Asymptotic Analysis of Heaps of Pieces and application to Timed Petri Nets*

Stéphane Gaubert
INRIA
Domaine de Voluceau, B.P. 105
78153 Le Chesnay Cedex, France

Stephane.Gaubert@inria.fr

Jean Mairesse
LIAFA
CNRS-Université Paris 7
Case 7014, 2 place Jussieu,
75251 Paris Cedex 05, France
mairesse@liafa.jussieu.fr

Abstract

What is the density of an infinite heap of pieces, if we let pieces fall down randomly, or if we select pieces to maximize the density? How many transitions of a safe timed Petri net can we fire per time unit? We reduce these questions to the computation of the average and optimal case Lyapunov exponents of max-plus automata, and we present several techniques to compute these exponents. First, we introduce a completed "non-linear automaton", which essentially fills incrementally all the gaps that can be filled in a heap without changing its asymptotic height. Using this construction, when the pieces have integer valued shapes, and when any two pieces overlap, the Lyapunov exponents can be explicitly computed. We present two other constructions (partly based on Cartier-Foata normal forms of traces) which allow us to compute the optimal case Lyapunov exponent, assuming only that the pieces have integer valued shapes.

1 Introduction

Heap models, where solid blocks are piled up according to the Tetris game mechanism, are a good model of Discrete Event Dynamic Systems. They offer a trade-off between modeling power and tractability: on the one hand, timed 1-bounded Petri nets can be represented by heap models [20]; on the other hand, the height of a heap can be computed by a max-plus automaton [5, 19], which can be analyzed via spectral theory techniques [1, 18, 19].

In this paper, we address the following questions. Given a finite set of pieces, what is the density of the infinite heap obtained by letting fall down a sequence of pieces taken from this set, either randomly, according to a given probability measure; or in order to maximize the density, according to some given logical constraints? When rephrased in terms of timed Petri nets, these two questions become: how many transitions will be fired per time unit when conflicts are resolved in a random way; or if we select the firing sequence precisely to maximize the associated firing rate?

In algebraic terms, we represent pieces and transitions by letters: the height of a heap of pieces, or the execution time of a firing sequence, is computed by a max-plus automaton. The above questions are equivalent to the computation of Lyapunov exponents, which measure the growth rate of long products of matrices arising from the linear representation of the automaton. Our approach is a variation on Furstenberg's cocycle technique (see e.g. [4]), which can be translated in automata theoretical terms as follows. A max-plus automaton is *determinizable* if there exists a deterministic max-plus automaton with the same output. When it is effectively determinizable, the associated Lyapunov exponents can be computed. However, as far as we know, it remains an open problem to decide whether an integer-valued max-plus automaton is determinizable. Here, we use some specific features of heap automata to determinize them, in some cases, and we introduce a weaker form of determinization, which is enough to compute optimal case Lyapunov exponents. Due to space restrictions, most proofs are omitted.

2 Heaps of Pieces, Timed Petri Nets, and Max-Plus Automata

In this section, we briefly recall the results of [19, 20].

2.1 Heaps of Pieces

Consider a finite set $\mathcal{R}$ of columns and a finite set $\mathcal{A}$ of *pieces*. A piece $a \in \mathcal{A}$ is a rigid polyomino-shaped, possibly non connected, "block" occupying the

*This work was partially supported by the European Community Framework IV programme through the research network ALAPEDES ("The ALgebraic Approach to Performance Evaluation of Discrete Event Systems")

subset of columns $R(a)$. It has a lower and an upper contour which are represented by two row vectors $l(a)$ and $u(a)$ in $(\mathbb{R} \cup \{-\infty\})^{\mathcal{R}}$, respectively, with the conventions $l(a)_r = u(a)_r = -\infty$ if $r \notin R(a)$ and $\min_{r \in R(a)} l(a)_r = 0$. The upper contour $u(a)$ must be above the lower contour $l(a)$, i.e. $u(a) \geq l(a)$, where $\geq$ denotes the usual (entry-wise) order of $(\mathbb{R} \cup \{-\infty\})^{\mathcal{R}}$.

With an ordered sequence of pieces, denoted by a word[1] $w = a_1 \cdots a_k$, we associate a *heap* by piling up successively the pieces $a_1, \ldots, a_k$ on a ground whose shape is determined by a row vector $I \in \mathbb{R}^{\mathcal{R}}$. A piece is only subject to vertical translations and occupies the lowest possible position, provided it is above the ground and the pieces previously piled up.

The 6-tuple $\mathcal{H} = (\mathcal{A}, \mathcal{R}, R, u, l, I)$ constitutes a *heap model*. To avoid trivial cases, we assume that each piece occupies at least one column ($R(a) \neq \emptyset, \forall a \in \mathcal{A}$), and that each column is occupied by at least one piece ($R^{-1}(r) \neq \emptyset, \forall r \in \mathcal{R}$).

The *upper contour* of the heap w is represented by a row vector $x_{\mathcal{H}}(w)$ in $\mathbb{R}^{\mathcal{R}}$, where $x_{\mathcal{H}}(w)_r$ is the height of the heap on column r. The *height* of the heap w is $\mathcal{H}(w) = \max_{r \in \mathcal{R}} x_{\mathcal{H}}(w)_r$.

Example 2.1. Let us consider the heap model $\mathcal{H}_P$ defined as follows.

$\mathcal{A} = \{a, b, c\}$, $\mathcal{R} = \{1, 2, 3, 4\}$;
$R(a) = \{1, 2\}$, $R(b) = \{2, 3\}$, $R(c) = \{3, 4\}$;
$u(a) = [3, 1, -\infty, -\infty]$, $l(a) = [0, 0, -\infty, -\infty]$;
$u(b) = [-\infty, 1, 2, -\infty]$, $l(b) = [-\infty, 0, 0, -\infty]$;
$u(c) = [-\infty, -\infty, 2, 3]$, $l(c) = [-\infty, -\infty, 0, 0]$;
$I = [0, 0, 0, 0]$.

We have represented, in Figure 1, the heap associated with the word $w = bcab$.

We see that $x_{\mathcal{H}_P}(bcab) = [4, 5, 6, 5]$ and $\mathcal{H}_P(bcab) = 6$. Since the pieces a and c do not overlap, we have $x_{\mathcal{H}_P}(bacb) = x_{\mathcal{H}_P}(bcab)$ and, a fortiori, $\mathcal{H}_P(bacb) = \mathcal{H}_P(bcab)$.

2.2 Timed Petri Nets

We denote by $\mathcal{G} = (\mathcal{P}, \mathcal{T}, \mathcal{F}, M)$ a standard safe (i.e. 1-bounded) Petri net, where $\mathcal{P}$ is the set of places, $\mathcal{T}$ is the set of transitions, $\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ is the set of arcs, and $M : \mathcal{P} \to \mathbb{N}$ is the initial marking. We denote by $x^\bullet$ (resp. ${}^\bullet x$) the set of direct successors (resp. predecessors) of a node (place or transition) x. We call *language* of the Petri net $\mathcal{G}$ the set $L \subset \mathcal{T}^*$ of firing sequences starting from M. We call *timed Petri net* a Petri net with a constant *firing time* $\tau(a)$ associated with each transition $a \in \mathcal{T}$, and a constant *holding time* $\tau(p)$ associated with each place $p \in \mathcal{P}$. Transitions are fired according to the following rule, which has the same effect as the one of Ramchandani [28] on the subclass of safe nets. We assume that transition *a becomes* enabled at instant t. Then, the firing of a may be *initiated*. If initiated, the firing occurs in three steps.

1. At instant t, one token is removed from each input place;

2. at instant $t + \tau(a)$, one token is added in each output place;

3. after instant $t + \tau(a) + \tau(p)$, the token added in place $p \in a^\bullet$ can contribute to the enabling of the transitions in $p^\bullet$.

With this semantic, if a transition fires, it does so *as soon as possible*. Also, the decisions on which transitions are fired are not based on time considerations. All logically feasible choices can be considered.

With a firing sequence $w \in L$, sending the initial marking M to M', we associate the *makespan* or *execution time* $\mathcal{G}(w)$, which is the first instant at which all transitions of w have fired, and each token of the marking M' has completed the holding time in its current place.

We have shown in [20] that a heap model $\mathcal{H}$ is canonically associated to a safe timed Petri net $\mathcal{G}$, in such a way that for all $w \in L$, we have $\mathcal{G}(w) = \mathcal{H}(w)$. Thus, the height of the heap w is equal to the execution time of the firing sequence w when $w \in L$ (when $w \notin L$, $\mathcal{H}(w)$ has no Petri net interpretation). We do not recall this construc-

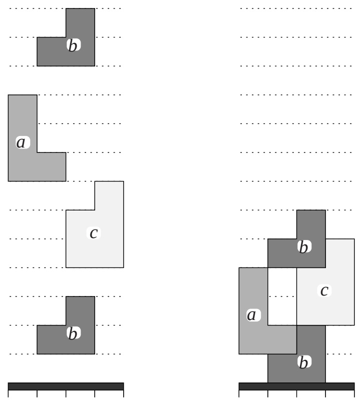

Figure 1. The heap model $\mathcal{H}_P$.

