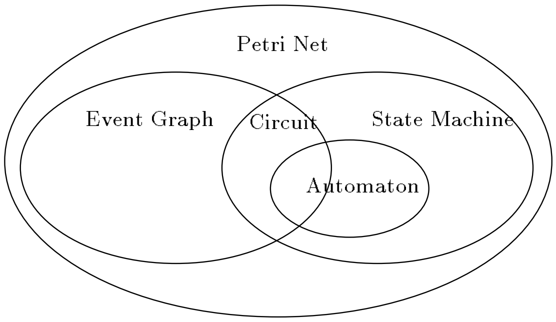
A Petri net is a bipartite directed graph with two different kinds of nodes, places $p \in \mathcal{P}$, (represented by circles) and transitions $a \in \mathcal{T}$ (represented by rectangles). An element of $\mathcal{F}$ is an arc from a place to a transition or from a transition to a place. The marking $M(p)$ is displayed by drawing $M(p)$ tokens in place p.
A Petri net is a dynamic object. The marking M evolves according to the *firing rule*.

1. Transition a is *enabled* at M if there is at least one token in each of its input places.

2. An enabled transition a can *fire*. The firing of a transforms M into M' (written $M \xrightarrow{a} M'$) by removing one token from each of the input places and adding one token in each of the output places of a.

We say that a word $w = a_1 a_2 \cdots a_n \in \mathcal{T}^*$ is a *firing sequence* starting from marking M' if there is a sequence of markings $M' = M_0, \ldots, M_n = M''$ such that transition a_i is enabled at M_{i-1} and $M_{i-1} \xrightarrow{a_i} M_i$ ($M' \xrightarrow{w} M''$). M'' is *reachable* from M' if $\exists w \in \mathcal{T}^*$ such that $M' \xrightarrow{w} M''$. Let $\mathcal{R}(M')$ be the set of markings reachable from M'.
The *language* of $(\mathcal{G}, M)$ is the set L (subset of the free monoid $\mathcal{T}^*$) of firing sequences starting from M.

Monoid: a set equipped with an associative internal law and having a unit element.
Free Monoid: Finite set (alphabet) $\mathcal{A}$. $\mathcal{A}^n$ = set of words of length n on $\mathcal{A}$ and $\mathcal{A}^* = e \cup \mathcal{A} \cup \mathcal{A}^2 \cup \cdots$ and $\mathcal{A}^+ = \mathcal{A}\mathcal{A}^* = \mathcal{A}^*\mathcal{A}$. The monoid $(\mathcal{A}^*, \cdot)$ = *free monoid* on $\mathcal{A}$ ('$\cdot$'= concatenation).



For $x \in \mathcal{T} \cup \mathcal{P}$, we denote by ${}^\bullet p$ and $p^\bullet$ the predecessors and successors of x.

1. An *Event Graph* (EG) is a PN such that $|{}^\bullet p| = |p^\bullet| = 1$ for all $p \in \mathcal{P}$.

2. A *State Machine* (SM) is a PN such that $|{}^\bullet t| = |t^\bullet| = 1$ for all $t \in \mathcal{T}$.

3. An *Automaton* is a SM with a single token, i.e. $\sum_{p \in \mathcal{P}} M(p) = 1$.

4. A *circuit* is a PN such that $|{}^\bullet t| = |t^\bullet| = |{}^\bullet p| = |p^\bullet| = 1$ for all $t \in \mathcal{T}, p \in \mathcal{P}$.

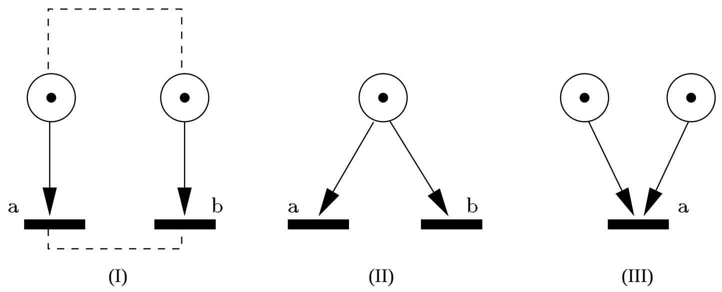


Figure 1: (I) Concurrency (II) Choice (III) Synchronization

SM allow choice but not synchronization whereas EG allow synchronization but not choice, see Fig. 1.
Our definition of an Automaton corresponds to the classical notion of a Deterministic Boolean Automaton (except that we do not specify final states, when needed we assume that all states are final). More classical (but equivalent) presentation is as follows:

3 bis. An *Automaton* is a 4-tuple $(\mathcal{T}, \mathcal{P}, p_0, \delta)$ where $\mathcal{T}$ is a finite alphabet, $\mathcal{P}$ is a finite state space, p_0 is the initial state, and $\delta : \mathcal{P} \times \mathcal{T} \to \mathcal{P}$ is the (partially defined) transition function.

The initial state p_0 corresponds to the place such that $M(p_0) = 1, M(p) = 0, p \neq p_0$. The transition function is defined by $\delta(p, a) = q$ if $(p, a) \in \mathcal{F}$ and $(a, q) \in \mathcal{F}$. The language of the Automaton as defined above via firing sequences coincides with the classical notion defined via accepted words (all states being final).
In Fig. 2: comparison between the classical and the Petri net graphical conventions for Automata.

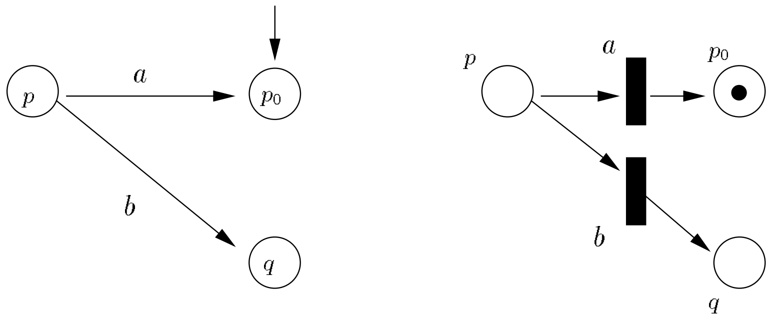


Figure 2: Classical and Petri net representations for Automata

Definition 2. A Petri net $(\mathcal{T}, \mathcal{P}, \mathcal{F}, M)$ with language L is

- *live* if $\forall w \in L, \forall a \in \mathcal{T}, \exists v \in \mathcal{T}^*$ s.t. $wva \in L$ ('transitions can always fire');
- *1-bounded* (or *safe*) if $\forall M' \in \mathcal{R}(M), \forall p \in \mathcal{P},\ M'(p) \leq 1$ ('not more than one token per place').

1.3 Timed Petri net

- $\tau : \mathcal{T} \cup \mathcal{P} \longrightarrow \mathbb{R}^+$;
- $a \in \mathcal{T}$, τ_a = firing time of a;
- $p \in \mathcal{P}$, τ_p = holding time of p.

Several semantics for the firing in timed Petri nets exist in the literature. We consider the following one:

We assume that transition a becomes enabled at instant t. A firing occurs in three steps.

1. At instant t, the firing of a may be *initiated*. If initiated, it removes one token from each input place.
2. One token is added in each of the output place at instant $t + \tau_a$.
3. The token added in place $p \in a^\bullet$ can contribute to the enabling of the transitions in $p^\bullet$ after instant $t + \tau_a + \tau_p$.

Between t and $t + \tau_a$, the tokens can be considered as being 'frozen' in their original input place. The tokens and the transition a can not be involved in any other firing between t and $t + \tau_a$.
With this semantic, if a transition fires, it does so *"as soon as possible"*. The decisions about which transitions are to fire is not based on time considerations. All logically feasible choices can be considered.

2 (Max,+) Semiring and Spectral Theory

2.1 (Max,+) semiring

Definition 3 (semiring).
A set K equipped with two operations $\oplus$ and $\otimes$ is a semiring if $\oplus$ is associative and commutative, $\otimes$ is associative and distributive with respect to $\oplus$, there is a zero element $\mathbb{0}$ ($a \oplus \mathbb{0} = \mathbb{0} \oplus a = a, a \otimes \mathbb{0} = \mathbb{0} \otimes a = \mathbb{0}$) and a unit element $\mathbb{1}$ ($a \otimes \mathbb{1} = \mathbb{1} \otimes a = a$).
A semiring is *idempotent* if $\forall a \in K, a \oplus a = a$.

Definition 4 ((max,+) semiring).
$\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \oplus, \otimes)$ with

- $a \oplus b = \max(a, b),\ a \otimes b = a + b$
- $\mathbb{0} = -\infty$, $\mathbb{1} = 0$

$\mathbb{R}_{\max}$ is an idempotent semiring.

