

---

# Handling Markov Chains with Membrane Computing

Mónica Cardona<sup>1</sup>, M. Angels Colomer<sup>1</sup>, Mario J. Pérez-Jiménez<sup>2</sup>,  
Alba Zaragoza<sup>1</sup>

<sup>1</sup> Department of Mathematics, University of Lleida  
Av. Alcalde Rovira Roure, 191  
25198 LLeida, Spain  
{colomer, alba, mcardona}@matematica.udl.es

<sup>2</sup> Research Group on Natural Computing  
Department of Computer Science and Artificial Intelligence  
University of Sevilla  
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain  
marper@us.es

**Summary.** In this paper we approach the problem of computing the  $n$ -th power of the transition matrix of an arbitrary Markov chain through membrane computing. The proposed solution is described in a semi-uniform way in the framework of P systems with external output. The amount of resources required in the construction is polynomial in the number of states of the Markov chain and in the power. The time of execution is linear in the power and is independent of the number of states involved in the Markov chain.

## 1 Introduction

In the field of the Natural Computing, two areas that have attracted a great interest are the *molecular computing based on DNA* and, more recently, the *cellular computing with membranes*. One of the advantages of these models with respect to the classic ones is the massive parallelism that in these models is implemented in a natural way and allows the simultaneous execution of many operations in an unit of the time.

The molecular computing provides a model of computation oriented to program and so, the computing devices proposed follow a structure similar to the classic algorithms, in which the operations realized in each step depend on the result obtained in the previous step. However, the cellular computing with membranes provides a model of computation oriented to machines. In this model, the computing devices (likewise Turing machines), start from an initial configuration (a structure of membranes with certain chemical compounds in its compartments)

which evolves by means of rules of the system (abstraction of the different chemical reactions which are allowed in membranes). The rules are applied according to a specific semantic, that is to say, the execution of such devices modifies the content of their components until arriving to a halting state in which the machine can any not work longer.

The calculation of the natural powers of the transition matrix of a finite and homogenous Markov chain is important, because it allows us to estimate its limit in the case that it is convergent, and so, we can know the stationary distribution of the process. This subject has been treated in [1] where two algorithms based on DNA are described that only allow us to obtain an *estimation* of the powers. These algorithms run in polynomial time and require a polynomial amount of resources.

In this work this problem is approached within the framework of the cellular computing with membranes, and an exact solution is provided in a time which is linear in the order of the power and is independent of the number of states of the Markov chain. The amount of used resources is polynomial in the power and the number of states.

The paper is structured as follows. In the next section, basic concepts concerning Markov chains and P systems that are necessary for the development of the work are introduced. In Section 3 a P system solving (in a semi-uniform way) the problem to find the  $n$ -th power of the transition matrix, and a formal verification of the system is presented; the run time and the resources required in the description of the system are analyzed.

## 2 Preliminaries

### 2.1 Markov Chains

Roughly speaking, a (discrete-time) Markov chain is a discrete-time stochastic process such that the past is irrelevant for predicting the future given knowledge of the present: the conditional distribution of what happens in the future given everything up to now depends only on the present state, and not on the past.

More formally, a *finite Markov chain* is a sequence  $\{X_t : t \in \mathbf{N}\}$  of random variables verifying the following (Markov) property:

$$P(X_{t+1} = j / X_0 = i_0, X_1 = i_1, \dots, X_t = i_t) = P(X_{t+1} = j / X_t = i_t).$$

That is, given the present, the future does not depend of the past: the result of each event only depends on the result of the previous event.

The range of the random variables is called the *state space* of the Markov chain, and the value of  $X_t$  is interpreted as the state of the process at time  $t$ .

We suppose that the state space is finite, that is, the random variables only take the discrete values  $e_1, \dots, e_k$ , called *states* or *results*.

Hence, a Markov chain  $\{X_t : t \in \mathbf{N}\}$  provides a random process by a change of states or results  $e_1, \dots, e_k$  in certain instants of discrete times  $t \in \mathbf{N}$ , and where

the result of each event only depends on the result of the previous event. So, such a Markov chain is characterized by the conditional distribution

$$p_{ij}(t) = P(X_t = e_j / X_{t-1} = e_i), \text{ for all } t \geq 1,$$

which is called the *transition probability* of the process, providing one-step transition probability.

The matrix  $P(t) = (p_{ij}(t))_{1 \leq i, j \leq k}$  is called the *transition matrix* associated with the Markov chain  $\{X_t : t \in \mathbf{N}\}$ . The term  $(i, j)$  of the transition matrix is the probability of a transition from the state  $e_i$  to the state  $e_j$ . For that, every element of the transition probability matrix is positive, and the sum of each row is 1 because for all  $i$  ( $1 \leq i \leq k$ ) we have

$$\sum_{j=1}^k p_{ij}(t) = \sum_{j=1}^k P(X_t = e_j / X_{t-1} = e_i) = 1.$$

Hence, the matrix of transition probabilities associated with a Markov chain is stochastic. Moreover, every stochastic matrix can be viewed as the matrix of transition probabilities of some Markov chain.