[1] Given a finite set (alphabet) $\mathcal{A}$, we denote by $\mathcal{A}^n$ the set of words of length n on $\mathcal{A}$. We denote by $\mathcal{A}^* = \cup_{n \in \mathbb{N}} \mathcal{A}^n$ the free monoid on $\mathcal{A}$, that is, the set of finite words equipped with concatenation. The unit (empty word) will be denoted by e. The length of the word w will be denoted by $|w|$. We shall write $|w|_a$ for the number of occurrences of a letter a in w. An *infinite word* (or *sequence*) is a mapping $u : \mathbb{N} \setminus \{0\} \to \mathcal{A}$. We write $u = u_1 u_2 \cdots$. The set of infinite words is denoted by $\mathcal{A}^\omega$. An infinite word u is *periodic* if there exists $l \in \mathbb{N} \setminus \{0\}$ such that $u_{i+l} = u_i, \forall i \in \mathbb{N} \setminus \{0\}$. In this case, we write $u = (u_1 \cdots u_l)^\omega$.

tion, but we next give two examples that we believe to be self-explanatory.

Example 2.2. Let us consider the safe Petri net $\mathcal{G}_P$ of Figure 2. The transitions a and c can be fired concurrently, while the transitions a and b (or, b and c) are in mutual exclusion. This net is a variant of the classical Dining Philosophers model [14] (the P in $\mathcal{G}_P$ stands for "Parallel" or "Philosopher"). The language of the Petri net is $\mathcal{T}^* = \{a, b, c\}^*$.

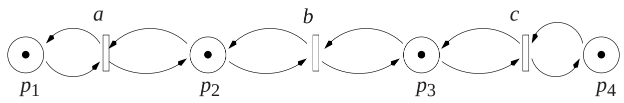

Figure 2. The Dining Philosophers Petri net $\mathcal{G}_P$.

The firing and holding times are

$$\tau(a) = \tau(b) = \tau(c) = 1,$$
$$\tau(p_1) = 2,\ \tau(p_2) = 0,\ \tau(p_3) = 1,\ \tau(p_4) = 2\,.$$

The heap model associated with $\mathcal{G}_P$ is $\mathcal{H}_P$, which was defined in Example 2.1. The pieces correspond to the transitions, the columns correspond to the tokens (equivalently to the places). A piece occupies the columns that correspond to the tokens involved in the firing of its associated transition. The height $u(t)_i - l(t)_i$ is equal to the sum of the firing time of transition t and of the holding time of place p_i. The execution time of the firing sequence $bcab$, $\mathcal{G}_P(bcab)$, coincides with the height of the heap of Figure 1, that is $\mathcal{H}_P(bcab) = 6$.

Example 2.3. This example is taken from [20]. We have represented in Figure 3 the safe timed Petri net $\mathcal{G}_S$ and its heap model $\mathcal{H}_S$. The firing and holding times are

$$\tau(a) = 1,\ \tau(b) = 2,\ \tau(c) = 2,\ \tau(d) = 1,$$
$$\tau(p_1) = 1,\ \tau(p_4) = 2,\ \tau(p_i) = 0,\ i = 2, 3, 5, 6.$$

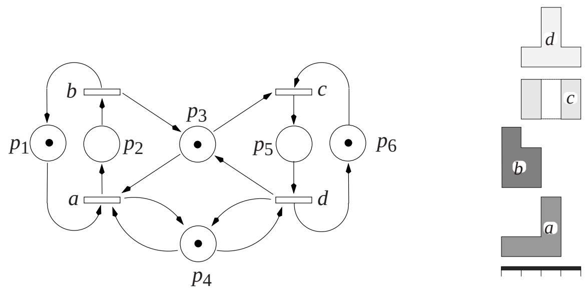

Figure 3. The Petri net $\mathcal{G}_S$ and the heap model $\mathcal{H}_S$.

There is no concurrency in this Petri net (the S in $\mathcal{G}_S$ stands for "Sequential"). The language of the Petri net is $L = \{ab, cd\}^*\{e, a, c\}$.

2.3 Automata with multiplicities over semirings

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 $\mathbb{0}$ ($a \oplus \mathbb{0} = a, a \otimes \mathbb{0} = \mathbb{0} \otimes a = \mathbb{0}$) and a unit element $\mathbb{1}$ ($a \otimes \mathbb{1} = \mathbb{1} \otimes a = a$).

For matrices A, B of appropriate sizes with entries in the semiring K, we set $(A \oplus B)_{ij} = A_{ij} \oplus B_{ij}$, $(A \otimes B)_{ij} = \bigoplus_k A_{ik} \otimes B_{kj}$, and for $a \in K$, $(a \otimes A)_{ij} = a \otimes A_{ij}$. We usually omit the $\otimes$ sign, writing for instance AB instead of $A \otimes B$. We denote by $\mathbb{0}$ (resp. $\mathbb{1}$) the matrix whose elements are all equal to $\mathbb{0}$ (resp. $\mathbb{1}$), the dimension depending on the context. Several semirings are used in this paper, hence the meaning of the symbols $\oplus, \otimes, \mathbb{0}$ and $\mathbb{1}$ depends on the context, but the symbols $+, -, \times, /, 0$ and 1 always have to be interpreted in the usual algebra. We will also use the "pseudo-norm" $|\cdot|_\oplus$, defined by $|A|_\oplus = \bigoplus_{ij} A_{ij}$.

The structure $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +)$ is a semiring, which is called the *max-plus semiring*. Here, $\oplus = \max, \otimes = +, \mathbb{0} = -\infty$ and $\mathbb{1} = 0$. This semiring is *idempotent*: $a \oplus a = a$, for all a. The min-plus semiring $\mathbb{R}_{\min}$ is obtained by replacing max by min and $-\infty$ by $+\infty$ in the definition of $\mathbb{R}_{\max}$. The structure $\mathbb{R}_+ = (\mathbb{R}_+, +, \times)$ is also a semiring. The Boolean semiring $\mathbb{B}$ can be identified with the subsemiring $(\{\mathbb{0}, \mathbb{1}\}, \oplus, \otimes) \subset \mathbb{R}_{\max}$.

An *automaton* of dimension k, with multiplicities in a semiring K, over an alphabet $\mathcal{A}$, is a triple $\mathcal{U} = (\alpha, \mu, \beta)$, where $\alpha \in K^{1\times k}$, $\beta \in K^{k\times 1}$, and where $\mu : \mathcal{A}^* \to K^{k\times k}$ is a morphism of monoids. The morphism μ is uniquely defined by the family of matrices $\{\mu(a)\}_{a\in\mathcal{A}}$, since $\mu(a_1 \dots a_n) = \mu(a_1)\cdots\mu(a_n)$ for all $a_1, \dots, a_n \in \mathcal{A}$. We say that the map $\mathcal{U} : \mathcal{A}^* \to K, \mathcal{U}(w) = \alpha\mu(w)\beta$, is *recognized* by the automaton.

Such an automaton is represented graphically by a labeled and weighted digraph with k nodes. There is an arc from node i to node j, with label a and weight $\mu(a)_{ij}$, if $\mu(a)_{ij} \neq \mathbb{0}$. There is an unlabeled input arc at node i with weight α_i if $\alpha_i \neq \mathbb{0}$, and there is an unlabeled output arc at node j with weight β_j if $\beta_j \neq \mathbb{0}$ (then, i and j are called input and output nodes).

An automaton (α, μ, β) is *deterministic* if there is a unique input node, and if for all i, and for all $a \in \mathcal{A}$, there is at most one j such that $\mu(a)_{ij} \neq \mathbb{0}$. When for all i and a, there is exactly one node j with this property, we say that the automaton is *complete*. If a deterministic automaton is not complete, we can make it complete without changing its output by introducing an extra node.

We call *max-plus automaton* an automaton with multiplicities over the max-plus semiring. Since $\mathbb{B}$ is a subsemiring of $\mathbb{R}_{\max}$, a classical automaton, that is an automaton with multiplicities over $\mathbb{B}$, is a special case of a max-plus automaton. In the graph associated with a Boolean automaton, we do not represent the weights, see Figure 9 for instance. For more details on automata with multiplicities, see [2, 16].

For each piece a of a heap model $\mathcal{H}$, we define the matrix $\mathcal{M}(a) \in \mathbb{R}_{\max}^{\mathcal{R}\times\mathcal{R}}$ by

$$\mathcal{M}(a)_{sr} = \begin{cases} \mathbb{1} & \text{if } s = r, r \notin R(a), \\ u(a)_r - l(a)_s & \text{if } r \in R(a), s \in R(a), \\ \mathbb{0} & \text{otherwise.} \end{cases} \tag{1}$$

Equivalently, we have

$$\mathcal{M}(a) = \mathrm{Id} \oplus \tilde{l}(a)u(a) \ , \tag{2}$$

where Id is the identity matrix ($\mathrm{Id}_{ii} = \mathbb{1}, \ \forall i; \mathrm{Id}_{ij} = \mathbb{0}, \ \forall i \neq j$), and $\tilde{l}(a)$ is the column vector defined by $\tilde{l}(a)_i = -l(a)_i$ if $l(a)_i \neq \mathbb{0}$ and $\tilde{l}(a)_i = l(a)_i = \mathbb{0}$ otherwise.