We consider the sets $\mathbb{R}_{\max}^{p \times q}$ of matrices with coefficients in $\mathbb{R}_{\max}$. For matrices A, B of appropriate sizes, we define:

$$(A \oplus B)_{ij} = A_{ij} \oplus B_{ij} = \max(A_{ij}, B_{ij}),$$

$$(A \otimes B)_{ij} = \bigoplus_k A_{ik} \otimes B_{kj} = \max_k (A_{ik} + B_{kj})\,.$$

`Exercise: Verify that` $(\mathbb{R}_{\max}^{k \times k}, \oplus, \otimes)$ `is a semiring.`
In the following the notations "$+,\times$" will stand for the usual addition and multiplication. Nevertheless, we will write ab for $a \otimes b$ whenever there is no possible confusion. For example, for $a \in \mathbb{R}_{\max}$, $a^d = a^{\otimes d} = d \times a$.
Given a set $\mathcal{S}$, we denote by $\mathbb{1}_{\mathcal{S}}$ the $|\mathcal{S}|$-dimensional column vector whose entries are all equal to $\mathbb{1}$.

Definition 5. A *(max,+) Automaton* is a 4-tuple $(\mathcal{A}, Q, I, \mathcal{M})$, where

- $\mathcal{A}$ is a finite set (*alphabet*); Q is a finite set (of *states*); $I \in \mathbb{R}_{\max}^{1 \times Q}$ is the *initial* vector; $\mathcal{M}$ is a morphism $\mathcal{A}^* \to \mathbb{R}_{\max}^{Q \times Q}$.

The morphism $\mathcal{M}$ is uniquely specified by the finite family of $Q \times Q$ matrices, $\mathcal{M}(a), a \in \mathcal{A}$. Then, for a word $w = a_1 \ldots a_n$, we have $\mathcal{M}(w) = \mathcal{M}(a_1 \ldots a_n) = \mathcal{M}(a_1) \ldots \mathcal{M}(a_n)$, the matrix product being interpreted in the (max,+) semiring.

An Automaton is a special case of a (max,+) Automaton (more precisely $(\mathcal{A}, Q, i, \delta) \longleftrightarrow (\mathcal{A}, Q, I, \mathcal{M})$ where $I_i = \mathbb{1}, I_j = \mathbb{0}, j \neq i$, and where $\mathcal{M}(a)_{ij} = \mathbb{1}$ if $\delta(i, a) = j$ and $\mathbb{0}$ otherwise).
A (max,+) Automaton is the specification to the (max,+) semiring of the general notion of a (non-deterministic) Automaton with multiplicities over a semiring, see [2, 10].

Definition 6. A *(max,+) linear system* (of order 1) is a system of the form:

$$x(n) = A \otimes x(n-1)\,,$$

where for $n \in \mathbb{N}$, $x(n) \in \mathbb{R}^k_{\max}$ and $A \in \mathbb{R}^{k\times k}_{\max}$. More generally, we also call *(max,+) linear system* a system of the form:

$$x(n) = A_0 \otimes x(n) \oplus A_1 \otimes x(n-1) \oplus A_2 \otimes x(n-2) \oplus \cdots \oplus A_l \otimes x(n-l)\,.$$

Given a finite set $\mathcal{A}$, a map $x : \mathcal{A}^* \to \mathbb{R}^k_{\max}$ (resp. a map $y : \mathcal{A}^* \to \mathbb{R}_{\max}$) is said to be *(max,+) recognizable* if for $w \in \mathcal{A}^*$

$$x(w) = I \otimes \mathcal{M}(w) \quad (\text{resp. } y(w) = I \otimes \mathcal{M}(w) \otimes F)\;;$$

for some (max,+) Automaton $(\mathcal{A}, Q, I, \mathcal{M})$ with $|Q| = k$ (resp. and a final vector $F \in \mathbb{R}^{k\times 1}_{\max}$).
The map x verifies $x(wa) = x(w) \otimes \mathcal{M}(a)$, hence it can be viewed as an order 1 *(max,+) linear system* whose dynamic is driven by letters.

2.2 Spectral theory

Let A belong to $\mathbb{R}^{k\times k}_{\max}$. The *spectral problem* is to find $u \in \mathbb{R}^k_{\max}$ and $\rho \in \mathbb{R}$ such that:

$$A \otimes u = \rho \otimes u$$

- ρ: eigenvalue, u: eigenvector.

Eigenvalue

Matrix A is irreducible if the graph of A (Def. 9) is strongly connected.

Theorem 1 (Cuninghame-Green [7]). *We consider an irreducible matrix $A \in \mathbb{R}^{k\times k}_{\max}$. Unique eigenvalue $\rho(A)$ given by*

$$\rho(A) = \max_{l=1}^{k} \max_{i_1,\ldots,i_l} \frac{A_{i_1 i_2} + A_{i_2 i_3} + \cdots + A_{i_l i_1}}{l} = \bigoplus_{l=1}^{k} \bigoplus_{i_1,\ldots,i_l} (A_{i_1 i_2} A_{i_2 i_3} \cdots A_{i_l i_1})^{1/l}$$

Eigenvectors

Let $A \in \mathbb{R}^{k\times k}_{\max}$ be irreducible.

Critical circuit : circuit (of the graph of A) whose average weight is $\rho(A)$.
Critical graph : nodes and arcs of A belonging to a critical circuit.
Cyclicity : cyclicity of the critical graph of A.

The spectral behavior of A is determined by:

- Number of maximal strongly connected subgraphs (s.c.s.) of its critical graph.
- Cyclicity.

Matrix A is **scs**j**-cyc**k if it has j strongly connected subgraphs and a cyclicity (of the critical graph) of k.

Matrix A with eigenvalue $\rho(A)$. Normalized matrix $A \longrightarrow \tilde{A}$, $\tilde{A}_{ij} = A_{ij} - \rho(A)$. We have $\rho(\tilde{A}) = \mathbb{1}$. Let us define

$$\tilde{A}^* = \bigoplus_{i=0}^{\infty} \tilde{A}^i, \quad \tilde{A}^+ = \bigoplus_{i=1}^{\infty} \tilde{A}^i . \tag{1}$$

The matrices $\tilde{A}^*$ and $\tilde{A}^+$ are well defined, see Theorem 3.

Theorem 2 (Romanovskii [24], Gondran-Minoux [15]). *Let A be an irreducible matrix, $\tilde{A}$ associated normalized matrix.*

a. For i belonging to the critical graph, $\tilde{A}^+_{.i}$ is an eigenvector of A.

b. $\tilde{A}^+_{.i}$ and $\tilde{A}^+_{.j}$ are different iff i and j belong to two different s.c.s. of the critical graph.

c. Every eigenvector of A is a linear combination of the critical columns $\tilde{A}^+_{.i}$.

Theorem 3 (Cohen-Dubois-Quadrat-Viot [5, 6]). *Let A be irreducible, $\rho(A)$ its eigenvalue, c its cyclicity.*

$$\exists N, \ \forall n \geqslant N, \quad A^{n+c} = \rho(A)^{\otimes c} \otimes A^n .$$

On Fig. 3, we have represented the set of eigenvectors (gray area) of matrix A and also the effect of matrix A on a ball (for a more complicated example, involving a cat, see the contribution of Gaubert in this book).

$$A = \begin{pmatrix} 2 & \mathbb{1} \\ 1 & 2 \end{pmatrix} .$$

Matrix A is scs2-cyc1.

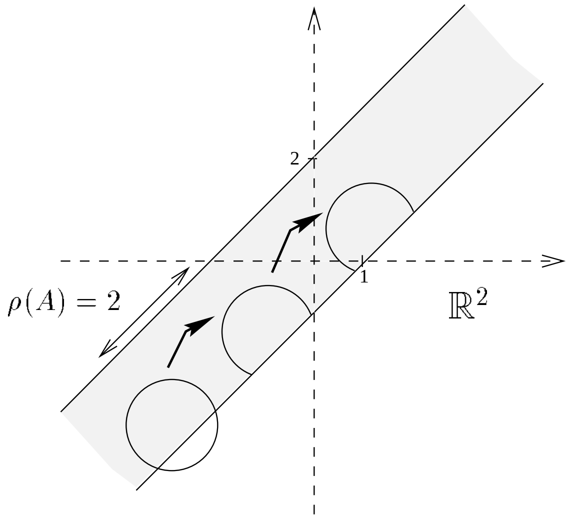


Figure 3: Evolution of a ball caught in a (max,+) dynamic.

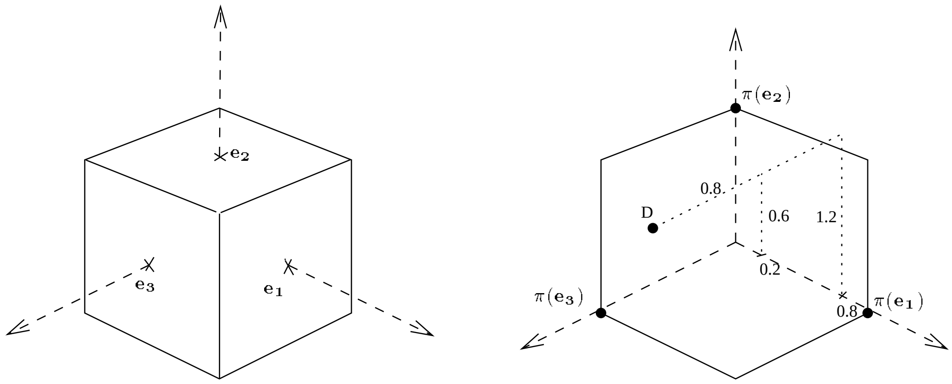


Figure 4: Unit ball of $\mathbb{PR}^3$

Additive projective space

If $u \in \mathbb{R}^k_{\max}$ is an eigenvector, then $\lambda \otimes u, \forall \lambda \in \mathbb{R}$, is also an eigenvector. It motivates the introduction of the following additive projective space.