We say that a finite Markov chain  $\{X_t : t \in \mathbf{N}\}$  with  $k$  states is *stationary* or *homogeneous* if the transition probabilities do not depend on the time, that is,  $\forall t \forall i, j$  ( $1 \leq i, j \leq k \rightarrow p_{ij}(t) = p_{ij}(t+1)$ ). In this case, we denote  $p_{ij}(t) = p_{ij}$ , for all  $t \in \mathbf{N}$ , and  $P = (p_{ij})_{1 \leq i, j \leq k} = P(t) = (p_{ij}(t))_{1 \leq i, j \leq k}$ .

The probability of a transition in two, three or more steps is derived in a natural way from the one-step transition probability and the Markov property. From the law of total probability, for all  $t \geq 2$  we have:

$$p_{ij}^{(2)}(t) = \sum_{r=1}^k P(X_t = e_j / X_{t-1} = e_r) \cdot P(X_{t-1} = e_r / X_{t-2} = e_i).$$

That is,  $p_{ij}^{(2)}(t) = \sum_{r=1}^k p_{rj}^{(1)}(t) \cdot p_{ir}^{(1)}(t-1)$ , where  $p_{ij}^{(1)}(t) = p_{ij}(t)$ , for all  $t \geq 1$ .

Now, if the Markov chain is homogeneous, then the transition matrix for the two-steps transition is:

$$(p_{ij}^{(2)})_{1 \leq i, j \leq k} = \left( \sum_{r=1}^k p_{rj}^{(1)} \cdot p_{ir}^{(1)} \right)_{1 \leq i, j \leq k} = P \cdot P = P^2.$$

In general, for each  $n \geq 2$  we have

$$p_{ij}^{(n)}(t) = \sum_{r=1}^k p_{rj}^{(1)}(t) \cdot p_{ir}^{(n-1)}(t-1).$$

If the Markov chain is homogeneous, then the transition matrix for the  $n$ -steps transition is:

$$(p_{ij}^{(n)})_{1 \leq i, j \leq k} = \left( \sum_{r=1}^k p_{rj}^{(1)} \cdot p_{ir}^{(n-1)} \right)_{1 \leq i, j \leq k} = P^{n-1} \cdot P = P^n.$$

The conditions

$$\begin{cases} p_{ij}^{(1)} = p_{ij}, \\ p_{ij}^{(n)} = \sum_{r=1}^k p_{rj}^{(1)} \cdot p_{ir}^{(n-1)}, \text{ for all } n \geq 2, \end{cases}$$

are called the *Kolmogorov–Chapmann equations* associated with the homogeneous Markov chain whose transition matrix is  $(p_{ij})_{1 \leq i, j \leq k}$ .

A finite and homogeneous Markov Chain  $\{X_t : t \in \mathbf{N}\}$  where the set of states is  $\{e_1, \dots, e_k\}$ , is characterized by the initial probabilities  $q_0^j = P(X_0 = e_j)$  ( $1 \leq j \leq k$ ), to get the state  $e_j$  in the first event, and the transition probability matrix  $P = (p_{ij})_{1 \leq i, j \leq k}$ .

We denote the initial probabilities by means of the vector  $q_0 = (q_0^1, \dots, q_0^k)$ , and for each  $n \geq 1$  we consider the vector  $q_n = (q_n^1, \dots, q_n^k)$ , where  $q_n^j$  ( $1 \leq j \leq k$ ) is the probability to reach the state  $e_j$  after  $n$ -steps of the random process.

Notice that we have  $q_n = q_0 \cdot P^n$ , for each  $n \geq 1$ . So, in order to determine the distribution  $q_n$  it is enough to study the matrix  $P^n$ . Moreover, the limit of the sequence  $\{P^n : n \in \mathbf{N}\}$  of these matrices allows us to obtain the distribution limit in the case that it exists. For more details see [2] and [3].

Markov chains have many applications. For example, they are used in chemical engineering (modeling the probabilities in chemical reactions and in flow systems), in biology (to model processes that are analogous to biological populations), in bioinformatics (for coding region/gene prediction), in physics (for simulation of particle systems and spatial statistics), in telecommunications (using Markov models for queues), and in geostatistics (in two or three dimensional stochastic simulations of discrete variables conditional on observed data).

## 2.2 Membrane Systems

Membrane computing is a branch of Natural Computing, considered in October 2003 by Thomson Institute for Scientific Information (**ISI**) as a *Fast Emerging Research Area* in Computer Science [6]. It was initiated at the end of 1998 by Gh. Păun (by a paper circulated at that time on web and published in 2000 [4]). Since then it has received important attention from the scientific community. Details can be found at the web page <http://psystems.disco.unimib.it>, maintained in Milano under the auspices of the European Molecular Computing Consortium.

In short, one abstracts computing models from the structure and the functioning of living cells, as well as from the organization of cell in tissues, organs, and other higher order structures. The main components of such a model are a cell-like *membrane structure*, in the *compartments* of which one places *multisets of symbol-objects* which evolve in a synchronous maximally parallel manner according to given *evolution rules*, also associated with the membranes. The objects can also

be described by strings, they can pass through membranes, can exit the system; in turn, membranes can be divided, dissolved, created.

A large variety of computing models, called P systems, were considered in this framework, based on the fundamental concept of biological membrane; the respective models are distributed (compartmentalized) parallel computing devices, processing multisets of abstract objects by means of various types of evolution rules. Parallelism, communication, non-determinism, synchronization, dynamic architecture of the model, etc. are central concepts of the theory, with biological, mathematical, and computer science sources of inspiration.