We extend $\mathcal{M} : \mathcal{A} \to \mathbb{R}_{\max}^{\mathcal{R}\times\mathcal{R}}$ to a morphism $\mathcal{M} : \mathcal{A}^* \to \mathbb{R}_{\max}^{\mathcal{R}\times\mathcal{R}}$, setting $\mathcal{M}(a_1 \ldots a_n) = \mathcal{M}(a_1)\cdots\mathcal{M}(a_n)$ (the products are of course in the max-plus semiring). The following theorem was proved by the authors in [19, 20] and, independently, with a different formulation, by Brilman and Vincent [5].

Theorem 2.4. *Let $\mathcal{H} = (\mathcal{A}, \mathcal{R}, R, u, l, I)$ be a heap model. The upper contour and the height of the heap $w \in \mathcal{A}^*$ are given respectively by*

$$\begin{aligned} x_{\mathcal{H}}(w) &= I\mathcal{M}(w) \ , \\ \mathcal{H}(w) &= I\mathcal{M}(w)\mathbb{1} \ . \end{aligned} \tag{3}$$

Thus, $\mathcal{H}$ is recognized by the max-plus automaton $(I, \mathcal{M}, \mathbb{1})$. It will be convenient to use the notation $\mathcal{H}$ for both the heap model and the associated max-plus automaton. We also call $\mathcal{H}$ a *heap automaton*.

Example 2.5. The heap automaton associated with the heap model $\mathcal{H}_P$ of Example 2.1 and 2.2 is $\mathcal{H}_P = (\mathbb{1}, \mathcal{M}, \mathbb{1})$, where the morphism $\mathcal{M}$ is defined by the matrices $\mathcal{M}(a)$, $\mathcal{M}(b)$ and $\mathcal{M}(c)$, given respectively by:

$$\begin{pmatrix} 3 & 1 & & \\ 3 & 1 & & \\ & & \mathbb{1} & \\ & & & \mathbb{1} \end{pmatrix}, \begin{pmatrix} \mathbb{1} & & & \\ & 1 & 2 & \\ & 1 & 2 & \\ & & & \mathbb{1} \end{pmatrix}, \begin{pmatrix} \mathbb{1} & & & \\ & \mathbb{1} & & \\ & & 2 & 3 \\ & & 2 & 3 \end{pmatrix}$$

(the $\mathbb{0}$ entries are omitted).

3 Worst, Optimal, and Average case Lyapunov Exponents

Following [22], we say that $p : \mathcal{A}^* \to [0, 1]$ is a *measure on words* if

$$p(e) = 1 \quad \text{and} \quad p(w) = \sum_{a\in\mathcal{A}} p(wa) \quad \text{for all } w \in \mathcal{A}^* \ .$$

Then, of course, $\sum_{a\in\mathcal{A}} p(a) = 1$. A measure on words is *rational* if the map p is recognized by an automaton with multiplicities in $\mathbb{R}_+$, over the alphabet $\mathcal{A}$. A rational measure is *Bernoulli* if it is recognized by an automaton of dimension 1. Then, $p(a_1 \ldots a_n) = p(a_1) \times \cdots \times p(a_n)$, for all $a_1, \ldots, a_n \in \mathcal{A}$.

Let $\mathcal{U} = (\alpha, \mu, \beta)$ be a max-plus automaton over the alphabet $\mathcal{A}$ and let p be a measure on words. We define the following performance measures:

$$\rho_{\max}(\mathcal{U}) = \limsup_{n\to+\infty} \frac{1}{n} \max_{w\in\mathcal{A}^n} \mathcal{U}(w) \ , \tag{4}$$

$$\rho_{\min}(\mathcal{U}) = \liminf_{n\to+\infty} \frac{1}{n} \min_{\substack{w\in\mathcal{A}^n \\ \mathcal{U}(w)\neq\mathbb{0}}} \mathcal{U}(w) \ , \tag{5}$$

$$\rho_E(\mathcal{U}) = \lim_{n\to+\infty} \frac{1}{n} \sum_{\substack{w\in\mathcal{A}^n \\ \mathcal{U}(w)\neq\mathbb{0}}} p(w) \times \mathcal{U}(w) \ . \tag{6}$$

The "E" in ρ_E stands for "expectation". The limit $\rho_E(\mathcal{U})$ need not exist, but Theorem 3.2 below gives a sufficient condition for its existence. We christen $\rho_{\max}(\mathcal{U})$ (resp. $\rho_{\min}(\mathcal{U})$, $\rho_E(\mathcal{U})$) *worst* (resp. *optimal*, *average*) *case Lyapunov exponent*.

Given a language $L \subset \mathcal{A}^*$, we define $\rho_{\max}(\mathcal{U}, L)$, $\rho_{\min}(\mathcal{U}, L)$ and $\rho_E(\mathcal{U}, L)$ by replacing $\mathcal{A}^n$ by $\mathcal{A}^n \cap L$ in the definitions (4),(5) and (6). These quantities are called *constrained Lyapunov exponents*.

For a heap automaton $\mathcal{H}$, the limits $\rho_{\max}(\mathcal{H})$, $\rho_{\min}(\mathcal{H})$ and $\rho_E(\mathcal{H})$ are equal respectively to the maximal, minimal and average asymptotic growth rates of an infinite heap of pieces. For a Petri net, the constrained Lyapunov exponents are particularly relevant, as performances have to be evaluated over the language of admissible firing sequences, which is recognizable.

Computing Lyapunov exponents constrained by a recognizable language is not theoretically harder than in the unconstrained case. Indeed, let $\mathcal{L}$ denote a Boolean automaton recognizing L and let $\mathcal{U} \odot \mathcal{L}$ denote the *tensor product*[2] of the two automata. Then, $\mathcal{U} \odot \mathcal{L}$ is a max-plus automaton and $\rho_{\max}(\mathcal{U}, \mathcal{L}) = \rho_{\max}(\mathcal{U} \odot \mathcal{L})$ (the corresponding identities hold for $\rho_{\min}$ and ρ_E).

We next recall when and how the three Lyapunov exponents can be computed. Recall that an automaton (α, μ, β) is *trim* if for all i, there exists $w, z \in \mathcal{A}^*$ such that $(\alpha\mu(w))_i \neq \mathbb{0}$ and $(\mu(z)\beta)_i \neq \mathbb{0}$, i.e. if each node is connected to an input node and to an output node.

The following result, which is taken from [18], is an immediate consequence of the max-plus spectral theorem

[2] In any semiring, the tensor product of a $r \times s$ matrix A by a $r' \times s'$ matrix A' is the $rr' \times ss'$ matrix $A \odot A'$ with entries $(A \odot A')_{(i,i')(j,j')} = A_{ij} A'_{i'j'}$. The tensor product of the automata (α, μ, β) and (α', μ', β') is $(\alpha \odot \alpha', \mu \odot \mu', \beta \odot \beta')$. We do not use the classical symbol $\otimes$ for the tensor product due to the conflict with the notation of the semiring product.

(see [1, 26] and the references therein).

Theorem 3.1. *If $\mathcal{U} = (\alpha, \mu, \beta)$ is a trim max-plus automaton, then $\rho_{\max}(\mathcal{U})$ coincides with the maximal max-plus eigenvalue of $M = \bigoplus_{a\in\mathcal{A}} \mu(a)$, which is given by*

$$\begin{aligned} \lambda &= \bigoplus_{1\leq p\leq k} \bigoplus_{i_1\dots i_p} (M_{i_1 i_2} \cdots M_{i_p i_1})^{1/p} \qquad (7) \\ &= \max_{1\leq p\leq k} \max_{i_1\dots i_p} \frac{M_{i_1 i_2} + \cdots + M_{i_p i_1}}{p} \ , \end{aligned}$$

where k is the size of M.

Let $a_{ij} \in \mathcal{A}$ be a letter which attains the maximum in $\bigoplus_{a\in\mathcal{A}} \mu(a)_{ij}$. Let $(i_1, \dots, i_l)$ be such that $(M_{i_1 i_2} + \cdots + M_{i_l i_1})/l = \lambda$ and let $w = a_{i_1 i_2} \cdots a_{i_l i_1}$. Then *worst case words* of increasing length can be obtained using the pattern w. More precisely, if u (resp. v) is the label of a path from an input node to i_1 (resp. from i_1 to an output node), we have $\lim_n \mathcal{U}(uw^n v)/|uw^n v| = \rho_{\max}(\mathcal{U})$.

Karp's algorithm [23, 1] computes λ in time $O(kE)$, where E is the number of edges of the graph of M. The max-plus version of Howard's policy improvement algorithm, presented in [10], is experimentally much faster, although no polynomial bound is known for its complexity.

There is no simple general formula for $\rho_{\min}$ or ρ_E, except in an important special case, that we next present.

Let $\mathcal{U} = (\alpha, \mu, \beta)$ be a deterministic and trim max-plus automaton. We denote by $\tilde{\mathcal{U}}$ the *min-plus* automaton defined by the same triple (α, μ, β) (but with $\mathbb{0} = +\infty$). As there is at most one path with label w in the graph of (α, μ, β), we obtain easily that $\tilde{\mathcal{U}}(w) = \mathcal{U}(w)$ if $\mathcal{U}(w) \neq -\infty$ (and $\tilde{\mathcal{U}}(w) = +\infty$ if $\mathcal{U}(w) = -\infty$). We deduce from Theorem 3.1 that $\rho_{\min}(\mathcal{U}) = \rho_{\min}(\tilde{\mathcal{U}})$ is equal to the minimal min-plus eigenvalue of the matrix $M = \bigoplus_{a\in\mathcal{A}} \mu(a)$ (with $\oplus = \min$), which is given by (7).