Definition 7. $\mathbb{PR}^k$ is the quotient of $\mathbb{R}^k$ by the congruence :

$$u, v \in \mathbb{R}^k \quad u \simeq v \Longleftrightarrow \exists a \in \mathbb{R} \text{ s.t. } u = a \otimes v \ \ (u_i = a + v_i, \forall i)\,.$$

$\pi \ : \ \mathbb{R}^k \longrightarrow \mathbb{PR}^k$, canonical projection.

The projection π can be interpreted geometrically as the orthogonal projection on the hyper-space orthogonal to the vector $(1, \ldots, 1)'$. For an illustration, see Fig. 4.
The practical way of representing a point X of $\mathbb{PR}^3$ is to choose a representative ($\in \mathbb{R}^3$) of X and to draw it in the three axes obtained by projection of the orthonormal basis of $\mathbb{R}^3$. The point we obtain does not depend on the chosen representative. The point

D of Figure 4 illustrates this. We have drawn two constructions : one corresponding to $(0.2,\ 0.6,\ 0.8)$ and the other one to $(0.8,\ 1.2,\ 1.4) = 0.6 \otimes (0.2,\ 0.6,\ 0.8)$.

Sets of eigenvectors in the projective space

We can illustrate the shape of sets of eigenvectors in the projective space for matrices of dimension 3, see Mairesse [19]. Here are some examples.

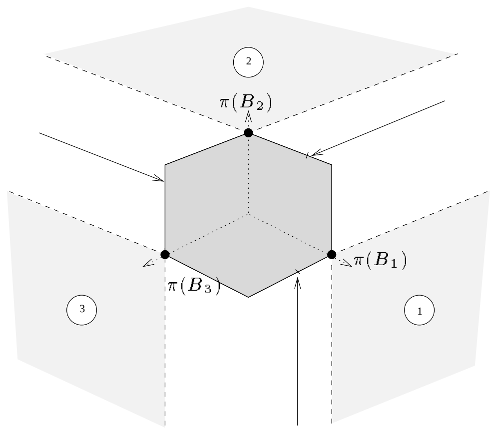


Figure 5: Set of eigenvectors of matrix B

$$B = \begin{pmatrix} \mathbb{1} & . & . \\ . & \mathbb{1} & . \\ . & . & \mathbb{1} \end{pmatrix} ,$$

with the convention $. = -1$. Matrix B is scs3-cyc1. The set of eigenvectors is the dark gray hexagon. If the initial condition x_0 is in the light gray zone number i, then the limit of $\pi(B^n x_0)$ is $\pi(B_i)$. If the initial condition is in one of the white strips, the limit value of $\pi(B^n x_0)$ is given by the arrows.

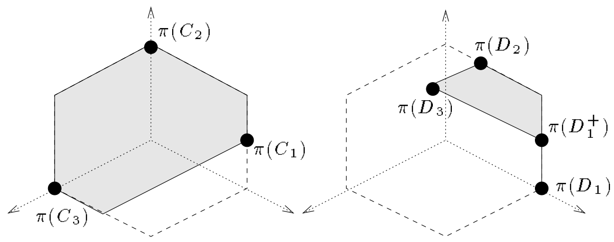


Figure 6: Sets of eigenvectors of matrices C and D

$$C = \begin{pmatrix} \mathbb{1} & . & . \\ -0.5 & \mathbb{1} & . \\ . & . & \mathbb{1} \end{pmatrix} ,\ D = \begin{pmatrix} \mathbb{1} & -0.6 & . \\ . & \mathbb{1} & 0.6 \\ . & . & \mathbb{1} \end{pmatrix} , (.) = -1 \,.$$

Matrices C and D are scs3-cyc1. Their set of eigenvectors are contained in the one of matrix B. In the example of matrix D, the semimodule generated by the columns $< D_1, D_2, D_3 >$ of D strictly contains the set of eigenvectors which is the semimodule $< D_1^+, D_2^+, D_3^+ >$ generated by the columns of D^+ (see Theorem 2). For more details on semimodules, see the contribution of Gaubert in this book.

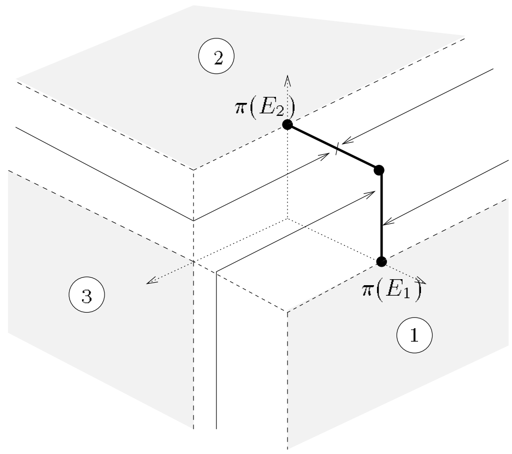


Figure 7: Set of eigenvectors of matrix E

$$E = \begin{pmatrix} . & \mathbb{1} & . \\ \mathbb{1} & . & . \\ . & . & -2 \end{pmatrix}, (.) = -1 \; .$$

Matrix E is scs2-cyc1. The set of eigenvectors is the 'broken segment' $[\pi(E_1), \pi(E_2)]$. The interpretation of the gray zones and of the arrows is the same as above.

Periodic regimes

Apart from the eigenvectors, it is interesting to know the periodic regimes of an irreducible matrix $A \in \mathbb{R}_{max}^{k\times k}$.

Definition 8 (periodic regime). A periodic regime of period d is a set of vectors $\{u1, \ldots, ud\}$ of $\mathbb{R}^k$ verifying $A \otimes (ui) = \rho(A) \otimes (ui+1), \; i = 1, \ldots, d-1$ and $A \otimes (ud) = \rho(A) \otimes (u1)$.

It implies that the vectors $u1, \ldots, ud$ are eigenvectors of A^d.

Periodic regimes in the projective space

Here are examples of matrices with periodic regimes.

$$F = \begin{pmatrix} . & . & \mathbb{1} \\ \mathbb{1} & . & . \\ . & \mathbb{1} & . \end{pmatrix}, \ (.) = -1 \ .$$

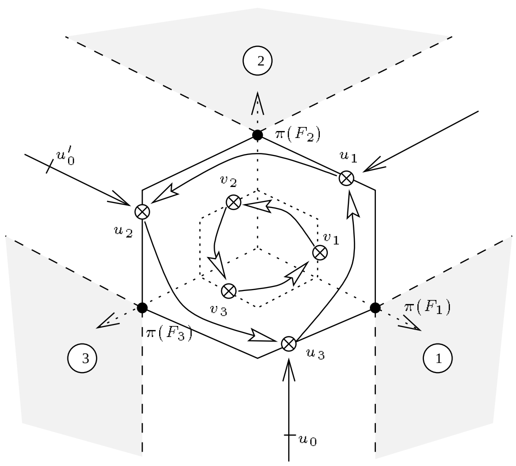


Figure 8: Set of periodic regimes and eigenvectors of matrix F.

Matrix F is scs1-cyc3. Unique eigenvector: $(\mathbb{1}, \mathbb{1}, \mathbb{1})'$. Initial condition in one of the gray zones: the limit periodic regime is $\{\pi(F_1), \pi(F_2), \pi(F_3)\}$. Initial condition in one of the white strips: the limit regime consists of three points on the boundary of the hexagon. For example for an initial condition u_0 or u'_0, the limit regime is $\{u_1, u_2, u_3\}$. Initial condition u in the hexagon: the periodic regime is $\{u, Fu, F^2u\}$.

$$G = \begin{pmatrix} . & \mathbb{1} & . \\ \mathbb{1} & . & . \\ . & . & \mathbb{1} \end{pmatrix}, \ (.) = -1 \ .$$

Matrix G is scs2-cyc2. Set of eigenvectors: $[\pi(G_3), \pi(g)]$. Set of periodic regimes and eigenvectors: the hexagon $\{\pi(G_1), \pi(G_2), \pi(G_3)\}$.

Algorithms

We consider $A \in \mathbb{R}_{max}^{k\times k}$, a normalized matrix (see above). We propose the following procedure to compute all the eigenvectors *and* the periodic regimes of A. Corresponding complexity can be shown to be $O(k^3\sqrt{k \log k})$, see [20].