In this way, a comprehensive and systematic interdisciplinary research area was developed, of a high generality and versatility, where models can be devised for a large range of processes where compartmentalization and multiset processing are natural ingredients. Thus, although the initial goal of membrane computing was only to learn new ideas, tools, techniques from cell biology to the help of standard computers, much in the same way as, e.g., evolutionary computing suggests algorithms to be implemented on the electronic computer, the membrane computing became a new framework for building models for a large variety of processes, especially from biology (cell biology, tissues, populations of bacteria, controlling networks of complex phenomena, tumor growth, etc.), but also from linguistics, management, with several applications to computer science (computer graphics, approximative solutions to computationally hard problems, modeling parallel architectures, cryptography).

Most of these models were proven to be computationally universal, able to compute whatever a Turing machine can compute. In the case when an enhanced parallelism is available, by means of membrane division, string-object replication, or membrane creation, polynomial (often linear) time solutions to **NP**-complete problems were found.

In many variants, P systems are seen as devices of a *generative* nature, that is, from a given initial configuration several distinct computations may be developed, in a non-deterministic manner, producing different outputs.

In this paper we work with P systems with external output and performing *computing* tasks. For example, if a certain natural number,  $n$ , is encoded by the multiplicity of a special object in the initial configuration and we consider the cardinality of the multiset contained in the environment of a halting configuration as the result of a successful computation, then we can interpret this to mean that the system *computes* a partial function from natural numbers onto sets of natural numbers.

In the following, we assume that the reader is familiar with the basic notions of P systems, and we refer, for details, to [5].

### 3 Computing the $n$ -th Power of a Markov Chain

#### 3.1 Designing a P System

The goal of this paper is to compute the natural powers of a transition probability matrix associated with a finite and homogeneous Markov chain within the framework of the cellular computing with membranes. To this aim, for each Markov chain and each natural number,  $n \geq 2$ , we construct a P system with external output computing the  $n$ -th power of the matrix associated with the Markov chain. Therefore, we provide a semi-uniform solution to this problem.

Let  $P_k = (p_{ij})_{1 \leq i, j \leq k}$  be the matrix of the transition probabilities associated with a finite and homogeneous Markov chain of order  $k$ . Having in mind that  $p_{ij}$  are real numbers in  $[0, 1]$  and P systems only work with natural numbers, we have to specify the approximation to be used in order to represent those numbers in our system. In this paper, as an example, we will work with an approximation to one decimal digit, and because of that several objects appear with a factor of 10 in their multiplicities in the description of our system (similarly, if we want to work with  $m$  decimal digits then we must use a  $10^m$  factor).

Let  $n \geq 2$  be a natural number. We define a P system of degree 3 with external output,

$$\Pi(P_k, n) = (\Gamma(P_k, n), \mu(P_k, n), \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, R),$$

associated with the matrix  $P_k$  and the natural number  $n$ , computing the  $n$ -th power of  $P_k$ , as follows:

- Working alphabet:

$$\begin{aligned} \Gamma(P_k, n) = & \{s_{ij}^{(r)} : 1 \leq i, j \leq k, 0 \leq r \leq n\} \cup \{s_{ij} : 1 \leq i, j \leq k\} \cup \\ & \{t_{ij} : 1 \leq i, j \leq k\} \cup \{t_{iju}^{(r)} : 1 \leq i, j, u \leq k, 0 \leq r \leq n-1\} \cup \\ & \{p_{ij} : 1 \leq i, j \leq k\} \cup \{c_i : 1 \leq i \leq n-1\}. \end{aligned}$$

- Membrane structure:  $\mu(P_k, n) = [1 [2 [3 ]_3 ]_2 ]_1$ .
- Initial multisets:

$$\mathcal{M}_1 = \mathcal{M}_2 = \emptyset;$$

$$\mathcal{M}_3 = \{t_{ij}^{k \cdot 10 \cdot p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ii}^{(0)10} : 1 \leq i \leq k\} \cup \{c_1\}.$$

- The set  $R$  of evolution rules consists of the following rules:

- Rules in the skin membrane labeled by 1:

$$\begin{aligned} & \{t_{iju}^{(r)10p_{ij}} \longrightarrow (t_{ij}^{10p_{ij}} s_{uj}^{(r+1)10p_{ij}}, in_2) : 1 \leq i, j, u \leq k, 0 \leq r \leq n-1\} \cup \\ & \{s_{ij} \longrightarrow (s_{ij}, out) : 1 \leq i, j \leq k\}. \end{aligned}$$

- Rules in the membrane labeled by 2:

$$\begin{aligned} & \{s_{ij}^{(r)10} t_{j1}^{10p_{j1}} \dots t_{jk}^{10p_{jk}} \longrightarrow (t_{j1i}^{(r)10p_{j1}} \dots t_{jki}^{(r)10p_{jk}}, out) : 1 \leq i, j \leq k, 0 \leq r \leq \\ & n-1\} \cup \{s_{ij}^{(n)} \longrightarrow (s_{ij}, out) : 1 \leq i, j \leq k\}. \end{aligned}$$

- Rules in the membrane labeled by 3:

$$\{s_{ii}^{(0)} \longrightarrow s_{ii}^{(0)10} : 1 \leq i \leq k\} \cup \{t_{ij} \longrightarrow t_{ij}^{10} : 1 \leq i, j \leq k\} \cup \{c_i \longrightarrow c_{i+1} : 1 \leq i \leq n-2\} \cup \{c_{n-1} \longrightarrow \delta\}.$$