We now consider the average case Lyapunov exponent ρ_E. Let $\mathcal{U} = (\alpha, \mu, \beta)$ be a complete and deterministic max-plus automaton and let p be a Bernoulli measure. Given a node i and a letter a, we denote by $i \cdot a$ the unique node such that $\mu(a)_{i,i\cdot a} \neq \mathbb{0}$. We define the *reward vector* c and the *transition matrix* P by

$$c_i = \sum_{a\in\mathcal{A}} p(a) \times \mu(a)_{i,i\cdot a} \ , \quad P_{ij} = \sum_{a\in\mathcal{A}, i\cdot a = j} p(a) \ . \qquad (8)$$

The next result, which is detailed in [18], is an immediate consequence of the ergodic theorem for Markov chains.

Theorem 3.2. *If p is a Bernoulli probability measure and if $\mathcal{U} = (\alpha, \mu, \beta)$ is a complete and deterministic max-plus automaton, with input node i_0, and with $\mathcal{U}(w) \neq \mathbb{0}$, for all w, we have*

$$\rho_E(\mathcal{U}) = (Q \times c)_{i_0} \ , \qquad (9)$$

where $Q = \lim_n (P + \cdots + P^n)/n$ is the spectral projector of P for the eigenvalue 1 (all the operations are in $\mathbb{R}_+$).

In particular, if P is irreducible or has a unique final class, it has a unique invariant measure π, which is solution of $\pi \times P = \pi$, we have $Q = [1, \dots, 1]^T \times \pi$, and $\rho_E(\mathcal{U}) = \pi \times c$. These results can be adapted to the case of a rational measure p, thanks to the following tensor product construction. We take a representation (δ, ν, γ) of p, and we introduce the representation $\hat{\mathcal{U}} = (\hat\alpha, \hat\mu, \hat\beta)$ over the semiring $\mathbb{R}_+$ obtained from (α, μ, β) by replacing all finite entries by 1, and $-\infty$ by 0. We denote by $\odot$ the tensor product of matrices in the usual algebra, and we set:

$$\begin{aligned} P &= \sum_{a\in\mathcal{A}} \nu(a) \odot \hat\mu(a), \qquad (10) \\ c &= \sum_{a\in\mathcal{A}} (\nu(a) \times \gamma) \odot (\mu(a) \otimes \mathbb{1}) \qquad (11) \end{aligned}$$

(all the operations are in the usual algebra, except the $\otimes$ which is in the max-plus semiring, and $\mathbb{1}$ denotes the column vector with the same size as β whose entries are all equal to 0). If (δ, ν, γ) is trim, the matrix $M = \sum_{a\in\mathcal{A}} \nu(a)$ has Perron root 1, this root is semisimple, and we may assume that $M \times \gamma = \gamma$ (see [22]). Then, an easy adaptation of the proof of [18] shows that:

$$\rho_E(\mathcal{U}) = (\delta \odot \hat\alpha) \times Q \times c \ ,$$

where Q denotes the spectral projector of P for the eigenvalue 1.

The exponents $\rho_{\min}$ and ρ_E can be explicitly computed for deterministic automata. This raises the following problem: given a map $\mathcal{U}$ recognized by a non-deterministic max-plus automaton, can we decide if $\mathcal{U}$ is recognized by a deterministic automaton, and if it is the case, can we find it effectively? To the best of our knowledge, this problem remains open. See [9] (resp. [17]), for a noneffective characterization "à la Nerode" (resp. "à la Hankel") of deterministic series.

4 Completed Automaton

In this section, we give a determinization procedure for a subclass of heap automata.

4.1 Determinization via normalization

Let us consider a non-deterministic automaton $\mathcal{U} = (\alpha, \mu, \beta)$. The classical determinization algorithm of Boolean automata admits the following extension.

Define the "normalization" operator $\pi : \mathbb{R}^k_{\max} \setminus \{\mathbb{0}\} \to \mathbb{R}^k_{\max} \setminus \{\mathbb{0}\}$ by $\pi(x) = x - |x|_\oplus \mathbb{1}$, i.e. $\pi(x)_i = x_i - \max_j x_j$.

By construction, $\pi(x)$ has maximum 0. We extend the operator π to $\mathbb{R}_{\max}^k$ by setting $\pi(\mathbb{0}) = \mathbb{0}$. We define the set $\pi(\mathcal{U}) \stackrel{\text{def}}{=} \{\pi(\alpha\mu(w)) \mid w \in \mathcal{A}^*\}$. If $\pi(\mathcal{U})$ is finite, we introduce the *normalized automaton* (κ, ν, γ), with set of states $\pi(\mathcal{U})$, and

$$\kappa_u = \begin{cases} |\alpha|_\oplus & \text{if } u = \pi(\alpha) \\ \mathbb{0} & \text{otherwise,} \end{cases} \quad \gamma_u = u\beta, \quad \text{and,}$$

$$\nu(a)_{uv} = \begin{cases} |u\mu(a)|_\oplus & \text{if } \pi(u\mu(a)) = v \\ \mathbb{0} & \text{otherwise.} \end{cases} ,$$

By construction, (κ, μ, γ) is complete, deterministic and trim, and it is quite easy to see that it recognizes the same map as $\mathcal{U}$. The only difficulty is that $\pi(\mathcal{U})$ is not finite, in general (as opposed to the Boolean case). In a heap automaton, as soon as two pieces do not overlap (i.e. $\exists a, b$, $R(a) \cap R(b) = \emptyset$), the set $\pi(\mathcal{U})$ is infinite and this determinization procedure does not work.

For instance, for the heap model $\mathcal{H}_S$ depicted in Figure 3, we have $\pi(\mathbb{1}\mathcal{M}(a)^n) = (-2, -2, 0, -3 \times n)$, i.e., if we pile up n times the piece a, we obtain a hole of depth $3n$ on column 4 of the heap. This shows that $\pi(\mathcal{U}) = \{\pi(\mathbb{1}\mathcal{M}(w)) \mid w \in \mathcal{A}^*\}$ is infinite. However, the hole on column 4 has no effect on the dynamics of the heap, and a simple look at Figure 3 should convince the reader that $(-2, -2, 0, -3 \times n)\mu(w) = (-2, -2, 0, -2)\mu(w)$, for all $w \in \mathcal{A}^*$. By filling all the gaps in the vectors of $\pi(\mathcal{U})$, we obtain in this case a finite number of possible contours, hence a deterministic automaton with the same behavior as $\mathcal{H}_S$. We next formalize this idea, using residuation theory.

4.2 Completion of heaps, residuation, and non-linear automata

We first recall some elements of residuation [15, 3]. In a nutshell, residuation replaces the notion of inverse, which need not exist in some ordered algebraic structures, by that of maximal subinverse. We say that an ordered semiring K is residuated if for all a, b, c, $\{x \in K \mid ax \leq b\}$ has a greatest element, that we denote by $a \backslash b$, and if dually, $\{x \in K \mid xc \leq b\}$ has a greatest element, that we denote by b/c. Of course, if the product is commutative $a \backslash b = b/a$.

The semiring $\mathbb{R}_{\max}$, equipped with its usual order structure, is not residuated, but it can be embedded in the completed semiring $\overline{\mathbb{R}}_{\max} = (\mathbb{R} \cup \{-\infty, +\infty\}, \max, +)$, which is residuated. Since $\mathbb{0} = -\infty$ is absorbing, we have $(-\infty) + (+\infty) = -\infty$ in $\overline{\mathbb{R}}_{\max}$. With the convention $(+\infty) - (+\infty) = (-\infty) - (-\infty) = +\infty$, we have

$$x/y = x - y \ .$$

Matrices over $\overline{\mathbb{R}}_{\max}$ are equipped with the product ordering $D \leq B$ iff $D_{ij} \leq B_{ij}$, for all i, j. For matrices A, B, C of compatible sizes with entries in $\overline{\mathbb{R}}_{\max}$, we define by extension $A \backslash B = \max\{X \mid AX \leq B\}$ and $B/C = \max\{X \mid XC \leq B\}$. Then, it is not difficult to check that $A \backslash B$ and B/C are given by ($\wedge$ denotes the min operation):

$$(A \backslash B)_{uv} = \bigwedge_i A_{iu} \backslash B_{iv}, \quad (B/C)_{uv} = \bigwedge_j B_{uj}/C_{vj} \ .$$

This can be rewritten as $(A \backslash B)_{uv} = \min_i(B_{iv} - A_{iu})$ and $(B/C)_{uv} = \min_j(B_{uj} - C_{vj})$.

The completion operator described informally in the preceding section can be expressed via residuation, as follows.

