


# Classifying States of a Finite Markov Chain with Membrane Computing

View metadata, citation and similar papers at [core.ac.uk](https://core.ac.uk)

brought to you by  CORE

provided by idUS. Depósito de Investigación Universidad de Sevilla

Mario J. Pérez-Jiménez<sup>2</sup>, and Alba Zaragoza<sup>1</sup>

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

**Abstract.** In this paper we present a method to classify the states of a finite Markov chain through membrane computing. A specific P system with external output is designed for each boolean matrix associated with a finite Markov chain. The computation of the system allows us to decide the convergence of the process because it determines in the environment the classification of the states (recurrent, absorbent, and transient) as well as the periods of states. The amount of resources required in the construction is polynomial in the number of states of the Markov chain.

## 1 Introduction

Markov chains constitute an important type of stochastic processes characterized by their evolution along determinate values (called states of the process) over time. These chains represent observations of physic systems whose evolution at a future time, conditioned on their present and past values, depends only on their present value. Thus, the Markov chain loses the *memory* of its starting state.

In order to study the evolution in time of a Markov chain as well as the existence of the stationary distribution it is necessary to classify its states. This classification depends on the path structure of the chain.

In this work this problem is approached within the framework of the cellular computing with membranes. The amount of resources that we use is polynomial in the number of states. This subject has been also treated in terms of DNA computing ([1]), based on a mathematical proposition of existence rather than on the classical definition of the period of a state. This is due to the fact that DNA computing is good in detecting the existence, but it has difficulties in obtaining numerical quantifications.

The paper is structured as follows. In the next section, basic concepts concerning Markov chains and P systems are introduced. In Section 3 a semi-uniform

solution to the problem of classifying the states of a Markov chain in the framework of membrane computing is presented. Moreover, a formal verification of the system is given, and the run time and the resources required in the description of the system are analyzed.

## 2 Preliminaries

### 2.1 Markov Chains

Markov chains are a class of random processes exhibiting a certain *memoryless property* and providing a fundamental ingredient in the study of randomized algorithms. Their study is one of the main areas in modern probability theory.

A Markov process is a stochastic process that has a limited form of historical dependency. Let  $\{X(t) : t \in \tau\}$  be a stochastic process defined on the parameter  $\tau$ . We will think of  $\tau$  in terms of time and the values that  $X(t)$  can assume are called *states* which are elements of a *state space*  $S$ . In the case when the set  $\tau$  is *discrete* and the set  $S$  is *finite*, the Markov process is called a *discrete-time finite Markov chain*. We consider this kind of Markov chains because computer programs work in discrete steps and computers work with a finite amount of resources and have a finite number of states.

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, the value of  $X_{t+1}$  conditioned on the value of  $X_t$ , is independent of the values of random variables  $X_m$  for  $m < t$ .

We suppose that the state space of the chain,  $S$ , is the (finite) set of nonnegative integers  $\{e_1, \dots, e_k\}$  (whose elements are called *states* or *results*), and the chain is characterized by its evolution among these states over time.

Hence, a finite 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.

We say that a finite Markov chain is *time homogeneous* or it has *stationary transition probabilities* if the dependence between consecutive states does not change, that is,  $P(X_n = e_j / X_{n-1} = e_i) = P(X_{n+m} = e_j / X_{n+m-1} = e_i)$ , for all  $n, m \in \mathbf{N}$ ,  $e_i, e_j \in S$ . In this case, we write the transition probability as  $p_{ij} = P(X_n = e_j / X_{n-1} = e_i)$ . These probabilities form a stochastic matrix  $P = (p_{ij})$  with  $\sum_{j=1}^k p_{ij} = 1, \forall i \in \{1, \dots, k\}$ , called *transition matrix*. If  $p_{ij} \neq 0$  then we say that the transition from the state  $e_i$  to state  $e_j$  is possible.

The Markov property allows us to write an expression for the probability of a transition in one, two, three or more steps. For  $n = 1$  this probability is simply  $p_{ij}$  and is given by the position  $(i, j)$  of the transition matrix  $P$ . For  $n = 2$  the probability that the chain is in state  $e_j$  at step 2 is  $p_{ij}^{(2)} = \sum_{r=1}^k p_{ir} p_{rj}$ , and is the position  $(i, j)$  of the matrix  $P^2$ . In general, the probability that the process is in state  $e_j$   $n$  steps after being in state  $e_i$  is given by  $p_{ij}^{(n)} = \sum_{r=1}^k p_{ir}^{(m)} p_{rj}^{(n-m)}$ ,  $0 \leq m \leq n$ , and is the position  $(i, j)$  of the matrix  $P^n = P^m P^{n-m}$  (by the rules for matrix multiplication).