### 3.2 An Overview of Computations

The P system  $\Pi(P_k, n)$  works in the following way. At the beginning, the skin membrane and the membrane labeled by 2 are empty and the membrane labeled by 3 has: (a) objects  $t_{ij}$  ( $1 \leq i, j \leq k$ ) encoding the elements  $p_{ij}$  of the transition matrix of the Markov chain; (b) objects  $s_{ii}$  ( $1 \leq i \leq k$ ) encoding the states  $e_i$  of the chain; and (c) objects  $c_i$  ( $1 \leq i \leq n-1$ ) interpreted as counters used to know when a suitable number of objects  $t_{ij}$  and  $s_{ii}$  have been produced.

In the  $n-2$  first steps only rules in the internal membrane labeled by 3 are applied. During this (called) first stage, each object  $s_{ii}^{(0)}$  and each  $t_{ij}$  is replicated 10 times in each transition step. Furthermore, in membrane 3 a counter  $c_i$  appears, initialized in  $c_1$ , whose subscript increases by one unit in each step. For that reason, after  $n-2$  steps, membrane 3 contains the multiset of objects  $s_{ii}^{(0) \cdot 10^{n-1}}$   $t_{ij}^{k \cdot 10^{n-1} \cdot p_{ij}}$ , and the counter  $c_{n-1}$ . In the  $(n-1)$ -th step, each object  $s_{ii}^{(0)}$  and each object  $t_{ij}$  are replicated 10 times and, moreover, membrane 3 is dissolved, and its content passes to the internal membrane labeled by 2.

Therefore, when the system is going to execute the  $n$ -th step, the skin membrane continues being empty and the content of the internal membrane labeled by 2 is  $s_{ii}^{(0) \cdot 10^n}$   $t_{ij}^{k \cdot 10^n \cdot p_{ij}}$ .

In a second stage, in each  $(n-1+2m+1)$ -th step, with  $m \in \mathbf{N}$ , only rules of membrane 2 will be applied; they will consume all the objects  $s_{ij}^{(m)}$  and some objects  $t_{ij}$ , sending to the skin certain objects  $t_{jui}$ . In each  $(n-1+2m)$ -th step, with  $m \in \mathbf{N} - \{0\}$ , only rules in the skin are applied (because there do not exist objects  $s_{ij}$  in membrane 2), sending new objects  $t_{ij}$  and objects  $s_{ij}^{(m)}$  to membrane 2. This second phase finalizes when  $m = n-1$ .

A third stage begins with the execution of the  $(3n-1)$ -th step, and after that the skin membrane will be empty and objects  $s_{ij}^{(n)}$  appear in membrane 2. Then, the execution of the rules  $s_{ij}^{(n)} \longrightarrow (s_{ij}, out)$  sends these objects to the skin membrane, and in the following step, they are sent to the environment by means of the skin rules  $s_{ij} \longrightarrow (s_{ij}, out)$ . In this moment, no rule of the system will be applicable and so, the configuration obtained after the  $(3n+1)$ -th step will be a halting one. Moreover, in the last step, the content of the environment will be  $s_{ij}^{w_{ij}^{(n)}}$ .

Finally, it remains to show that the multiplicity  $w_{ij}^{(n)}$  of the object  $s_{ij}$  is equal to the  $(i, j)$ -term of the matrix  $10^n \cdot P_k^n$ .

### 3.3 Formal Verification

Throughout this section we are going to justify that the system  $\Pi(P_k, n)$  is deterministic and that the computation of the system codifies in the environment of the halting configuration the  $n$ -th power of the transition probability matrix,  $P_k$ , associated with a finite and homogeneous Markov chain.

First of all, let us list the necessary resources to build the system  $\Pi(P_k, n)$  from the matrix  $P_k$  and the natural number  $n \geq 2$ :

- Size of the alphabet:  $(n + 1)k^2 + k^2 + nk^3 + (n - 1) \in \Theta(nk^3)$ .
- Sum of the sizes of initial multisets:  $\leq 10k^3 + 10k + 1 \in \Theta(k^3)$ .
- Maximum of rules' lengths:  $20k + 10 \in \Theta(k)$ .
- Number of rules:  $nk^3 + k^2 + nk^2 + k^2 + k + k^2 + (n - 1) \in \Theta(nk^3)$ .

Bearing in mind the recursive description of the rules and the fact that the amount of resources is polynomial in  $n \cdot k$ , it is possible to construct the system  $\Pi(P_k, n)$  from the matrix  $P_k$  and the natural number  $n \geq 2$ , by means of a deterministic Turing machine working in polynomial time.

Next we are going to define in a recursive manner in  $r$  the expression  $w_{ij}^{(r)}$ , for each  $i, j$  such that  $1 \leq i, j \leq k$ , that is necessary in the formal verification of the system that will follow.