Let us consider a heap model $\mathcal{H} = (\mathcal{A}, \mathcal{R}, R, u, l, I)$ and its associated heap automaton $\mathcal{H} = (I, \mathcal{M}, \mathbb{1})$. For $a \in \mathcal{A}$, we define the non-linear operator

$$\begin{aligned} \Phi(a) : \quad \mathbb{R}_{\max}^{\mathcal{R}} &\to \mathbb{R}_{\max}^{\mathcal{R}} \\ u &\mapsto u\Phi(a) \ , \end{aligned} \tag{12}$$

where

$$\begin{aligned} (u\Phi(a))_i = \quad \textstyle\bigwedge_{b \in R^{-1}(i)} \quad & [(u\mathcal{M}(a)\mathcal{M}(b))/\mathcal{M}(b)]_i \\ & \wedge \bigoplus_j (u\mathcal{M}(a))_j \ . \end{aligned} \tag{13}$$

We extend[3] the definition of Φ to words, by morphism. For $w = a_1 \ldots a_n \in \mathcal{A}^n$, we set:

$$\begin{aligned} \Phi(w) \ : \mathbb{R}_{\max}^{\mathcal{R}} \quad &\to \quad \mathbb{R}_{\max}^{\mathcal{R}} \\ u \quad &\mapsto \quad u\Phi(w) \stackrel{\text{def}}{=} u\Phi(a_1) \ldots \Phi(a_n) \ . \end{aligned} \tag{14}$$

Clearly, we have $u\Phi(a) \geq u\mathcal{M}(a)$. The term $(u\mathcal{M}(a)\mathcal{M}(b))/\mathcal{M}(b)$ is, by definition, the maximal contour v such that $v\mathcal{M}(b) \leq u\mathcal{M}(a)\mathcal{M}(b)$. Using this property, we can show that $u\Phi(a)\mathcal{M}(w)\mathbb{1} = u\mathcal{M}(a)\mathcal{M}(w)\mathbb{1}$, for all $w \in \mathcal{A}^*$, which implies readily the following theorem.

Theorem 4.1. *Let $\mathcal{H} = (I, \mathcal{M}, \mathbb{1})$ be a heap automaton and let Φ be defined as in (12)-(14). We have for all $w \in \mathcal{A}^*$:*

$$\mathcal{H}(w) = I\Phi(w)\mathbb{1} \ .$$

By analogy, we can see $\mathcal{H}\Phi = (I, \Phi, \mathbb{1})$ as a "non-linear automaton" which recognizes the height of the heap.

Intuitively, for any row vector u, $u\Phi(w)$ is the maximal upper contour such that the height of a heap piled up on $u\Phi(w)$ is the same as the height of a heap piled up on $u\mathcal{M}(w)$.

[3] For words are read from left to right, we write here the composition of operators from left to right. The conventional notation is $\Phi(w)(u) = \Phi(a_k) \circ \ldots \circ \Phi(a_1)(u)$.

The action of Φ admits another geometric interpretation. Let a be a piece and $\mathcal{M}(a)$ its associated matrix. We associate with a the pieces $\overline{a}$ and $\underline{a}$, which both have height 0, and have the shapes of the upper and lower contour of a, respectively. More precisely, $\overline{a}$ and $\underline{a}$ are defined by their matrices

$$\mathcal{M}(\overline{a}) = \mathrm{Id} \oplus \tilde{u}(a)u(a), \quad \mathcal{M}(\underline{a}) = \mathrm{Id} \oplus \tilde{l}(a)l(a)\ , \qquad (15)$$

where $\tilde{u}(a)$ is defined as $\tilde{l}(a)$, see (2).

An example is provided in Figure 4. For clarity, pieces of height 0 are represented by thick broken lines. We have:

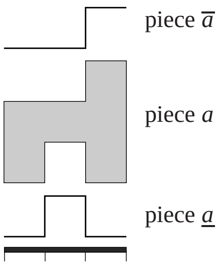

Figure 4. A piece and its associated upper and lower contour pieces.

Lemma 4.2. *Let $\mathcal{M}(a) \in \mathbb{R}_{\max}^{\mathcal{R}\times\mathcal{R}}$ be the matrix associated with the piece a of a heap model. For all $u \in \mathbb{R}_{\max}^{1\times\mathcal{R}}$ and $v \in \mathbb{R}_{\max}^{\mathcal{R}\times 1}$, we have*

$$\mathcal{M}(a)\not{/}\mathcal{M}(a) = \mathcal{M}(\underline{a}), \qquad [u\mathcal{M}(a)]\not{/}\mathcal{M}(a) = u\mathcal{M}(\underline{a})$$
$$\mathcal{M}(a)\not{\backslash}\mathcal{M}(a) = \mathcal{M}(\overline{a}), \qquad \mathcal{M}(a)\not{\backslash}[\mathcal{M}(a)v] = \mathcal{M}(\overline{a})v\ .$$

This lemma allows us to rewrite (13) in a geometrically more appealing way:

$$(u\Phi(a))_i = \bigwedge_{b\in R^{-1}(i)} [u\mathcal{M}(a)\mathcal{M}(\underline{b})]_i \wedge \bigoplus_j (u\mathcal{M}(a))_j\ .$$

This formula is illustrated, on the model of Example 2.3, in Figure 5.

The operator $\Phi(w)$ is a min-max-plus function. In particular, it satisfies the following properties:

- *monotonicity*: $\forall u, v \in \mathbb{R}^{\mathcal{R}}, u \le v \implies u\Phi(w) \le v\Phi(w)$;
- *homogeneity*: $\forall \lambda \in \mathbb{R}, u \in \mathbb{R}^{\mathcal{R}}$, $(\lambda\mathbb{1} + u)\Phi(w) = \lambda\mathbb{1} + (u\Phi(w))$, where $\lambda\mathbb{1} = (\lambda, \dots, \lambda)$;
- *non-expansiveness*: $\forall u, v \in \mathbb{R}^{\mathcal{R}}$, $\|u\Phi(w) - v\Phi(w)\| \le \|u - v\|$, where $\|\cdot\|$ denotes the sup-norm.

The larger class of maps defined by these three properties has been studied by many authors [11, 27, 21, 24]. We note in passing that it is possible, and interesting, to develop an automata theory in which the transition mappings belong to this class. For instance, as illustrated in the next section, the normalization procedure of §4.1 only needs the homogeneity property.

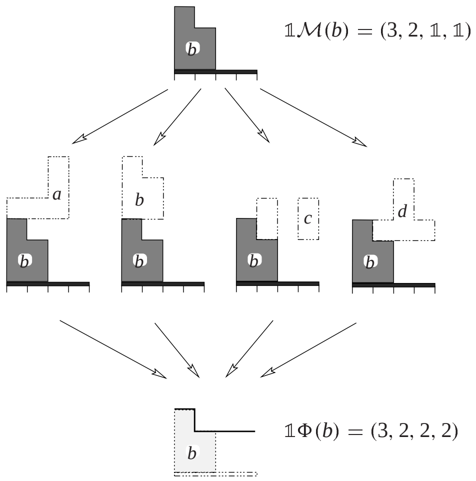

Figure 5. The contours $\mathbb{1}\mathcal{M}(b)$ and $\mathbb{1}\Phi(b)$

4.3 Determinization via Completion

Let $\mathcal{H} = (I, \mathcal{M}, \mathbb{1})$ be a heap automaton, and let $\mathcal{H}\Phi = (I, \Phi, \mathbb{1})$ be the associated non-linear automaton. We define the set of *normalized completed contours*

$$\pi(\mathcal{H}\Phi) = \{\pi(I\Phi(w)),\ w \in \mathcal{A}^*\}\ . \qquad (16)$$

If $\pi(\mathcal{H}\Phi)$ is finite, we define the *normalized completed automaton* of $\mathcal{H}$. This is the deterministic max-plus automaton $\mathcal{H}^{\det} = (\delta, \nu, \gamma)$ over the alphabet $\mathcal{A}$, where

$$\delta_u = \begin{cases} |I|_\oplus & \text{if } u = \pi(I), \\ \mathbb{0} & \text{otherwise,} \end{cases} \qquad \gamma_u = u\mathbb{1},\ \text{and,}$$

$$\nu(a)_{uv} = \begin{cases} |u\Phi(a)|_\oplus & \text{if } \pi(u\Phi(a)) = v\ , \\ \mathbb{0} & \text{otherwise.} \end{cases}$$

The proof of the following lemma is easy.

Lemma 4.3. *If $\pi(\mathcal{H}\Phi)$ is finite, the automata $\mathcal{H}$ and $\mathcal{H}^{\det}$ recognize the same map, i.e.*

$$I\mathcal{M}(w)\mathbb{1} = \delta\nu(w)\gamma,\ \forall w \in \mathcal{A}^*\ .$$

Thus, if $\pi(\mathcal{H}\Phi)$ is finite, the automaton $\mathcal{H}$ is determinizable.

Theorem 4.4. *Let $\mathcal{H} = (\mathcal{A}, \mathcal{R}, R, u, l, I)$ denote a heap model. If the pieces have rational valued shapes (i.e. $u(a), l(a) \in (\mathbb{Q} \cup \{-\infty\})^{\mathcal{R}}, \forall a \in \mathcal{A}$), and if any two pieces overlap (i.e. $R(a) \cap R(b) \neq \emptyset, \forall a, b \in \mathcal{A}$), then the set $\pi(\mathcal{H}\Phi)$ if finite, and $\mathcal{H}^{\det}$ is a deterministic automaton which recognizes the height function $\mathcal{H}$.*

In this case, as illustrated in the next section, the Lyapunov exponents $\rho_{\min}$ and ρ_E can be explicitly computed using the results of §3.