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  $P = (p_{ij})_{1 \leq i, j \leq k}$  ([4]).

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  $q_n = q_0 P^n$ , for each  $n \geq 1$ . So, in order to determine the distribution  $q_n$  it is enough to study the matrix  $P^n$ . In [2] the natural powers of the transition matrix of a finite and homogeneous Markov chain within the framework of membrane computing are computed. 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, and to know the stationary distribution of the process. For more details see [3] and [4].

There is a well known result [5] relating the existence of the limit of the sequence  $\{P^n : n \in \mathbf{N}\}$  with the classification of the states of the Markov chain. So, we give now a classification of the states of a Markov chain and the condition by the existence of the limit.

A state  $e_j$  is *accessible* from the state  $e_i$ , denoted by  $e_i \rightarrow e_j$ , if there is a natural number  $n > 0$  such that  $p_{ij}^{(n)} > 0$ . Two states  $e_i, e_j$  are *communicating states*, denoted by  $e_i \leftrightarrow e_j$ , if  $e_i$  is accessible from  $e_j$  and  $e_j$  is accessible from  $e_i$ . The relation of communication is an equivalence, so we can consider the equivalence classes associated with it.

The following result shows that in a finite Markov chain with  $k$  states, if we know all the paths of length  $k - 1$ , then we can know all the communicating states.

**Proposition 1.** *Let  $e_i, e_j$  be states of a finite Markov chain with  $k$  states such that  $e_j$  is accessible from  $e_i$ . Then there exists a path with length smaller than  $k$  from  $e_i$  to  $e_j$ .*

This result can be proved by substituting the nodes repeated in the path by only one copy of each of them.

A state  $e_i$  is called *recurrent* if for all  $e_j$  such that  $e_i \rightarrow e_j$ , then  $e_j \rightarrow e_i$ . On the contrary, if there exist  $j$  such that  $e_i \rightarrow e_j$  but  $e_j \not\rightarrow e_i$  (that is, there is a natural number  $m$  and a state  $e_j$  such that  $p_{ij}^{(m)} > 0$  but  $p_{ji}^{(n)} = 0$  for all  $n \in \mathbf{N}$ ), the state  $e_i$  is called *transient*. If in an equivalence class there exists a recurrent (resp. transient) state, then every state of the class is recurrent (respectively) transient. If the class of a recurrent state  $e_i$  is formed only by this state, we say that  $e_i$  is an *absorbent state* ( $p_{ij}^{(n)} = 0$  for all  $e_j \neq e_i$  and for all  $n \in \mathbf{N}$ ).

Given a state  $e_i$  such that there exists  $n > 0$  and  $p_{ii}^{(n)} > 0$ , we define its *period* as

$$d(i) = \text{g.c.d.}\{n \geq 1 \mid p_{ii}^{(n)} > 0\}.$$

All states that belong to the same class have the same period. If the period is 1, the class is said to be *aperiodic*, otherwise we refer to it as a *periodic* class.

The problem of classification is an important one in the mathematical study of Markov chains and related stochastic processes because it allows us to study their asymptotic behavior. If we think a Markov chain as a system evolving along the time, then we are interested in analyzing how that evolution is carried out. For that, we study the existence and the uniqueness of the stationary distributions, and the convergence to stationarity starting from any initial distribution. That study is related with the number of recurrent classes of a finite Markov chain.

There are some necessary conditions for the existence of stationary distributions, that is to say, there are some results which provide us with information about the existence of the limit of the sequence of the matrix powers of a finite Markov chain ([5]).

**Theorem 1.** *For any Markov chain with finite states, there exists a unique stationary distribution if and only if the set of states contains precisely one recurrent class.*

**Theorem 2.** *A necessary and sufficient condition for the existence of a limit distribution is that there is, in the set of states of the chain, exactly one aperiodic recurrent class.*

## 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 Front* in Computer Science [9]. It was initiated at the end of 1998 by Gh. Păun (by a paper circulated at that time on web and published in 2000 [6]). 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, EMCC.