**Definition 1.** Let  $w_0 = 10^n$ . For each  $i, j$  such that  $1 \leq i, j \leq k$  we define:

$$w_{ij}^{(0)} = \begin{cases} w_0 & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases}$$

$$w_{ij}^{(r+1)} = \sum_{u=1}^k w_{iu}^{(r)} p_{uj}, \text{ for } r < n.$$

*Remark:* In the case of a finite and homogeneous Markov chain, with states  $e_1, \dots, e_k$  and transition matrix  $P_k$ , the definition of the values  $w_{ij}^{r+1}$  can be interpreted as an *abstraction* of the equation of Kolmogorov–Chapmann: the transition of the state  $e_i$  to the state  $e_j$  in  $(r + 1)$  steps (that is, the value  $w_{ij}^{(r+1)}$ ) is obtained from all the transitions in  $r$  steps of the state  $e_i$  to any state  $e_u$  (that is, the value  $w_{iu}^{(r)}$ ), with  $1 \leq u \leq k$ , multiplied by the transitions of  $e_u$  to  $e_j$  in only one step (that is, the value  $p_{uj}$ ).

Next we establish the relation that exists between the elements  $w_{ij}^{(r)}$  and the term  $(i, j)$  of the matrix  $10^n \cdot P_k^r$ , for each  $1 \leq r \leq n$ .

**Proposition 1.** Let  $n \geq 2$ . Let us denote  $B(r, n) = 10^n \cdot P_k^r$ , for each  $r$  such that  $r \geq 1, r \leq n$ . If  $B(r, n) = (b_{ij}^{(r, n)})_{1 \leq i, j \leq k}$ , then

$$\forall r \geq 1 (r \leq n \longrightarrow \forall i, j (1 \leq i, j \leq k \longrightarrow b_{ij}^{(r, n)} = w_{ij}^{(r)})).$$

*Proof.* By induction on  $r$ .

The base case,  $r = 1$ , follows from the following remark:

$$w_{ij}^{(1)} = \sum_{s=u}^k w_{iu}^{(0)} p_{sj} = w_0 p_{ij} = 10^n p_{ij} = b_{ij}^{(1,n)}.$$

Let  $r \geq 1$  be such that  $r < n$  and suppose that  $b_{ij}^{(r,n)} = w_{ij}^{(r)}$  is verified. Bearing in mind that  $10^n \cdot P_k^{r+1} = 10^n P_k^r \cdot P_k = B(r, n) \cdot P_k$ , we deduce that for each  $i, j$  such that  $1 \leq i, j \leq k$ :

$$b_{ij}^{(r+1,n)} = \sum_{u=1}^k b_{iu}^{(r,n)} p_{uj} \stackrel{h.i.}{=} \sum_{u=1}^k w_{iu}^{(r)} p_{uj} = w_{ij}^{(r+1)}. \quad \square$$

For each  $m \in \mathbf{N}$ , we denote by  $\mathcal{C}_m$  the configuration of the system obtained after the execution of  $m$  steps. For each label  $l \in \{1, 2, 3\}$ , we denote by  $C_m(l)$  the multiset of objects contained in the membrane labeled by  $l$  in the configuration  $\mathcal{C}_m$ . Also, we denote by  $C_m(env)$  the content of the environment of the system in the configuration  $\mathcal{C}_m$ .

From the definition of the system  $\Pi(P_k, n)$  the following holds:  $C_0(1) = C_0(2) = \emptyset$ , and  $C_0(3) = \{t_{ij}^{k \cdot 10 \cdot p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ii}^{(0)10} : 1 \leq i \leq k\} \cup \{c_1\}$ .

Next, we are going to determine the content of the different membranes of the system along the execution, and will show that after  $3n + 1$  steps the system reaches a halting configuration, in which the content of the environment is the multiset of objects  $\{s_{ij}^{w_{ij}^{(n)}} : 1 \leq i, j \leq k\}$ .

In the proof of the following result, it can be checked that there exists only one multiset of rules applicable to a non-halting configuration in each transition step, and, consequently, the membrane system  $\Pi(P_k, n)$  is deterministic.

**Theorem 1.** *Let  $n \geq 2$ .*

(a) *For each  $r \in \mathbf{N}$  such that  $1 \leq r \leq n - 2$  we have:*

$$\begin{cases} C_r(1) = \emptyset, \\ C_r(2) = \emptyset, \\ C_r(3) = \{t_{ij}^{k \cdot 10^{r+1} \cdot p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ii}^{(0)10^{r+1}} : 1 \leq i \leq k\} \cup \{c_{r+1}\}. \end{cases}$$

(b) *Moreover, we have:*

$$\begin{cases} C_{n-1}(1) = \emptyset, \\ C_{n-1}(2) = \{t_{ij}^{k \cdot 10^n \cdot p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ii}^{(0)10^n} : 1 \leq i \leq k\}. \end{cases}$$

(c) *For each  $m \in \mathbf{N}$  such that  $m < n$  we have:*

$$\begin{cases} C_{(n-1)+2m}(1) = \emptyset, \\ C_{(n-1)+2m}(2) = \{t_{ij}^{k w_0 p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ij}^{(m)w_{ij}^{(m)}} : 1 \leq i, j \leq k\}, \\ C_{(n-1)+2m+1}(1) = \{t_{ju i}^{(m)w_{ij}^{(m)} p_{ju}} : 1 \leq i, j, u \leq k\}, \\ C_{(n-1)+2m+1}(2) = \{t_{ij}^{k w_0 p_{ij} - \sum_{u=1}^k w_{ui}^{(m)} p_{ij}} : 1 \leq i, j \leq k\}. \end{cases}$$

(d) The configuration  $\mathcal{C}_{3n+1}$  is a halting one, and

$$\mathcal{C}_{3n+1}(env) = \{s_{ij}^{w_{ij}^{(n)}} : 1 \leq i, j \leq k\}.$$

*Proof.*

(a) If  $n = 2$ , then the result is obvious. Let us assume now that  $n > 2$ . We will prove the result by induction on  $r$ .