The completion procedure is central in [25], where the Lyapunov exponents of all heap models with two pieces are computed.

The timed safe Petri nets for which the associated heap model satisfies the assumptions of Theorem 4.4 are the ones for which any two transitions share at least one input or output place, i.e. the ones without concurrency.

4.4 Applications

The Dining Philosophers heap model $\mathcal{H}_P$, described in Example 2.1, does not verify the assumptions of Theorem 4.4 since pieces a and c do not overlap. The sequential heap model $\mathcal{H}_S$, introduced in Example 2.3 does verify the assumptions of Theorem 4.4. We apply to this automaton the procedure described in the previous sections.

The set of normalized completed contours is:

$$\pi(\mathcal{H}_S\Phi) = \{[0,0,0,0],[0,-1,-1,-1],[-2,-2,0,-2]\}.$$

The construction of $\mathbb{1}\Phi(b)$ starting from $\mathbb{1}\mathcal{M}(b)$ is illustrated in Figure 5. The normalized completed automaton $\mathcal{H}_S^{\text{det}}$ is represented in Figure 6.

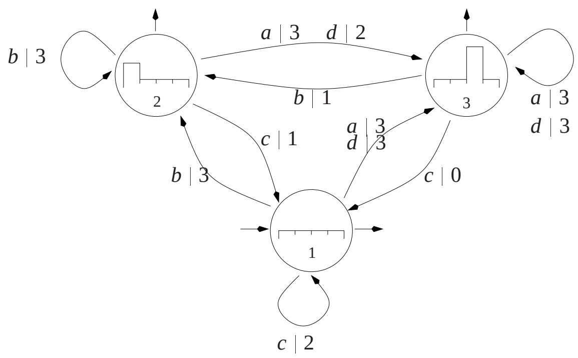

Figure 6. The automaton $\mathcal{H}_S^{\text{det}}$.

Optimal behavior. If follows from the min-plus analogue of Theorem 3.1 that the optimal behavior of $\mathcal{H}_S$ is determined by the min-plus matrix

$$M' = \begin{pmatrix} 2 & 3 & 3 \\ 1 & 3 & 2 \\ 0 & 1 & 3 \end{pmatrix}.$$

Its eigenvalue $\lambda' = 3/2$ is equal to the average weight of the circuits $3 \to 1 \to 3$ and $2 \to 3 \to 2$. These two circuits correspond to paths with label ca, cd, and db in the graph of Figure 6. Thus, the optimal case Lyapunov exponent is $\rho_{\min}(\mathcal{H}_S) = 3/2$, and infinite words realizing this optimum are $(ca)^\omega$, $(cd)^\omega$, and $(db)^\omega$.

Let us come back to the Petri net of Example 2.3. Its maximal firing rate is given by $1/\rho_{\min}(\mathcal{H}_S, L)$. The exponent $\rho_{\min}(\mathcal{H}_S, L)$ is obtained from the automaton $\mathcal{H}_S^{\text{det}} \odot \mathcal{L}$ (where $\mathcal{L}$ is a deterministic automaton recognizing L) as recalled in §3. However, here, building $\mathcal{H}_S^{\text{det}} \odot \mathcal{L}$ is not necessary as the finite prefixes of $(cd)^\omega$ belong to L. Thus, $\rho_{\min}(\mathcal{H}_S, L) = \rho_{\min}(\mathcal{H}_S)$ and the maximal firing rate is $2/3$, an infinite firing sequence of maximal rate being $(cd)^\omega$. The other optimal solutions for the heap model, for instance $(db)^\omega$, have no interpretation in terms of the Petri net.

Average case behavior Let us compute the average behavior of the heap model $\mathcal{H}_S$. Let p be a Bernoulli measure. As in (8), we associate with $\mathcal{H}_S^{\text{det}}$ a vector c and a matrix P, which is irreducible and has an unique invariant probability measure $\pi = [p(c), p(b), p(a) + p(d)]$. We deduce an expression for $\rho_E(\mathcal{H}_S)$ as a polynomial function of degree 2 of $\{p(x), x \in \mathcal{A}\}$. For instance in the uniform case, $p(x) = 1/4$, $x \in \mathcal{A}$, we obtain $\rho_E(\mathcal{H}_S) = 35/16$.

Let us come back to the Petri net $\mathcal{G}_S$. The language L of the Petri net is recognized by the Boolean deterministic automaton $\mathcal{L}$ of Figure 7 (forgetting the weights). The

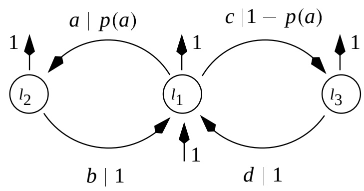

Figure 7. Automaton recognizing L and p.

random evolution of $\mathcal{G}_S$ is defined as follows. Each time there is a token in place p_3 (see Figure 3), we choose to fire transition a with probability $p(a)$ and transition c with probability $1 - p(a)$. The successive choices are independent. This defines a rational measure on words p such that $p(w) > 0$ if and only if $w \in L$. In Figure 8, we have represented the accessible part of the tensor product (in the semiring $\mathbb{R}_+$) of the linear representation of p by the linear representation $(\hat{\alpha}, \hat{\mu}, \hat{\beta})$ defined as in §3. The restriction to accessible states of the transition matrix and of the reward vector, defined by (10)-(11), are respectively

$$P = \begin{pmatrix} 0 & 0 & 0 & p(a) & 1-p(a) \\ 0 & 0 & 0 & p(a) & 1-p(a) \\ 0 & 0 & 0 & p(a) & 1-p(a) \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix},$$

and $c = [2 + p(a), 2p(a) + 1, 3p(a), 1, 3]^T$. The matrix P has a unique final class, its stationary distribution is $\pi =$

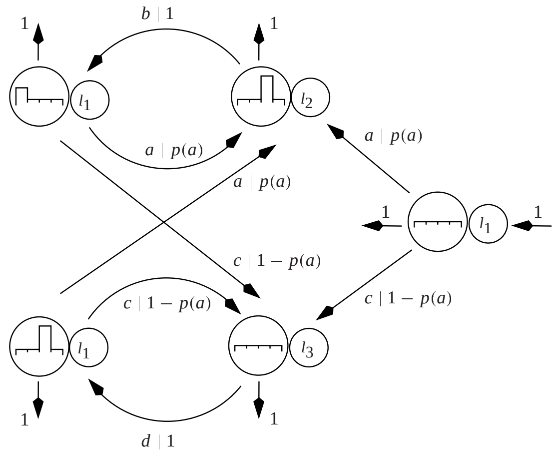

Figure 8. The accessible part of the tensor product automaton which "computes" $\rho_E(\mathcal{H}_S, L)$.

$1/2[0, p(a), 1 - p(a), p(a), 1 - p(a)]$. We deduce that $\rho_E(\mathcal{H}_S, L) = (1 + p(a))(3 - p(a))/2$. For instance, when $p(a) = 1/2$, we obtain $\rho_E(\mathcal{H}_S, L) = 15/8$. The average number of firings per time unit is $1/\rho_E(\mathcal{H}_S, L) = 8/15$.

5 Partial Determinization and Optimal Case Lyapunov Exponents

In this section, we give two constructions, based on a "partial determinization" idea. We build deterministic automata whose behaviors coincide with the behavior of the original heap automaton $\mathcal{H}$, only on some specific words. This is enough to compute $\rho_{\min}(\mathcal{H}, L)$ when L has a suitable saturation property. Throughout this section, $\mathcal{H} = (\mathcal{A}, \mathcal{R}, R, u, l, I)$ denotes a heap model, with heap automaton $(\mathbb{1}, \mathcal{M}, \mathbb{1})$.

5.1 Using Cartier-Foata Normal Forms

With the heap model $\mathcal{H}$, we associate the *trace monoid* $\mathbb{M}(\mathcal{A}, R)$, which is the quotient of $\mathcal{A}^*$ by the congruence $\sim$ generated by the relations $ab = ba$ if $R(a) \cap R(b) = \emptyset$. Two words are representatives of the same trace if they can be obtained one from the other by repeatedly interchanging adjacent commuting letters. More details on trace monoids can be found in [13]. We denote by $\mathfrak{p} : \mathcal{A}^* \to \mathbb{M}(\mathcal{A}, R)$ the canonical morphism. Our interest in trace monoids stems from the following observation:

$$\forall w, z \in \mathcal{A}^*, \quad w \sim z \implies \mathcal{M}(w) = \mathcal{M}(z) \tag{17}$$

(exchanging the piling order of pieces that do not share a column does not change the shape of a heap). Hence, the map $\mathcal{H} : \mathcal{A}^* \to \mathbb{R}_{\max}$ induces a map $\mathbb{M}(\mathcal{A}, R) \to \mathbb{R}_{\max}$, that we also denote by $\mathcal{H}$.