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 that 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 [7].

### 3 Computing the Classification of the Steps of a Finite Markov Chain

#### 3.1 Designing a P System

The goal of this paper is to obtain the classification of the states of a finite and homogeneous Markov chain within the framework of the cellular computing with membranes.

Let  $P_k = (p_{ij})_{1 \leq i, j \leq k}$  be a boolean matrix associated with a finite and homogeneous Markov chain of order  $k$  such that  $p_{ij} = 0$  if the transition from  $e_i$  to  $e_j$  is not possible, and  $p_{ij} = 1$  if the transition from  $e_i$  to  $e_j$  is possible; that is,  $P_k$  is the adjacency matrix of the directed graph associated with the Markov chain.

The solution presented in this paper is a *semi-uniform* solution to the problem of classification, in the following sense: we give a family  $\Pi = \{\Pi(P_k) : k \in \mathbf{N}\}$ , associating with  $P_k$  a P system with external output, such that:

- There exists a deterministic Turing machine working in polynomial time which constructs the system  $\Pi(P_k)$  from  $P_k$ .
- The output of the P system  $\Pi(P_k)$  provides the classification of the  $k$  states of the Markov chain as well as the period of the states.

We associate with the matrix  $P_k$  a P system of degree 4 with external output,

$$\Pi(P_k) = (\Gamma(P_k), \mu(P_k), \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, R, \rho)$$

defined as follows:

- Working alphabet:

$$\Gamma(P_k) = \{a_{ij}, b_{ij}, d_{ij}, t_{ij} : 1 \leq i, j \leq k, \} \cup \{c_r : 0 \leq r \leq 2k + 2\} \cup \{t_{ijur} : 1 \leq i, j, u \leq k, 0 \leq r \leq k\} \cup \{\beta_i : 0 \leq i \leq \alpha + 1\} \cup \{s_{ijr} : 1 \leq i, j \leq k, 0 \leq r \leq k\} \cup \{A_{i1}, \gamma_i : 1 \leq i \leq k\} \cup \{T_{ij}, R_{ij} : 1 \leq i, j \leq k\}$$

where  $\alpha = 2k + 4 + \lceil \lg_2 k \rceil + \frac{(k-1)(k+2)}{2}$ .

- Membrane structure:  $\mu(P_k) = [1 [2 [3 [4 ]4 ]3 ]2 ]1$ .

- Initial multisets:

$$\mathcal{M}_1 = \emptyset; \mathcal{M}_2 = \{\beta_0\}; \mathcal{M}_3 = \{c_0\};$$

$$\mathcal{M}_4 = \{s_{ii0} \quad t_{ij}^{p_{ij}(k-1)} : 1 \leq i, j \leq k\}.$$

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

- Rules in the skin membrane labeled by 1:

$$r_1 = \{b_{ij}b_{ji} \rightarrow a_{ij}a_{ji} : 1 \leq i < j \leq k\}$$

$$r_2 = \{b_{ij} \rightarrow \gamma_i; \quad \gamma_i a_{ij} d_{ip} d_{jp} \rightarrow (T_{ip} T_{jp}, out) : 1 \leq i, j, p \leq k\}$$

$$r_3 = \{\gamma_i d_{ip} \rightarrow (T_{ip}, out) : 1 \leq i, j, p \leq k\}$$

$$r_4 = \{a_{ij} d_{ip} \rightarrow (R_{ip}, out) : 1 \leq i, j, p \leq k\}$$

$$r_5 = \{d_{i1} \rightarrow (A_{i1}, out) : 1 \leq i \leq k\}$$

- Rules in the membrane labeled by 2:

$$r_6 = \{b_{ij}^2 \rightarrow b_{ij} : 1 \leq i, j \leq k\} \cup \{\beta_i \rightarrow \beta_{i+1} : 0 \leq i \leq \alpha\} \cup \{\beta_{\alpha+1} \rightarrow \delta\}.$$

$$r_7 = \{d_{ij}^2 \rightarrow d_{ij} : 1 \leq i, j \leq k\}$$

$$r_8 = \{d_{ij}d_{i(j+l)} \rightarrow d_{ij}d_{il} : 1 \leq i \leq k, 2 \leq j+l \leq k\}$$