1. Compute all the eigenvectors of A.
2. Determine the spectral type of A.

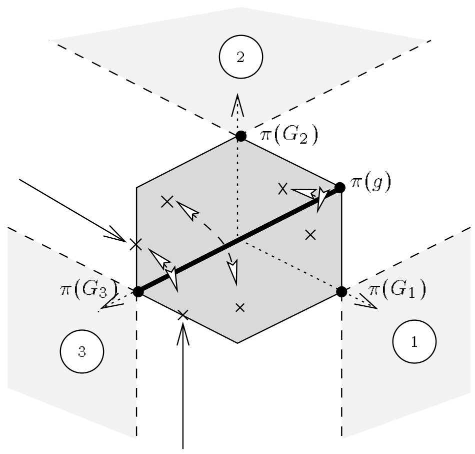


Figure 9: Set of periodic regimes and eigenvectors of matrix G

3. Compute all the periodic regimes of A.

Let us detail the different steps.

1. Compute all the eigenvectors of A.
Using Theorem 2, we have to compute matrix A^+.

2. Determine the spectral type of A.
We need to determine the critical graph of A. Let $u = (u_1, \ldots u_k)'$ be one of the eigenvectors of A. We compute

$$A' = P^{-1}AP, \text{ where } P_{ii} = u_i \text{ and } P_{ij} = \mathbb{0}\,. \tag{2}$$

All critical terms of A' are now equal to $\mathbb{1}$ and all non-critical terms are less than or equal to $\mathbb{1}$ (Lemma !). We consider the boolean "critical" matrix $\tilde{A}$ associated with A':

$$\begin{cases} \tilde{A}_{ij} = \mathbb{0} & \text{if } A'_{ij} < \mathbb{1} \\ \tilde{A}_{ij} = \mathbb{1} & \text{if } A'_{ij} = \mathbb{1}\,. \end{cases}$$

We compute the cyclicity of the matrix $\tilde{A}$ restricted to each s.c.s.
Let $l_1, \ldots, l_p$ be the cyclicities of the different s.c.s. The spectral type of matrix A is scs p-cyc l where $l = \operatorname{lcm}(l_1, \ldots, l_p)$.

3. Compute all the eigenelements of A.
We compute matrix $(A^l)^+$. The critical columns of $(A^l)^+$ form the periodic regimes of the matrix.

3 Timed Event Graphs are (max,+) Linear Systems

We restrict our attention to timed Event Graphs. In an Event Graph: no choice, only concurrency and synchronization, see Figure 1.

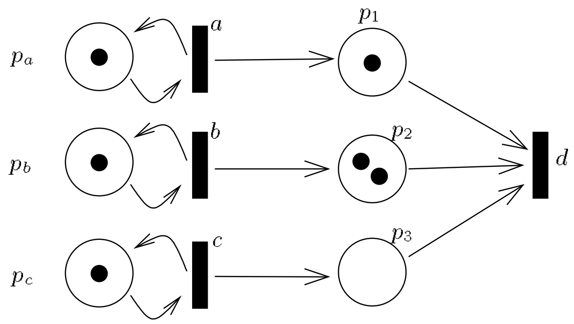


Figure 10: Timed Event Graph

$x(n) \in \mathbb{R}^{\mathcal{T}}$, $x(n)_a$ = instant of completion of the n-th firing at transition a.
The dynamic of the live Event Graph of Fig. 10 is given by

$$x(n)_d = \max\left[x(n-1)_a + \tau_{p_1}, x(n-2)_b + \tau_{p_2}, x(n)_c + \tau_{p_3}\right] + \tau_d \, . \tag{3}$$

Rewriting Equation (3) using the (max,+) notations:

$$x(n)_d = \bigoplus \left[x(n-1)_a \otimes \tau_{p_1}, x(n-2)_b \otimes \tau_{p_2}, x(n)_c \otimes \tau_{p_3}\right] \otimes \tau_d \, .$$

The whole vector $x(n)$ will satisfy an equation of the form:

$$x(n) = A_0 \otimes x(n) \oplus A_1 \otimes x(n-1) \oplus A_2 \otimes x(n-2) \, ,$$

with $(A_0)_{dc} = \tau_{p_3} \otimes \tau_d$, and $(A_1)_{da} = \tau_{p_1} \otimes \tau_d, (A_1)_{ii} = \tau_{p_i} \otimes \tau_i, i \in \{a, b, c\}$, and $(A_2)_{db} = \tau_{p_2} \otimes \tau_d$, all the other coefficients being equal to $\mathbb{0}$.

Theorem 4. *The dynamic of a* live *timed Event Graph* $\mathcal{G} = (\mathcal{T}, \mathcal{P}, \mathcal{F}, M, \tau)$ *can be represented as a (max,+) linear system:*

$$x(n) = A_0 x(n) \oplus A_1 x(n-1) \oplus \cdots \oplus A_m x(n-m) \, , \tag{4}$$

where $m = \max_{p \in \mathcal{P}} M(p)$*. We have* $(A_l)_{ij} = \tau_j + \max \tau_{p(i,j)}$ *where the* max *is taken over all places* $p(i,j)$ *such that* $(i, p(i,j)) \in \mathcal{F}, (p(i,j), j) \in \mathcal{F}$ *and* $M(p(i,j)) = l$*.*

For more details, see *Course Notes: Max Algebra Approach to Discrete Event Systems*, by G.J. Olsder in the present book, or *Stochastic (max,+) systems*, by F. Baccelli also in the present book.

Remark 1. A live Event Graph has the following property (`Exercise: prove it`): there exists $w \in L$ s.t. $|w|_a = 1, \forall a \in \mathcal{T}$ (i.e. each transition fires exactly once) and for any such w, $M \xrightarrow{w} M$. It explains why the vector $x(n)$ depends on the vector $x(n-i)$ in a way that does not depend on n.

Graph associated with a matrix

Definition 9. The graph of a matrix $A \in \mathbb{R}_{\max}^{k \times k}$ is a valued directed graph with k nodes. This graph contains an arc from i to j iff $A_{ji} \neq \mathbb{0}$. The valuation of this arc is A_{ji}.

Let us consider a matrix $A \in \mathbb{R}_{\max}^{k \times k}$ and its associated graph. We replace each node by a transition, each arc (i,j) by two arcs $(i, p(i,j))$ and $(p(i,j), j)$ where $p(i,j)$ is a newly created place with one token. The valuation of the arc (i,j) is transformed into two valuations: one associated with the transition j, τ_j, and one with the place $p(i,j)$, $\tau_{p(i,j)}$. We set:

$$\tau_j = \min_{l \,|\, A_{jl} \neq \mathbb{0}} A_{jl}, \;\; \tau_{p(i,j)} = A_{ji} - \tau_j \; .$$

We have obtained a timed Event Graph with exactly one token per place. If this Event Graph is live (`Exercise: what are the conditions on the original matrix` A `for the EG to be live?`), according to Theorem 4, we can associate with it a (max,+) linear system of order 1. It is easy to check that:

$$x(n+1) = Ax(n) \,,$$

where A is the matrix we started from. We obtain a simplified version of Eqn (4), as there is one token per place.

Dependence graph

In Fig. 11, we consider an Event Graph, and its representation as a Dependence Graph.

Definition 10. The timed Dependence Graph associated with a timed Event Graph $\mathcal{G} = (\mathcal{T}, \mathcal{P}, \mathcal{F}, M, \tau)$ is a valued infinite graph with $(\mathcal{T} \times \mathbb{Z})$ as the set of nodes. There is an arc from the node (i,n) to the node (j,m) if there exists $p \in \mathcal{P}$ such that $(i,p) \in \mathcal{F}$, $(p,j) \in \mathcal{F}$ and $M(p) = m - n$. The nodes (i,n) are valued by τ_i and the arcs $(i,n) \to (j,m)$ are valued by τ_p.

The Dependence Graph is 1-periodic, i.e. $(i,n) \to (j,m) \Leftrightarrow (i, n+1) \to (j, m+1)$ and the valuations are 1-periodic. It also verifies $(i,n) \to (j,m) \Rightarrow m \geq n$ (all arcs are pointing from left to right).
The other way round, let us consider a valued infinite graph with set of nodes $k \times \mathbb{Z}$, which is 1-periodic and such that $(i,n) \to (j,m) \Rightarrow m \geq n$. Then this infinite graph is the Dependence Graph of a timed Event Graph with k transitions.