We say that a non-empty set $C \subset \mathcal{A}$ is a *clique* if $a, b \in C$ and $a \neq b \implies R(a) \cap R(b) = \emptyset$. We denote by $\mathcal{C}$ the set of cliques. Let $\mathfrak{q}$: $\mathcal{C}^* \to \mathbb{M}(\mathcal{A}, R)$ denote the unique morphism of monoids such that if $C = \{a_1, \dots, a_k\} \in \mathcal{C}$, $\mathfrak{q}(C) = a_1 \dots a_k$ (this product is independent of the order of $a_1, \dots, a_k$ by definition of $\mathbb{M}(\mathcal{A}, R)$). For $C \in \mathcal{C}$, we denote by $\mathfrak{n}(C) = |\mathfrak{q}(C)|$ the number of elements of C, and we extend $\mathfrak{n}$ to $\mathcal{C}^*$, by $\mathfrak{n}(C_1 \dots C_n) = \mathfrak{n}(C_1) + \dots + \mathfrak{n}(C_n)$, for all $C_1, \dots, C_n \in \mathcal{C}$. We say that $C_1 \dots C_n \in \mathcal{C}^*$ is a (Cartier-Foata) *normal form* [7, 13] if for all $2 \leq i \leq n$, $a \in C_i \implies \exists b \in C_{i-1}, R(a) \cap R(b) \neq \emptyset$. For each $w \in \mathcal{A}^*$, there is a unique normal form $z \in \mathcal{C}^*$ such that $\mathfrak{q}(z) = \mathfrak{p}(w)$, and this z is called the normal form of w. The *height* of a piece a is $\max_{i \in R(a)} u(a)_i - \min_{i \in R(a)} l(a)_i = |\mathcal{M}(a)|_{\oplus}$. We say that a piece a is *rectangular* if both $u(a)_i$ and $l(a)_i$ are independent of $i \in R(a)$.

Lemma 5.1. *If all pieces are rectangular and of height 1, then, we have for all $z \in \mathcal{C}^*$,*

$$|z| \geq \mathcal{H}(\mathfrak{q}(z)) \ . \tag{18}$$

The equality holds if z is a normal form.

The lemma follows from Viennot's observation [30] that trace monoids are 'isomorphic' to heap models with rectangular pieces of height 1. Cliques can be identified with height 1 slices of heaps, and the height of the heap associated with $w \in \mathcal{A}^*$ coincides with the number of cliques of the normal form of w (see [20, § III.B] for details).

We say that a language $L \subset \mathcal{A}^*$ is *saturated* if $L = \mathfrak{p}^{-1}\mathfrak{p}(L)$. We will need the (one-dimensional) max-plus automaton $\mathcal{H}' = (\mathbb{1}, \mathfrak{n}, \mathbb{1})$ over the alphabet $\mathcal{C}$, together with the language $L' = \mathfrak{q}^{-1}(\mathfrak{p}(L)) \subset \mathcal{C}^*$.

Theorem 5.2. *If all pieces are rectangular and of height 1, and if $L \subset \mathcal{A}^*$ is saturated, then we have*

$$\rho_{\min}(\mathcal{H}, L) = \frac{1}{\rho_{\max}(\mathcal{H}', L')} \ . \tag{19}$$

Proof. Using the first part of Lemma 5.1, we get that for all $w \in \mathcal{A}^*$ and for all $z \in \mathcal{C}^*$,

$$\mathfrak{q}(z) = \mathfrak{p}(w) \implies \frac{\mathcal{H}(w)}{|w|} \leq \frac{|z|}{\mathfrak{n}(z)} \ .$$

Hence, $\rho_{\min}(\mathcal{H}, L) = \liminf_{|w| \to \infty, w \in L} \mathcal{H}(w)/|w| \leq \liminf_{|z| \to \infty, z \in L'} |z|/\mathfrak{n}(z) = 1/\rho_{\max}(\mathcal{H}', L')$. If L is saturated, the normal form of $w \in L$ belongs to L', and the equality follows from the second part of Lemma 5.1. $\square$

Since $L = \mathcal{A}^*$ is saturated, we have by Theorem 5.2,

$$\rho_{\min}(\mathcal{H}) = \frac{1}{\rho_{\max}(\mathcal{H}')} \ ,$$

which allows us to compute $\rho_{\min}(\mathcal{H})$. More generally, $\rho_{\min}(\mathcal{H}, L)$ can be computed when L is recognizable and saturated. Then, $\mathfrak{p}(L)$ is a recognizable trace language [13, Chap. 6], and $L' = \mathfrak{q}^{-1}(\mathfrak{p}(L))$ is a recognizable language. Indeed, if L is recognized by a deterministic (Boolean) automaton with N nodes, the Nerode construction of the minimal automaton of L' shows that we can build a deterministic automaton recognizing $\mathfrak{q}^{-1}(\mathfrak{p}(L))$ with at most N nodes and $N \times |\mathcal{C}|$ arcs. Then, using the results of §3, we obtain $\rho_{\max}(\mathcal{H}', L')$ (and thus, $\rho_{\min}(\mathcal{H}, L)$) in $O(N^2|\mathcal{C}|)$ time. Note also that if L is the language of a safe timed Petri net, it is automatically recognizable and saturated [13, Prop. 1.8.2.].

The idea of using cliques of the Cartier-Foata normal form to compute Lyapunov exponents is inspired by Cérin and Petit [8], but the use of max-plus spectral theory and the trick of Lemma 5.1 reduce the complexity.

Example 5.3. We consider the simplified version of the Philosophers heap model $\mathcal{H}_P$ (see Figure 1), in which all pieces have height 1. This corresponds to firing times 1 for all the transitions and holding times 0 for all the places of the Petri net of Figure 2. The language L of this Petri net is the whole set $\{a, b, c\}^*$. The set of cliques is $\mathcal{C} = \{\{a\}, \{b\}, \{c\}, \{a, c\}\}$, and we have $\mathfrak{n}(\{a\}) = \mathfrak{n}(\{b\}) = \mathfrak{n}(\{c\}) = 1$ and $\mathfrak{n}(\{a, c\}) = 2$. Thus, $\rho_{\min}(\mathcal{H}_P, L) = \rho_{\min}(\mathcal{H}_P) = 1/\rho_{\max}(\mathcal{H}')$, where $\mathcal{H}' = (\mathbb{1}, \mathfrak{n}, \mathbb{1})$, and, by Theorem 3.1, $\rho_{\max}(\mathcal{H}') = \rho(\bigoplus_{C \in \mathcal{C}} \mathfrak{n}(C)) = 2$: any infinite heap of $\mathcal{H}$ contains an average number of at most two pieces per slice of height 1.

To give a less trivial illustration, we take the recognizable language $L = \{abc, cba\}^*$. This language is not saturated, for $L \ni abcabc \sim abacbc \notin L$, but Ochmański's theorem [13, Prop. 6.3.11] shows that the language $\overline{L} = \{z \mid z \sim w \in L\}$ obtained by saturating L is still recognizable. Alternatively, we can check that $\overline{L}$ is recognized by the automaton of Figure 9, taking into account only the plain arcs. The automaton that recognizes $\mathfrak{q}^{-1}(\mathfrak{p}(\overline{L})) \subset \mathcal{C}^*$, whose arcs are labeled by cliques, is obtained on Figure 9 by identifying a letter $x \in \mathcal{A}$ with the clique $\{x\}$, and by adding the two dashed lines. The method of §3 shows that $\rho_{\min}(\mathcal{H}, L)$ is equal to the inverse of the maximum of the quantity $\mathfrak{n}(z)/|z|$, where z is a label of a circuit of the automaton. Here, an optimal circuit is $z = (\{b\}, \{a, c\})$, which has ratio $3/2$. Thus, $\rho_{\min}$ is equal to $2/3$, and completing the circuit by a path from the input node, we obtain directly the minimizing sequences $a(bac)^\omega = abacbac\ldots$ and $c(bac)^\omega = cbacbac\ldots$

5.2 Remembering the Top Rows is Enough

The preceding technique could be extended to pieces of general (integer valued) shapes, by replacing the set of cliques of the normal form by the set of slices of height 1 which appear in all possible heaps. We present here a simpler method.

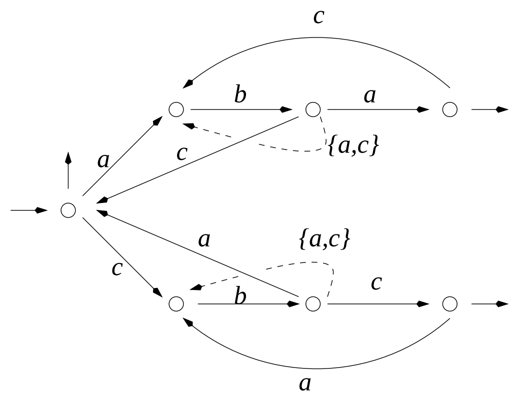

Figure 9. Lifting to the clique alphabet an automaton recognizing $\overline{\{abc, cba\}^*}$.

Recall that $x_{\mathcal{H}}(w)$ denotes the upper contour vector of the heap w (see §2). We say that a piece a is *piled at depth* $d \geq 0$ in the heap w if the bottom of piece a, piled on the heap w, is at height $\mathcal{H}(w) - d$. The following lemma states that it is always possible to build incrementally a heap of pieces, from bottom to top, without piling pieces at a large depth. The congruence $\sim$ is defined as in the previous section.