In order to prove the base case  $r = 1$ , let us observe that from the initial configuration of the system we have:  $C_0(1) = C_0(2) = \emptyset$ , and  $C_0(3) = \{t_{ij}^{k \cdot 10 \cdot p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ii}^{(0)10} : 1 \leq i \leq k\} \cup \{c_1\}$ .

Then, only in membrane 3 of configuration  $C_0$  there are applicable rules, specifically, the rules:  $s_{ii}^{(0)} \rightarrow s_{ii}^{(0)10}$ , for  $1 \leq i \leq k$ ;  $t_{ij} \rightarrow t_{ij}^{10}$ , for  $1 \leq i, j \leq k$ , and  $c_1 \rightarrow c_2$ .

Therefore,  $C_1(1) = C_1(2) = \emptyset$ , and  $C_1(3) = \{t_{ij}^{k \cdot 10^2 \cdot p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ii}^{(0)10^2} : 1 \leq i \leq k\} \cup \{c_2\}$ .

Let  $r$  be a natural number such that  $1 \leq r < n - 2$ . Let us suppose that  $C_r(1) = C_r(2) = \emptyset$ , and  $C_r(3) = \{t_{ij}^{k \cdot 10^{r+1} \cdot p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ii}^{(0)10^{r+1}} : 1 \leq i \leq k\} \cup \{c_{r+1}\}$ .

Let us note that only in membrane 3 there are applicable rules to configuration  $C_r$ ; specifically, this is the case for the rules:  $s_{ii}^{(0)} \rightarrow s_{ii}^{(0)10}$ , for  $1 \leq i \leq k$ ;  $s_{ij} \rightarrow t_{ij}^{10}$ , for  $1 \leq i, j \leq k$ ; and  $c_{r+1} \rightarrow c_{r+2}$  (recall that  $r < n - 2$  and then  $r + 1 < n - 1$ ).

Therefore,  $C_{r+1}(1) = C_{r+1}(2) = \emptyset$ , and  $C_{r+1}(3) = \{t_{ij}^{k \cdot 10^{r+2} \cdot p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ii}^{(0)10^{r+2}} : 1 \leq i \leq k\} \cup \{c_{r+2}\}$ .

(b) Directly from (a) it follows that:  $C_{n-2}(1) = C_{n-2}(2) = \emptyset$ , and  $C_{n-2}(3) = \{t_{ij}^{k \cdot 10^{n-1} \cdot p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ii}^{(0)10^{n-1}} : 1 \leq i \leq k\} \cup \{c_{n-1}\}$ .

In order to obtain the configuration  $C_{n-1}$ , let us note that only in membrane 3 there are applicable rules, namely the rules  $s_{ii}^{(0)} \rightarrow s_{ii}^{(0)10}$ , for  $1 \leq i \leq k$ ,  $t_{ij} \rightarrow t_{ij}^{10}$ , for  $1 \leq i, j \leq k$ , and  $c_{n-1} \rightarrow \delta$ .

Then, membrane 3 will be dissolved, its content go to membrane 2, and the counters  $c_i$  disappear. Thus,  $C_{n-1}(1) = \emptyset$ , and  $C_{n-1}(2) = \{t_{ij}^{k \cdot 10^n \cdot p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ii}^{(0)10^n} : 1 \leq i \leq k\}$ .

(c) We prove the result by induction on  $m$ . From (b) we deduce that  $C_{n-1}(1) = \emptyset$ , and  $C_{n-1}(2) = \{t_{ij}^{k w_0 p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ij}^{(0)w_{ij}^{(0)}} : 1 \leq i, j \leq k\}$ .

From Definition 1 we have

$$\{s_{ii}^{(0)10^n} : 1 \leq i \leq k\} = \{s_{ii}^{(0)w_0} : 1 \leq i \leq k\} = \{s_{ij}^{(0)w_{ij}^{(0)}} : 1 \leq i, j \leq k\}.$$

Let us note that only in the internal membrane (labeled by 2) there are applicable rules to configuration  $C_{n-1}$ , namely the rules:

$$s_{ij}^{(0)10} t_{j1}^{10p_{j1}} \dots t_{jk}^{10p_{jk}} \longrightarrow (t_{j1i}^{(0)10p_{j1}} \dots t_{jki}^{(0)10p_{jk}}, out), \text{ for } 1 \leq i, j \leq k.$$

By the condition of maximal parallelism, each one of these rules will be applied  $\frac{w_{ij}^{(0)}}{10}$  times. Consequently, we have:

$$\begin{cases} C_n(1) = \{t_{juu}^{(0)w_{ij}^{(0)}p_{ju}} : 1 \leq i, j, u \leq k\}, \\ C_n(2) = \{t_{ij}^{kw_0p_{ij}-w_0p_{ij}} : 1 \leq i, j \leq k\} = \{t_{ij}^{kw_0p_{ij}-\sum_{u=1}^k w_{ui}^{(0)}p_{ij}} : 1 \leq i, j \leq k\}. \end{cases}$$