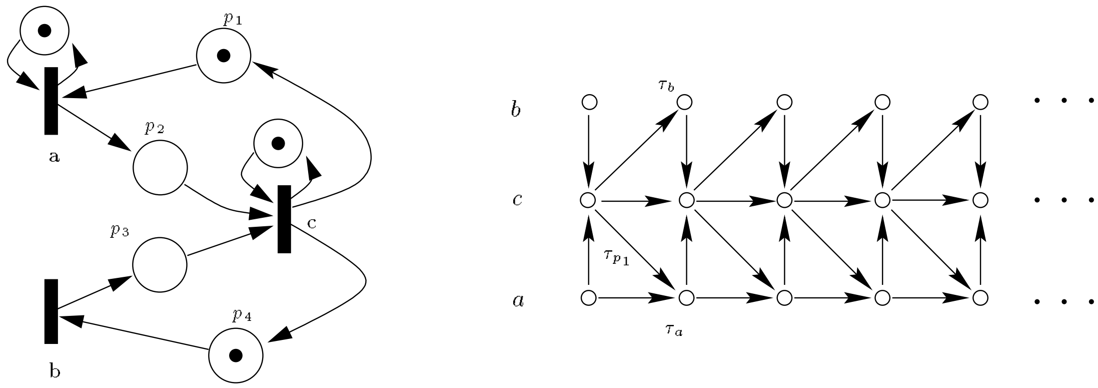


Figure 11: Event Graph and Dependence Graph

We have the following equivalences, between the three models introduced so far (timed Event Graph, (max,+) linear system and timed Dependence Graph):

(Max,+) linear system	*Timed Event Graph*	*Timed Dependence Graph*
$x(n) = A_0x(n) \oplus \cdots \oplus A_m x(n-m)$	Timed Event Graph with $\max_p M(p) = m$	1-periodic Timed Dependence Graph with arcs pointing from left to right
$x(n) = Ax(n-1)$	Timed Event Graph with $M(p) = 1, \forall p \in \mathcal{P}$	1-periodic Timed Dependence Graph with arcs between consecutive columns

Given a timed Event Graph, there exists various ways to obtain an equivalent timed Event Graph such that $M(p) = 1, \forall p \in \mathcal{P}$. We sketch below the simplest way to do so. For details, see [14] and [13].

Let us consider a live timed Event Graph. We split each place with several tokens as in Figure 12.
We obtain a live timed Event Graph with at most one token per place. We can associate with it a (max,+) linear system

$$x(n) = A_0x(n) \otimes A_1x(n-1)\ . \tag{5}$$

The maximal solution of this implicit equation is given by

$$x(n) = A_0^* A_1 x(n-1)\ , \tag{6}$$

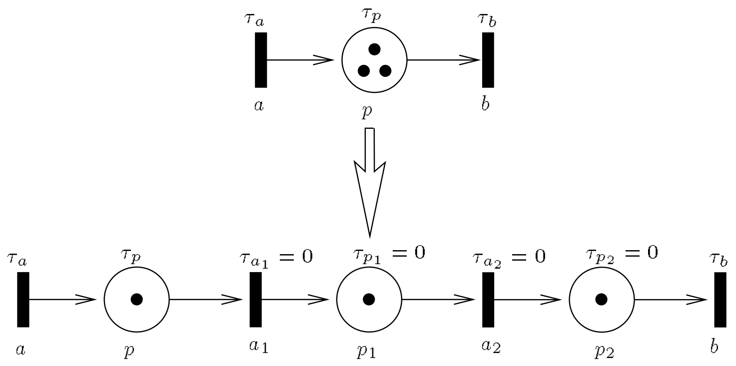


Figure 12: Splitting of a place with several tokens

where A_0^* is defined as in (1). `Exercise: Check that (6) is a solution of (5). Prove that it is the maximal one.`

Now we can associate with Eqn (6) a live timed Event Graph having exactly one token per place.

Given a live timed Event Graph with one token per place and its representation as a (max,+) linear system $x(n) = Ax(n-1)$, here are some

Generic problems that can be considered

I) asymptotic throughput: $\lim_n n/x(n)$;

II) stationary regime: $x(n+1) = x(n) + (\rho, \cdots, \rho)^t$;

III) meeting a deadline: $\max\{x(0) \mid x(n) \leq b\}$.

From a mathematical point of view, the solution to these problems is given by:

I) $\rho^{-1} \in \mathbb{R}$ where $Au = \rho u$ for some $u \in \mathbb{R}^k_{\max} \longrightarrow$ Spectral theory;

II) $u \in \mathbb{R}^k_{\max}$ where $Au = \rho u$ for some $\rho \in \mathbb{R} \longrightarrow$ Spectral theory;

III) $y = \max\{x \mid A^n x \leq b\} \longrightarrow$ Residuation theory.

The solutions are computed as follows. The spectral theory was discussed above, for details on residuation, see the contribution *Residuation and Applications* by G. Cohen in the present book.

I) $\rho = \max_{l=1}^k \max_{i1,\ldots,il}(A_{i1,i2} + \cdots + A_{il,i1})/l$;

II) $\forall i \in \mathrm{Crit}(A)$, $(A-\rho)^+_{.i}$;

III) $A^n \backslash b$, $(A^n \backslash b)_i = \max_{l=1}^k (b_l - A_{li})$;

3.1 A first (simple) manufacturing example

See Fig. 13. There are two types of items which have to be assembled together to form a part. There is an infinite stock for each kind of item. Each time a part is completed at the assembly line, a new request is sent to the storage facilities. New items are then sent to the assembly line. The work-in process of each item is 2 (i.e. at each instant at most two copies of each item can be in the system).

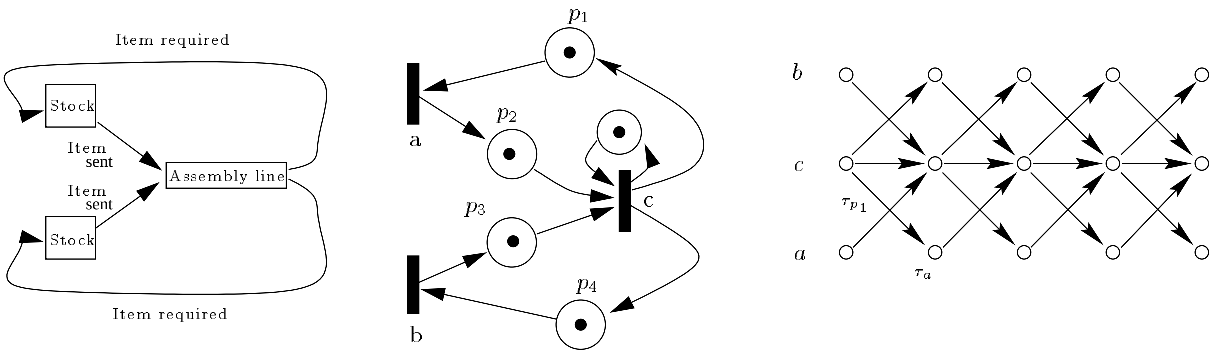


Figure 13: A manufacturing system, its associated Event Graph and Dependence Graph

- α : operating time at the assembly line.
- $\beta_i, i = 1, 2$: communication time between the assembly line and stock i.
- $\gamma_i, i = 1, 2$: transportation time between the stocks and the assembly line.

Corresponding (max,+) linear system:

$$x(n+1) = M \otimes x(n),\ M = \begin{pmatrix} \mathbb{0} & \mathbb{0} & \beta_1 \\ \mathbb{0} & \mathbb{0} & \beta_2 \\ \gamma_1 \otimes \alpha & \gamma_2 \otimes \alpha & \alpha \end{pmatrix} . \tag{7}$$

$\rho(M)$ = eigenvalue of M = mean cycle time = inverse of the throughput.

$$\rho(M) = \max(\alpha, (\beta_1 + \gamma_1 + \alpha)/2, (\beta_2 + \gamma_2 + \alpha)/2)$$

δ = idle time of the assembly line between the completion of a task and the beginning of the next one.

$$\delta(n) = x_3(n) - x_3(n-1) - \alpha . \tag{8}$$

Matrix M is either scs1-cyc2 or scs1-cyc1, depending on the values of α, β_i and γ_i.
Example: $\alpha = 2,\ \beta_1 = 2,\ \beta_2 = 1,\ \gamma_1 = 2$ and $\gamma_2 = 0$.

$$x(n+1) = M \otimes x(n), \ M = \begin{pmatrix} \mathbb{0} & \mathbb{0} & 2 \\ \mathbb{0} & \mathbb{0} & 1 \\ 4 & 2 & 2 \end{pmatrix} .$$

Set of periodic regimes: $\{\pi(u_\eta), \pi(v_\eta)\} = \{\pi(1 \otimes \eta, \mathbb{1}, 2), \pi(1, \mathbb{1}, 2 \otimes \eta)\}$, $\eta \in [\mathbb{1}, 1]$. Unique eigenvector: $\pi(u_0) = \pi(v_0) = \pi(1, \mathbb{1}, 2)'$. See Figure 14.

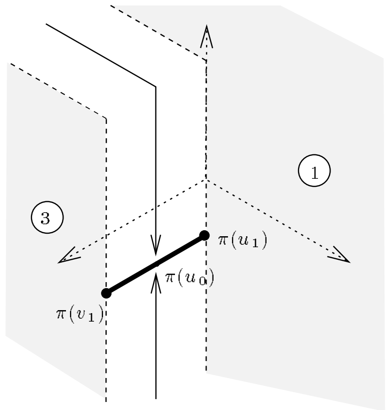


Figure 14: Set of periodic regimes of M

We consider a specific periodic regime $\{u_\eta, v_\eta\}$. We suppose that the manufacturing system is in this regime, i.e. $\pi(x(2n)) = \pi(u_\eta)$ and $\pi(x(2n+1)) = \pi(v_\eta)$.
We want to minimize the idle time and we want a control which consists in choosing only the initial condition.
We have $\delta(2n) = 1 + (u_\eta)_3 - (v_\eta)_3$ and $\delta(2n+1) = 1 + (v_\eta)_3 - (u_\eta)_3$. The maximal idle time is :

$$\begin{aligned} \overline{\delta} &= \max\left(\delta(2n), \delta(2n+1)\right) \\ &= \max\left(3 - \eta + (u_\eta)_3 - (v_\eta)_3, 3 - \eta + (v_\eta)_3 - (u_\eta)_3\right) = 3 + \eta \, . \end{aligned}$$

We minimize $\overline{\delta}$ for the eigenvector $\pi(u_0)$. We have to choose the initial condition in the domain of attraction of $\pi(u_0)$, i.e. on the broken line represented on Figure 14.