Lemma 5.4. *There is an integer $\overline{H}$ with the following property. For all $n \geq 2$ and for all $w \in \mathcal{A}^n$, we can find a word $a_1 \ldots a_n \sim w$, with $a_i \in \mathcal{A}$, such that for all $2 \leq i \leq n$, a_i is piled at depth at most $\overline{H}$ in the heap $a_1 \ldots a_{i-1}$.*

Let $H = \max_{a \in \mathcal{A}} \max_{i \in R(a)} (u(a)_i - l(a)_i) = \bigoplus_{a \in \mathcal{A}} |\mathcal{M}(a)|_\oplus$ denote the maximal height of a piece. We can show that the lemma holds with $\overline{H} = |\mathcal{R}|H$, for general pieces. When all pieces have an horizontal basis, i.e. when $l(a)_r = 0$ for all $a \in \mathcal{A}$ and $r \in R(a)$, we can take $\overline{H} = H$.

From now on, we assume that the pieces have integer valued shapes. We introduce the down shift operator $\theta : \mathbb{N}^{\mathcal{R}} \to \mathbb{N}^{\mathcal{R}}, \theta(x)_i = \max(x_i - 1, 0)$. For $L \in \mathbb{N} \setminus \{0\}$, we define the normalization operator $\mathcal{N}_L(x) = \theta^{|x|_\oplus - L}(x)$, if $|x|_\oplus \geq L$, and $\mathcal{N}_L(x) = x$, otherwise. By construction, for all $x \in \mathbb{N}^{\mathcal{R}}$, $y = \mathcal{N}_L(x)$ is such that $y \in \mathbb{N}^{\mathcal{R}}$ and $|y|_\oplus \leq L$.

We next define a deterministic max-plus automaton, $\mathcal{V}$, by its graph. The input node of $\mathcal{V}$ is the contour $x = \mathbb{1} \in \mathbb{R}_{\max}^{\mathcal{R}}$. The corresponding input arc has weight $\mathbb{1}$. There is an arc from x to $\mathcal{N}_{\overline{H}}(x\mathcal{M}(a))$, with label a and weight $\max(|x\mathcal{M}(a)|_\oplus - \overline{H}, 0)$. All nodes are output nodes, and the output arcs have weight $\mathbb{1}$. There is a finite number of nodes since all the vectors arising in this construction have integer coordinates between 0 and $\overline{H}$.

The automata $\mathcal{H}$ and $\mathcal{V}$ do not recognize the same map, but their output coincide on the words defined as in Lemma 5.4. Next theorem follows readily.

Theorem 5.5. *For all heap model with integer valued shapes of pieces, and for all saturated language L, there holds:*

$$\rho_{\min}(\mathcal{H}, L) = \rho_{\min}(\mathcal{V}, L) \ . \tag{20}$$

In particular, $\rho_{\min}(\mathcal{H}) = \rho_{\min}(\mathcal{V})$. Since the automaton $\mathcal{V}$ is deterministic, the min-plus analogue of Theorem 3.1 allows us to compute $\rho_{\min}(\mathcal{V}, L)$. Note that it is possible to reduce the number of nodes of $\mathcal{V}$ by incorporating the completion procedure of §4 in the construction of $\mathcal{V}$.

When the heap model $\mathcal{H}$ arises from a timed Petri Net, the construction of Theorem 5.5 coincides essentially with the construction of Carlier and Chretienne [6], which holds more generally for bounded (possibly not safe) timed Petri nets.

Example 5.6. The automaton $\mathcal{V}$ corresponding to the Philosopher's Petri net is depicted in Figure 10. Output arcs are omitted. For readability, only a few weights and labels are displayed. We have, $\rho_{\min}(\mathcal{V}) = \rho_{\min}(\mathcal{H}) = 6/5$, hence the maximal firing rate is $5/6$. This quantity is given by the circuit displayed in bold lines on the figure. From this circuit, we obtain the optimal firing sequence $(acbab)^{\omega}$, which corresponds to the infinite heap whose first pieces are shown on Figure 11.

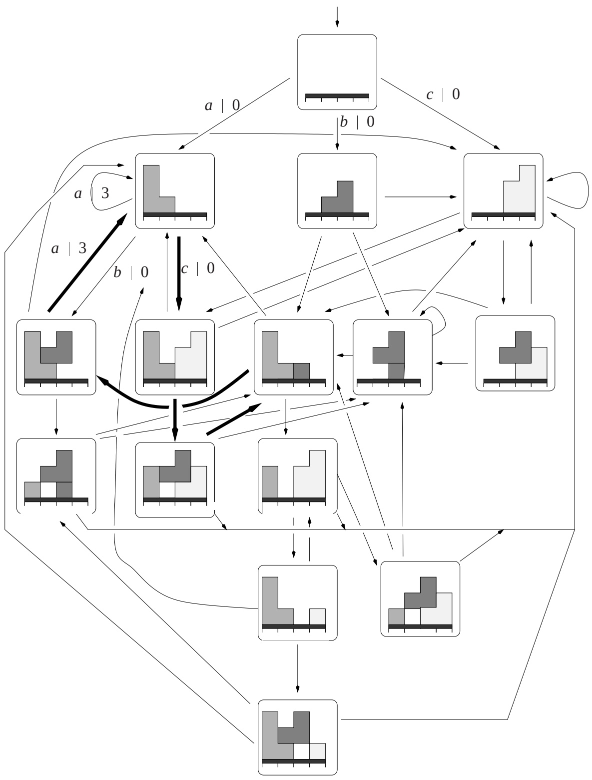

Figure 10. Computing $\rho_{\min}(\mathcal{H}_P)$

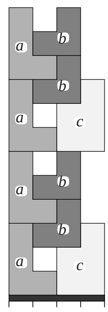

Figure 11. An optimal heap for $\mathcal{H}_P$

References

[1] F. Baccelli, G. Cohen, G. Olsder, and J. 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] T. Blyth and M. Janowitz. *Residuation Theory*. Pergamon Press, 1972.

[4] P. Bougerol and J. Lacroix. *Products of Random Matrices with Applications to Schrödinger Operators*. Progress in Probability and Statistics. Birkhäuser, 1985.

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

[6] J. Carlier and P. Chretienne. Timed Petri net schedules. In *Advances in Petri Nets*, number 340 in LNCS, pages 62–84. Springer Verlag, 1988.

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

[8] C. Cérin and A. Petit. Speedup of recognizable trace languages. In *Proc. MFCS 93*, number 711 in Lect. Notes Comput. Sci. Springer, 1993.

[9] C. Choffrut. *Contribution à l'étude de quelques familles remarquables de fonctions rationnelles*. Thèse de doctorat d'état, Université Paris VII, 1978.

[10] J. Cochet-Terrasson, G. Cohen, S. Gaubert, and M. MacGettrick. Numerical computation of spectral elements in max-plus algebra. In *IFAC Conference on System Structure and Control*, 1998.

[11] M. Crandall and L. Tartar. Some relations between nonexpansive and order preserving mappings. *Proceedings of the AMS*, 78(3):385–390, 1980.

[12] V. Diekert. *Combinatorics on traces*. Number 454 in LNCS. Springer Verlag, 1990.

[13] V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, Singapour, 1995.

[14] E. Dijkstra. Cooperating sequential processes. In F. Genvys, editor, *Programming languages*, pages 43–112. Academic Press, New York, 1968.

[15] M. Dubreil-Jacotin, L. Lesieur, and R. Croisot. *Leçons sur la Théorie des Treillis, des Structures Algébriques Ordonnées, et des Treillis géométriques*, volume 13 of *Cahiers Scientifiques*. Gauthier Villars, 1953.

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

[17] S. Gaubert. Discrete event systems and (max,+) automata. In *MOVEP'94, Nantes, June 21–24*, 1994.

[18] S. Gaubert. Performance evaluation of (max,+) automata. *IEEE Trans. Aut. Cont.*, 40(12):2014–2025, 1995.

[19] 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.

[20] S. Gaubert and J. Mairesse. Modeling and analysis of timed Petri nets using heaps of pieces. *IEEE Trans. Aut. Cont.*, 44(4):683–698, 1999.

[21] J. Gunawardena. Min-max functions. *Discrete Event Dynamic Systems*, 4:377–406, 1994.

[22] G. Hansel and D. Perrin. Mesures de probabilités rationnelles. In M. Lothaire, editor, *Mots*, pages 335–357. Hermes, 1990.

[23] R. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Math.*, 23:309–311, 1978.

[24] V. Kolokoltsov. On linear, additive and homogeneous operators in idempotent analysis. In V. Maslov and S. Samborskiĭ, editors, *Idempotent Analysis*, volume 13 of *Adv. in Sov. Math.* AMS, 1992.

[25] J. Mairesse and L. Vuillon. Optimal sequences in a heap model with two pieces. Liafa research report 98/09, Université Paris 7, 1998.

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

[27] R. Nussbaum. Hilbert's projective metric and iterated nonlinear maps. *Memoirs of the AMS*, 95(391), 1990.

[28] C. Ramchandani. *Analysis of asynchronous concurrent systems by timed Petri nets*. PhD thesis, MIT, Boston, 1974.

[29] N. Saheb. Concurrency measure in commutation monoids. *Discrete Applied Mathematics*, 24:223–236, 1989.

[30] G. 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.