Hence, the result is true for  $m = 0$ . Let  $m < n - 1$ . Let us suppose that the result holds for  $m$ , that is,

$$\begin{cases} C_{(n-1)+2m}(1) = \emptyset, \\ C_{(n-1)+2m}(2) = \{t_{ij}^{kw_0p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ij}^{(m)w_{ij}^{(m)}} : 1 \leq i, j \leq k\}, \\ C_{(n-1)+2m+1}(1) = \{t_{juu}^{(m)w_{ij}^{(m)}p_{ju}} : 1 \leq i, j, u \leq k\}, \\ C_{(n-1)+2m+1}(2) = \{t_{ij}^{kw_0p_{ij}-\sum_{u=1}^k w_{ui}^{(m)}p_{ij}} : 1 \leq i, j \leq k\}. \end{cases}$$

In order to obtain the configuration  $\mathcal{C}_{(n-1)+2m+2}$ , let us note that there are applicable rules only in the skin membrane of the configuration  $\mathcal{C}_{(n-1)+2m+1}$ . Specifically, this is the case with the rules:

$$t_{juu}^{(m)10p_{ju}} \longrightarrow (t_{ju}^{10p_{ju}} s_{iu}^{(m+1)10p_{ju}}, in_2), \text{ for } 1 \leq i, j, u \leq k.$$

By the condition of maximal parallelism, each one of these rules will be applied  $\frac{w_{ij}^{(m)}p_{js}}{10p_{js}} = \frac{w_{ij}^{(m)}}{10}$  times. Hence,

$$\begin{cases} C_{(n-1)+2m+2}(1) = \emptyset, \\ C_{(n-1)+2m+2}(2) = \{t_{ij}^{kw_0p_{ij}-\sum_{u=1}^k w_{ui}^{(m)}p_{ij}} : 1 \leq i, j \leq k\} \cup \\ \quad \{t_{ij}^{w_{1i}^{(m)}p_{ij}+\dots+w_{ki}^{(m)}p_{ij}} : 1 \leq i, j \leq k\} \cup \\ \quad \{s_{ij}^{(m+1)w_{i1}^{(m)}p_{1j}+\dots+w_{ik}^{(m)}p_{kj}} : 1 \leq i, j \leq k\} \\ = \{t_{ij}^{kw_0p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ij}^{(m+1)w_{ij}^{(m+1)}} : 1 \leq i, j \leq k\}. \end{cases}$$

In order to obtain the configuration  $\mathcal{C}_{(n-1)+2m+3}$ , let us note that there are applicable rules only in the internal membrane of the configuration  $\mathcal{C}_{(n-1)+2m+2}$  (let us recall that  $m < n - 1$ , and then  $m + 1 < n$ ). Specifically, the following rules can be applied:

$$s_{ij}^{(m+1)10} t_{j1}^{10p_{j1}} \dots t_{jk}^{10p_{jk}} \longrightarrow (t_{j1i}^{(m+1)10p_{j1}} \dots p_{jki}^{(m+1)10p_{jk}}, out), 1 \leq i, j \leq k.$$

By the condition of maximal parallelism, each one of these rules will be applied  $\frac{w_{ij}^{(m+1)}}{10}$  times. Thus,

$$\begin{cases} C_{(n-1)+2m+3}(1) = \{t_{juu}^{(m+1)w_{ij}^{(m+1)}p_{ju}} : 1 \leq i, j, u \leq k\}, \\ C_{(n-1)+2m+3}(2) = \{t_{ij}^{kw_0p_{ij}-\sum_{u=1}^k w_{ui}^{(m+1)}p_{ij}} : 1 \leq i, j \leq k\}. \end{cases}$$

Hence, the result is true for  $m + 1$ , concluding the proof of (c).

(d) From (c) we deduce that

$$\begin{cases} C_{3n-2}(1) = \{t_{ju}^{(n-1)w_{ij}^{(n-1)}p_{ju}} : 1 \leq i, j, u \leq k\}, \\ C_{3n-2}(2) = \{t_{ij}^{kw_0p_{ij} - \sum_{u=1}^k w_{ui}^{(n-1)}p_{ij}} : 1 \leq i, j \leq k\}. \end{cases}$$

Let us observe that there are applicable rules only in the skin membrane of the configuration  $\mathcal{C}_{3n-2}$ . Specifically, the following rules can be used:

$$t_{ju}^{(n-1)10p_{ju}} \longrightarrow (t_{ju}^{10p_{ju}} s_{iu}^{(n)10p_{ju}}, in_2), \text{ for } 1 \leq i, j, u \leq k.$$

By the condition of maximal parallelism, each one of these rules will be applied  $\frac{w_{ij}^{(n-1)}p_{js}}{10p_{js}} = \frac{w_{ij}^{(n-1)}}{10}$  times. Consequently, we have:

$$\begin{cases} C_{3n-1}(1) = \emptyset, \\ C_{3n-1}(2) = \{t_{ij}^{kw_0p_{ij} - \sum_{u=1}^k w_{ui}^{(n-1)}p_{ij}} : 1 \leq i, j \leq k\} \cup \\ \quad \{t_{ij}^{w_{1i}^{(n-1)}p_{ij} + \dots + w_{ki}^{(n-1)}p_{ij}} : 1 \leq i, j \leq k\} \cup \\ \quad \{s_{ij}^{(n)w_{i1}^{(n-1)}p_{1j} + \dots + w_{ik}^{(n-1)}p_{kj}} : 1 \leq i, j \leq k\} \\ = \{t_{ij}^{kw_0p_{ij}} : 1 \leq i, j \leq k\} \cup \{s_{ij}^{(n)w_{ij}^{(n)}} : 1 \leq i, j \leq k\}. \end{cases}$$