4 Timed 1-Bounded Petri Nets are (max,+) Automata

4.1 Timed 1-bounded Petri net

Let $\mathcal{G} = (\mathcal{T}, \mathcal{P}, \mathcal{F}, M, \tau)$ be a timed 1-bounded Petri net. We are going to associate with it 3 (max,+) Automata!

Marking Automaton

The *Marking Automaton* of $(\mathcal{G}, M)$ is an Automaton, see §1, with

- Alphabet $\mathcal{T}$; state space $\mathcal{R}(M)$; initial state M; transition function $\delta(M', a) = M''$ if $M' \xrightarrow{a} M''$.

Alternative description (as a (max,+) Automaton):

- Alphabet $\mathcal{T}$; state space $\mathcal{R}(M)$; initial vector I defined by $I_M = \mathbb{1}, I_m = \mathbb{0}, m \neq M$; morphism defined by: for $a \in \mathcal{T}$, $\mathcal{M}(a)_{M'M''} = \mathbb{1}$ if $M' \xrightarrow{a} M''$ and $\mathcal{M}(a)_{M'M''} = \mathbb{0}$ otherwise.

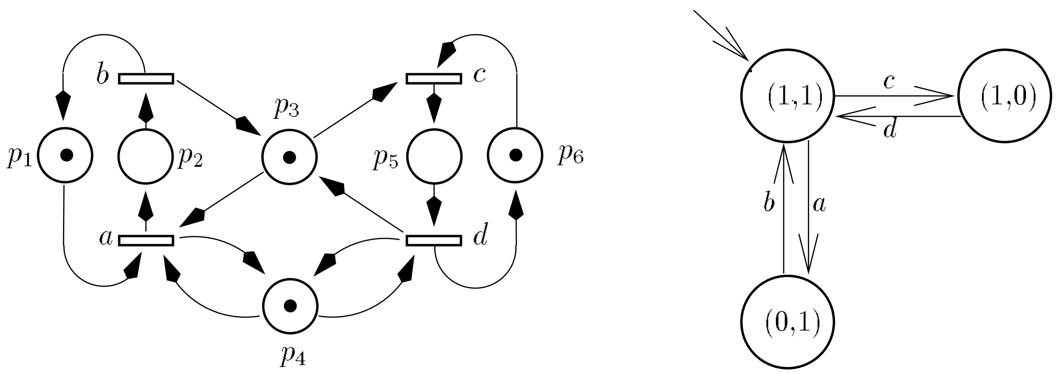


Figure 15: A 1-bounded Petri net and its Marking Automaton

This is the first (max,+) Automaton associated with the Petri net.

Trace Monoid

First studied by Cartier and Foata [4]. Introduced by Mazurkiewicz [22] for Petri nets as an alternative to the description of behaviors using L (subset of the free monoid).

Definition 11. Let $\mathcal{T}$ be an alphabet equipped with a reflexive symmetric relation D called *dependence relation*. We denote by I the complement of D, called *independence* relation. The *trace monoid* $(\mathcal{T}^*/_\sim)$ is the quotient of the free monoid $\mathcal{T}^*$ by the least congruence $\sim$ containing the relations $ab \sim ba$, $\forall (a, b) \in I$. The elements of $(\mathcal{T}^*/_\sim)$ will be called *traces*.

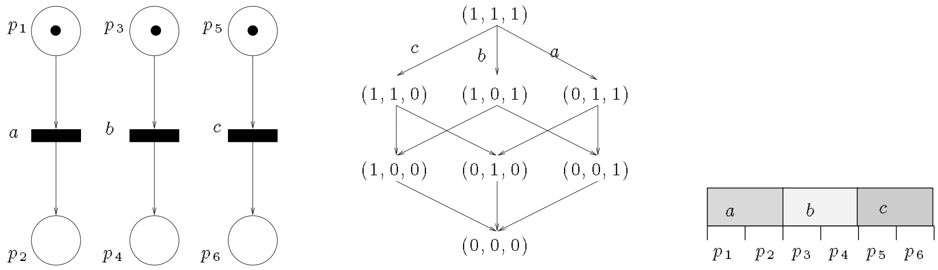


Figure 16: A Petri net, its Marking Automaton and ...

Two words are representatives of the same trace if they can be obtained one from the other by repeatedly interchanging adjacent independent letters.

Given a Petri net $(\mathcal{T}, \mathcal{P}, \mathcal{F}, M)$, we define the following independence relation:

$$I = \{(a, b) \in \mathcal{T}^2 \mid ({}^\bullet a \cup a^\bullet) \cap ({}^\bullet b \cup b^\bullet) = \emptyset\} \ , \tag{9}$$

and the associated trace monoid $(\mathcal{T}^*/_\sim)$. Two transitions are independent iff they do not share any input or output place.
In the example of Fig. 16, the same event (the concurrent firing of the transitions a, b and c) is described by $\{abc, acb, bac, bca, cab, cba\}$ in the free monoid and by the unique trace abc in the trace monoid.

Heap Monoids

Trace monoids can be visualized using heap monoids, as originally proposed by Viennot in [25]. Let $(\mathcal{T}^*/_\sim)$ be a trace monoid. Imagine an horizontal axis with a finite number of slots. A *piece* is a solid "block" occupying some of the slots and of height 1. With each letter is associated a piece. 2 pieces associated with dependent letters share some slots, and 2 pieces associated with independent letters do not. With a sequence of pieces, we associate a *heap* by piling up the pieces. This piling occurs as in the Tetris game: a piece is only subject to vertical translations and occupies the lowest possible position, above the pieces previously piled up.
See Fig. 16 and 17 for the trace monoid $(\mathcal{T}^*/_\sim)$ associated with a 1-bounded Petri net. The Petri net of Fig. 17 is the same as the one of Two words $w_1, w_2 \in \mathcal{T}^*$ are equivalent ($w_1 \sim w_2$) if and only if they provide the same heap. In the heap associated with $w \in L$, the pieces associated with one level (i.e. the pieces occupying a common vertical position) correspond to the events (i.e. the transition firings) occurring concurrently. The height of the heap w corresponds to the length of the longest chain of dependent firings in w (the *'sequentiality'* of the firing sequence).

Heap Automaton

For a heap monoid with slots Q and for a word w, we define the vector $x(w) \in \mathbb{R}^Q$ where $x(w)_i$ is the height of the heap on slot i after piling of the pieces of w. Next Theorem was

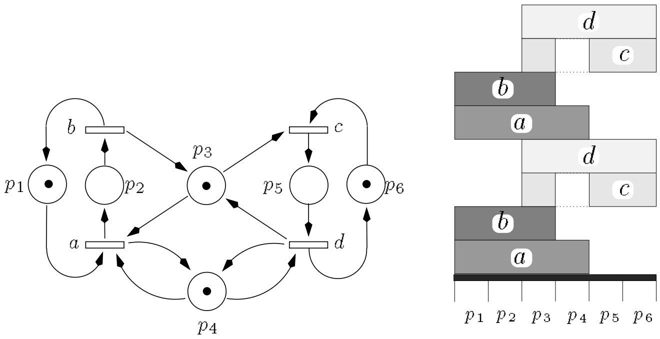


Figure 17: A 1-bounded Petri net and the execution $w = abcdabcd$ in the associated heap monoid

proved under various forms: Vincent [26], Brilman-Vincent[3], Gaubert-Mairesse [11]. We illustrate it below in the case of general pieces (see **Heap Automaton-II**).

Theorem 5. *The map $x : \mathcal{T}^* \to \mathbb{R}_{\max}^Q$ is (max,+) recognizable (see Def. 6). The corresponding (max,+) Automaton is $(\mathcal{T}, Q, \mathbb{1}_Q, \mathcal{M})$ where $\mathcal{M}$ is defined by*

$$\mathcal{M}(a)_{ij} = \begin{cases} 1 & \text{if piece } a \text{ occupies the slots } i \text{ and } j, \\ \mathbb{0} & \text{otherwise.} \end{cases}$$

In other words, we have $x(x) = \mathbb{1}_Q \mathcal{M}(a_1) \cdots \mathcal{M}(a_n)$ for $w = a_1 \cdots a_n$. This Automaton is called a Heap Automaton*.*

This is the second (max,+) Automaton associated with the Petri net.

Heap Automaton-II

We consider a heap model as above, except that the pieces associated with the letters can be pieces of general shape, see Fig. 18. Theorem 5 can easily be adapted to this case, as illustrated below on an example.

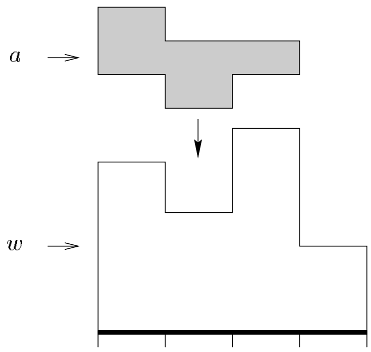


Figure 18: Heap of pieces of general shape.

For the example of Figure 18, the map $x(.)$ evolves as follows:

$$
\begin{aligned}
x_1(wa) &= \max[x_1(w), x_2(w)+1, x_3(w)] + 2 \\
&= \bigoplus [x_1(w) \otimes 2, x_2(w) \otimes 3, x_3(w) \otimes 2] \\
x(wa) &= x(w) \otimes \mathcal{M}(a) \, ,
\end{aligned}
$$

where

$$
\mathcal{M}(a) = \begin{pmatrix} 2 & 1 & 1 & \mathbb{0} \\ 3 & 2 & 2 & \mathbb{0} \\ 2 & 1 & 1 & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{1} \end{pmatrix} .
$$

The corresponding Automaton is still called a Heap Automaton.

Heap Automaton-III

- $a \in \mathcal{T}$, τ_a = firing time of a, $p \in \mathcal{P}$, τ_p = holding time of p.

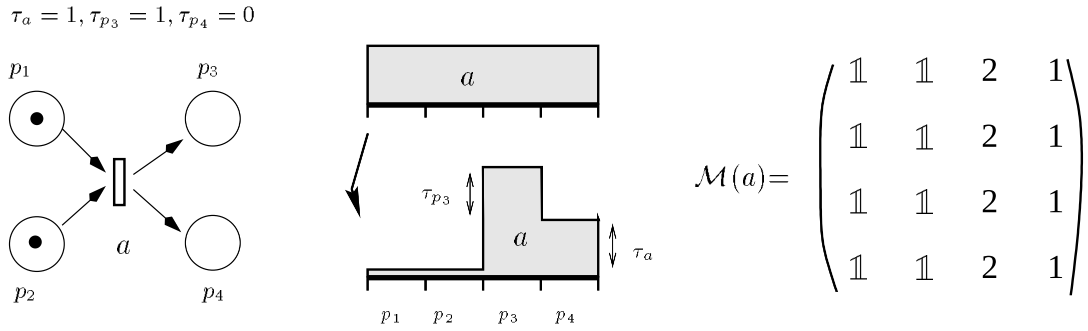


Figure 19: Transition in a timed Petri net, associated piece, associated (max) matrix.

The essential idea is that considering general pieces in a Heap Automaton enables to model time through the height of a piece. See Fig. 19, here the firing time of the transition is 1 and the holding times of p_3 and p_4 are 1 and 0 resp.

Theorem 6 (Gaubert-Mairesse [12]). *Let $(\mathcal{T}, \mathcal{P}, \mathcal{F}, M, \tau)$ be a timed 1-bounded Petri net with language L. For a word $w \in L$, let $y(w)$ be the last instant of activity in the Petri net under the firing sequence w. The map y is (max,+) recognizable; the Automaton being $(\mathcal{T}, \mathcal{P}, \mathbb{1}_{\mathcal{P}}, \mathcal{M})$ and the final vector being $\mathbb{1}_{\mathcal{P}}$. The morphism $\mathcal{M}$ is defined by*

$$
\mathcal{M}(a)_{ij} = \begin{cases} \mathbb{1} & \text{if } i,j \in {}^\bullet a \backslash a^\bullet, \\ \tau_a \otimes \tau_j & \text{if } i \in {}^\bullet a \cup a^\bullet, j \in a^\bullet, \\ \mathbb{0} & \text{otherwise.} \end{cases}
$$

In other words, we have $y(w) = \mathbb{1}_{\mathcal{P}} \mathcal{M}(a_1) \cdots \mathcal{M}(a_n) \mathbb{1}_{\mathcal{P}}$ for $w = a_1 \cdots a_n \in L$.

This is the third (max,+) Automaton associated with the Petri net.

Minimal Realization

Problem: Given a 1-bounded timed Petri net, and y the 'instant of last activity' map, find a representation $y(w) = I \otimes \mathcal{M}(w) \otimes F$, the matrices $\mathcal{M}(a)$ being of minimal dimension. In Th. 6, the dimension was the number of places of the net. In some cases, it can be reduced.

Definition 12. A *state machine component* (SMC) of a Petri net $\mathcal{G}$ is a subnet $\mathcal{G}'$ of $\mathcal{G}$, that is a state machine, and satisfies

$$^\bullet p \cup p^\bullet \subset \mathcal{G}' \text{ for every place } p \text{ of } \mathcal{G}'.$$

Let C be the subclass of 1-bounded Petri nets admitting a covering by 1-bounded SMC. C contains all 1-bounded Free Choice Nets (Theorem).

Theorem 7. *Let us consider a timed Petri net belonging to C. For $w \in L$, let $y(w)$ be the last instant of activity in the net for the sequence w. The map y is recognized by a Heap Automaton of dimension the number of SMC in the covering.*

We consider the 1-bounded Petri net of Fig. 15,17 with the following timings:

a	b	c	d	p_1	p_2	p_3	p_4	p_5	p_6
1	2	2	1	1	0	0	2	0	0

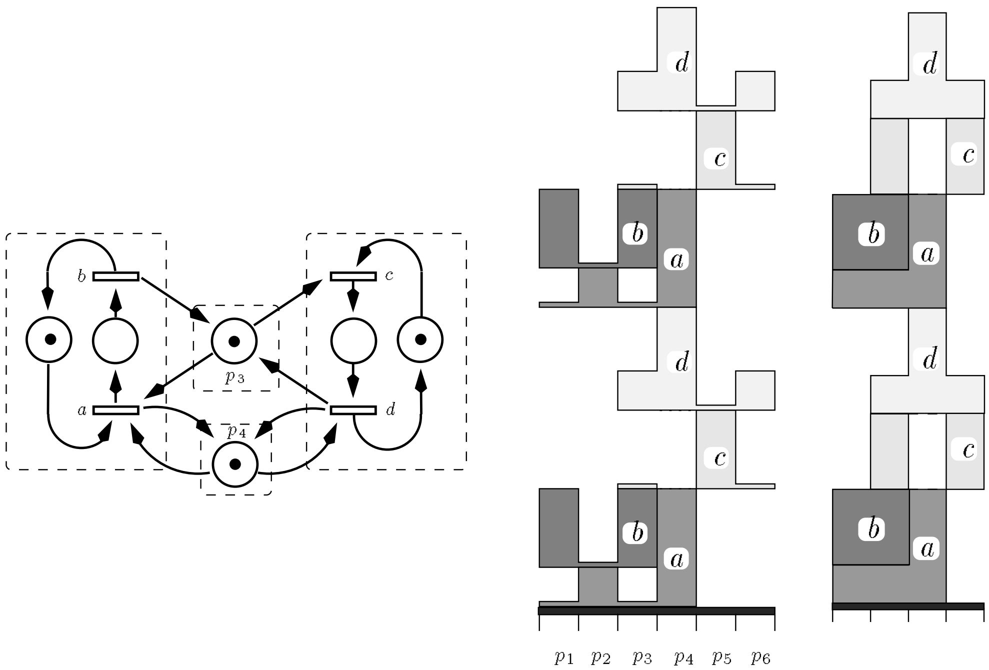


Figure 20: Heap Automata associated with a 1-bounded timed Petri net

We have represented in Fig. 20, the heap and the reduced heap associated with the execution $w = abcdabcd$ (to be compared with the same, but untimed, execution shown in Fig. 17).

Summary

Let $\mathcal{G}$ be 1-bounded Petri net, τ its timing. We associate with it 3 (max,+) Automata, recalled below with their utilization.

1. Marking Automaton, associated with $\mathcal{G} \to$ description of L;
2. Untimed Heap Automaton, associated with $\mathcal{G} \to$ sequentiality of w;
3. Timed Heap Automaton, associated with $(\mathcal{G}, \tau) \to$ execution time of w.

The 2 Heap Automata do not distinguish between legal executions (the ones in L) and non-legal ones. Hence it is necessary to use them together with the Marking Automaton.

In Figures 15, 17 and 20, we consider the same 1-bounded Petri net and the corresponding three automata.

4.2 Timed Automaton

By Definition, an Automaton is a SM with one token, hence it is 1-bounded! Following Theorem 7, it can be represented by a Heap Automaton of dimension 1. This can be translated in the following elementary way.

For $w = a_1 a_2 \cdots a_n \in L$, the last instant of completion of an event in the Automaton is given by:

$$y(w) = \sum_{i=1}^{n} m(a_i) = \sum_{i=1}^{n} \tau_{a_i} + \tau_{a_i^\bullet} \ .$$

4.3 A second manufacturing example

Heap representations have a natural domain of application in the scheduling of Jobshops. They can be used to design efficient performance evaluation methods.
We consider the Petri net of Fig. 15,17,20. This Petri net can be interpreted as a manufacturing system processing two job types $J_1 = ab$, $J_2 = cd$, using two machines, represented by places p_3 and p_4.