Then, to obtain the configuration  $\mathcal{C}_{3n}$ , it is possible to apply rules only in the internal membrane of the configuration  $\mathcal{C}_{3n-1}$ , namely, the following rules:

$$s_{ij}^{(n)} \longrightarrow (s_{ij}, out), \text{ for } 1 \leq i, j \leq k.$$

Thus,  $C_{3n}(1) = \{s_{ij}^{w_{ij}^{(n)}} : 1 \leq i, j \leq k\}$ , and  $C_{3n}(2) = \{t_{ij}^{kw_0p_{ij}} : 1 \leq i, j \leq k\}$ .

Then, there are applicable rules only in the skin membrane of the configuration  $\mathcal{C}_{3n}$ . Specifically, the following rules can be applied:

$$s_{ij} \longrightarrow (s_{ij}, out), \text{ for } 1 \leq i, j \leq k.$$

Hence, we have:  $C_{3n+1}(1) = \emptyset$ ;  $C_{3n+1}(2) = \{t_{ij}^{kw_0p_{ij}} : 1 \leq i, j \leq k\}$ , and  $C_{3n+1}(env) = \{s_{ij}^{w_{ij}^{(n)}} : 1 \leq i, j \leq k\}$ .

Then, there is no other applicable rule to configuration  $\mathcal{C}_{3n+1}$ . Consequently, this configuration is a halting one.  $\square$

**Theorem 2.** Let  $k \geq 1, n \geq 2$ . Let  $P_k = (p_{ij})_{1 \leq i, j \leq k}$  be the transition matrix associated with a finite and homogeneous Markov chain. Let  $\Pi(P_k, n)$  be the  $P$  system defined in Section 3.1. The output of the only computation of this system (that is, the content of the environment in the halting configuration) codifies the matrix  $B(n, n) = 10^n \cdot P_k^n$ .

*Proof.* From (d) in Theorem 1 we deduce that the configuration  $\mathcal{C}_{3n+1}$  of the system  $\Pi(P_k, n)$  is a halting one and, moreover, the multiset associated with the environment is  $C_{3n+1}(env) = \{s_{ij}^{w_{ij}^{(n)}} : 1 \leq i, j \leq k\}$ .

Directly from Proposition 1, with  $r = n$ , it follows that  $\forall i \forall j (1 \leq i, j \leq k \rightarrow w_{ij}^{(n)} = b_{ij}^{(n, n)})$ . That is, the multiplicity  $w_{ij}^{(n)}$  of the object  $s_{ij}$  in the environment of the halting configuration  $\mathcal{C}_{3n+1}$  coincides with  $b_{ij}^{(n, n)}$ , the  $(i, j)$ -term of the matrix  $B(n, n) = 10^n \cdot P_k^n$ .  $\square$

## 4 Conclusions

In this paper a deterministic P system with external output associated with a natural number,  $n \geq 2$ , and with a finite and homogeneous Markov chain, is described. This P system provides the  $n$ -th power of the transition matrix associated with the Markov chain, encoding the power in the environment of a halting configuration of the system.

In [1] this problem has been addressed by means of a molecular DNA based algorithms, giving an *estimation* of this power in polynomial time, and providing a new approach to the problem of computing the limit of a Markov chain.

The solution presented in this work is placed in the scope of the cellular computing with membranes. It is a *semi-uniform* solution, because for each Markov chain and each power, a specific P system is designed. The solution is efficient, because it is linear in the power and independent of the number of states of the Markov chain. Furthermore, the amount of resources initially required to construct the system is polynomial in the power and in the order of the Markov chain.

The paper also provides a new example of formal verification of P systems designed to solve a problem, following a specific methodology valid in some cases like those considered in the paper. These examples are always interesting, for instance, in order to find systematic processes of formal verification in a model of computation oriented to machines, like the cellular model, a framework where it is well known that the mechanisms of verification are often a very hard task.

## Acknowledgement

The second author wishes to acknowledge the support of the project TIN2005-09345-C04-01 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds.

## References

1. M. Cardona, M.A. Colomer, J. Conde, J.M. Miret, J. Miró, A. Zaragoza: Markov chains: computing limit existence and approximations with DNA. *BioSystems*, 81, 3 (2005), 261–266.
2. O. Häggström: *Finite Markov Chains and Algorithmic Applications*. London Mathematical Society, Cambridge University Press, 2002.
3. R. Nelson: *Probability, Stochastic Processes, and Queueing Theory: The Mathematics of Computer Performance Modeling*. Springer-Verlag, New York, 1995.
4. Gh. Păun: Computing with membranes. *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143, and *Turku Center for Computer Science-TUCS Report Nr. 208*, 1998.
5. Gh. Păun: *Membrane Computing. An Introduction*. Springer-Verlag, Berlin, 2002.
6. ISI web page: <http://esi-topics.com/erf/october2003.html>