Given an infinite schedule $z = a_1 a_2 a_3 \ldots$, we define the *asymptotic throughput* of the jobs of type J_i (for a word w, $|w|_{J_i}$ is the number of type J_i jobs completed) under the schedule z:

$$\lambda_i = \liminf_{n\to\infty} \frac{|a_1 \ldots a_n|_{J_i}}{\text{execution time of } a_1 \ldots a_n} \ . \tag{10}$$

We are interested in finding the infinite schedules maximizing the throughput, under a production ratio constraint 1/1 (one job J_1 for one job J_2, in the average).

We require the schedules to be periodic sequences of the form $v^\omega = vvvv\dots$, where v is a finite *pattern* satisfying the ratio constraint.

Problem 1: How to evaluate $\lambda(v)$, the throughput associated with a given pattern.
Problem 2: How to choose the best schedule among patterns of a given length, i.e.

$$\max\{\lambda(v) \mid |v| = n\}$$

Problem 1

Event Graph Algorithm (Cohen-Dubois-Quadrat-Viot [6], Hillion-Proth [18])

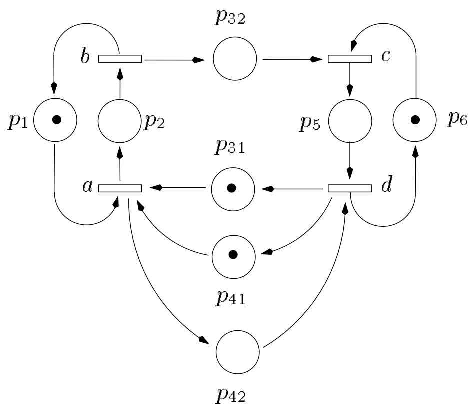


Figure 21: Timed Event Graph associated with the schedule $(abcd)^\omega$.

For a given periodic schedule, one is able to build a timed Event Graph representing the system. This is done by replacing some of the choice places by *circuits*, forcing the periodic behavior.
Let us consider for example the schedule $(abcd)^\omega$. This functioning is represented by the timed Event Graph displayed in Fig. 21, which is obtained from the timed Petri net of Fig. 15 by replacing the resource places p_3 and p_4 by *circuits*, forcing the sequence $abcdabcd\dots$

To compute the throughput of this timed Event Graph $\rightarrow$ back to the section on timed Event Graphs!
n : length of pattern, k : number of jobs + machines.
Complexity $O(nk + k^3)$ (details in the article given in Appendix).

Automata-based Algorithm

We use the following consequence of Theorem 6:

Proposition 1. *For a pattern v, the asymptotic throughput of job J is given by*

$$\lambda(v)_J = |v|_J \times \rho(\mathcal{M}(v))^{-1} \ , \tag{11}$$

where $\rho(\mathcal{M}(v))$ is the (max,+) eigenvalue of $\mathcal{M}(v)$.

Example: Scheduling $v^\omega = (abcdabcd)^\omega \longrightarrow \lambda(v)_{J_1} = 2\rho(\mathcal{M}(abcdabcd))^{-1}$.

Complexity $O(nk + k^3)$ (details in the article given in Appendix).

Complexities of both algorithms are the same. However, the "modeling" complexity is a strong argument in favor of the Automata-based algorithm. A new timed Event Graph must be built for each new schedule in the EG method, whereas the Heap Automaton $\mathcal{M}$ is built only once, and remains valid for all (even non-periodic!) schedules. Moreover, the size of the timed EG grows with the length of the pattern, whereas the size of the Heap Automaton remains constant.

Problem 2

It is known that finding the optimal periodic schedule (minimizing the throughput) v^ω is NP-complete as a function of $|v|$. Many classical heuristics: branch and bound, Lagrangian relaxation, stochastic algorithms, genetic algorithms

The Automata based approach can provide with a way to obtain local optimization starting from the schedule given by one of the classical methods.

$$\mathcal{M}(a_1)\mathcal{M}(a_2)\cdots\mathcal{M}(a_k)\mathcal{M}(a_{k+1})\cdots\mathcal{M}(a_n)$$

Local optimization. For example: exchanging $\mathcal{M}(a_k)$ and $\mathcal{M}(a_{k+1})$. By keeping in memory the products $\mathcal{M}(a_1)\mathcal{M}(a_2)\cdots\mathcal{M}(a_{k-1})$ and $\mathcal{M}(a_{k+2})\cdots\mathcal{M}(a_n)$, this requires only 3 matrix products.

Conclusion

Comments on the bibliography below. General references on Petri nets: [9, 23]; on automata with multiplicities [2, 10]; on the (max,+) semiring: [1, 8, 17, 21]; on algorithms in idempotent semirings [16]. The other references correspond to results quoted in the text.

References

[1] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity.* John Wiley & Sons, New York, 1992.

[2] J. Berstel and C. Reutenauer. *Rational Series and their Languages.* Springer Verlag, 1988.

[3] M. Brilman and J.M. Vincent. Dynamics of synchronized parallel systems. *Stochastic Models*, 13(3):605–619, 1997.

[4] P. Cartier and D. Foata. *Problèmes combinatoires de commutation et réarrangements.* Number 85 in Lecture Notes in Mathematics. Springer Verlag, 1969.

[5] G. Cohen, D. Dubois, J.P. Quadrat, and M. Viot. Analyse du comportement périodique des systèmes de production par la théorie des dioïdes. Technical Report 191, INRIA, 1983.

[6] G. Cohen, D. Dubois, J.P. Quadrat, and M. Viot. A linear system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing. *IEEE Trans. Automatic Control*, AC-30:210–220, 1985.

[7] R. Cuninghame-Green. Describing industrial processes with interference and approximating their steady-state behaviour. *Oper. Res. Quat.*, 13(1):95–100, 1962.

[8] R. Cuninghame-Green. *Minimax Algebra*, volume 166 of *Lecture Notes in Economics and Mathematical Systems.* Springer-Verlag, Berlin, 1979.

[9] J. Desel and J. Esparza. *Free Choice Petri Nets*, volume 40 of *Cambridge Tracts in Theoretical Comp. Sc.* Cambridge Univ. Press, 1995.

[10] S. Eilenberg. *Automata, languages and machines*, volume A. Academic Press, New York, 1974.

[11] S. Gaubert and J. Mairesse. Task resource models and (max,+) automata. In J. Gunawardena, editor, *Idempotency*, volume 11, pages 133–144. Cambridge University Press, 1997.

[12] S. Gaubert and J. Mairesse. Modeling and analysis of timed Petri nets using heaps of pieces. *IEEE Trans. Aut. Cont.*, 1998. To appear.

[13] B. Gaujal, A. Jean-Marie, and J. Mairesse. Minimal representation of uniform recurrence equations. Technical Report RR-2568, INRIA, Sophia-Antipolis, France, 1995. Submitted to SIAM J. Computing.

[14] B. Gaujal and A. Jean Marie. Computational issues in stochastic recursive systems. In J. Gunawardena, editor, *Idempotency.* Cambridge University Press, 1997.

[15] M. Gondran and M. Minoux. Valeurs propres et vecteurs propres dans les dioïdes et leur interprétation en théorie des graphes. *EDF, Bulletin de la Direction des Etudes et Recherches, Serie C, Mathématiques Informatique*, 2:25–41, 1977.

[16] M. Gondran and M. Minoux. *Graphes et algorithmes.* Eyrolles, Paris, 1979. Engl. transl. *Graphs and Algorithms*, Wiley, 1986.

[17] J. Gunawardena, editor. *Idempotency.* Publications of the Newton Institute. Cambridge University Press, 1998.

[18] H. Hillion and J.M. Proth. Performance evaluation of job shop systems using timed event graphs. *IEEE Trans. Automatic Control*, 34(1):3–9, 1989.

[19] J. Mairesse. A graphical approach of the spectral theory in the (max,+) algebra. *IEEE Trans. Automatic Control*, 40(10):1783–1789, October 1995.

[20] J. Mairesse. *Stabilité des systèmes à événements discrets stochastiques. Approche algébrique.* PhD thesis, Ecole Polytechnique, Paris, 1995. In english.

[21] V. Maslov and S. Samborskiĭ, editors. *Idempotent analysis*, volume 13 of *Adv. in Sov. Math.* AMS, 1992.

[22] A. Mazurkiewicz. Trace theory. In *Petri Nets, Applications and Relationship to other Models of Concurrency*, number 255 in LNCS, pages 279–324. Springer Verlag, 1987.

[23] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[24] I.V. Romanovskiĭ. Optimization and stationary control of discrete deterministic process in dynamic programming. *Cybernetics*, 3:66–78, 1967.

[25] G.X. Viennot. Heaps of pieces, I: Basic definitions and combinatorial lemmas. In Labelle and Leroux, editors, *Combinatoire Énumérative*, number 1234 in Lect. Notes in Math., pages 321–350. Springer, 1986.

[26] J.M. Vincent. Some ergodic results on stochastic iterative discrete events systems. *Discrete Event Dynamic Systems*, 7(2):209–233, 1997.