- Rules in the membrane labeled by 3:

$$r_9 = \{t_{ijur} \rightarrow (t_{ij}s_{uj(r+1)}, in_4) b_{uj} : p_{ij} = 1, u \neq j, 1 \leq i, j, u \leq k, 0 \leq r < k\}$$

$$r_{10} = \{t_{ijuk} \rightarrow (t_{ij}, in_4) b_{uj} : p_{ij} = 1, u \neq j, 1 \leq i, j, u \leq k\}$$

$$r_{11} = \{t_{ijjr} \rightarrow (t_{ij}, in_4) d_{j(r+1)} : p_{ij} = 1, 1 \leq i, j \leq k, 0 \leq r < k\}$$

$$r_{12} = \{t_{ijjk} \rightarrow (t_{ij}, in_4) : p_{ij} = 1, 1 \leq i, j \leq k\}$$

$$r_{13} = \{c_r \rightarrow c_{r+1} : 0 \leq r \leq 2k+1\} \cup \{c_{2k+2} \rightarrow \delta\}$$

- Rules in the membrane labeled by 4:

$$r_{14} = \{s_{uir}t_{i1}^{pi1} \dots t_{ik}^{pik} \rightarrow (t_{i1ur}^{pi1} \dots t_{ikur}^{pik}, out) : 1 \leq u, i \leq k, 0 \leq r \leq k\}.$$

- The partial order relation  $\rho$  over  $R$  consists of the following relations on the rules of  $R$ :
  - Priority relation in the skin membrane:  $\{r_1 > r_2 > r_3 > r_4 > r_5\}$ .
  - Priority relation in the membrane labeled by 2:  $\{r_7 > r_8\}$ .
  - Priority relation in the membranes labeled by 3:  $\emptyset$ .
  - Priority relation in the membranes 4:  $\emptyset$ .

*Remark 1.* Let us observe that the resources initially required for constructing the P system  $P_k$  are the following:

- Size of the alphabet:  $\theta(k^4)$ .
- Initial number of membranes: 4.
- Number of rules:  $O(k^4)$ .
- Maximal length of a rule:  $O(k)$ .
- Number of priority relations:  $O(k^6)$ .

### 3.2 An Overview of Computations

At the beginning, the skin membrane is empty. The membrane labeled by 2 only contains the object  $\beta_0$  which is a counter used to dissolve that membrane in the  $(\alpha+2)$ -th step, where  $\alpha = 2k+4 + \lceil lg_2k \rceil + (k-1)(k+2)/2$ . The membrane labeled by 3 contains the object  $c_0$  which is a counter used to dissolve the membrane 2 in the  $(2k+3)$ -th step. Initially, the membrane labeled by 4 contains: (a) objects  $s_{i0}$  ( $1 \leq i \leq k$ ) encoding the states  $e_i$  of the chain; and (b) objects  $t_{ij}$  ( $1 \leq i, j \leq k$ ) encoding the elements  $p_{ij}$  of the boolean matrix associated to the transition matrix of the Markov chain.

In the first  $2k+3$  steps one applies rules only in the internal membranes labeled by 2, 3, and 4. During this (so called) first stage, we determine the accessibility between states (encoded by the objects  $b_{ij}$  meaning that we can

reach  $e_j$  from  $e_i$ ) as well as the recurrent time of each state (encoded by the objects  $d_{ij}$  meaning that there exists a path from  $e_i$  to  $e_j$  with length  $j$ ). In the even steps, the rules of membrane 4 will consume all the objects  $s_{uir}$  and some objects  $t_{ij}$ , sending to membrane 3 some objects  $t_{ijur}$ . In the odd steps, only rules in membrane 3 are applied (but not in membrane 4, because there do not exist objects  $s_{uir}$  in that membrane), sending new objects  $t_{ij}$  and objects  $s_{uj(r+1)}$  (with  $u \neq j$ ) to that membrane and producing objects  $b_{uj}$  and  $d_{jr}$  in membrane 3. The first stage finalizes in the configuration  $C_{2k+3}$  when the rule  $c_{2k+3} \rightarrow \delta$  dissolves membrane 3. In this moment we have some objects  $t_{ij}$  (with  $1 \leq i, j \leq k$ ) in membrane 4, objects  $d_{jr}, b_{uj}$  (with  $1 \leq j, u, r \leq k$ ) and the object  $\beta_{2k+3}$  in membrane 2 (notice that in each step of this first stage the rule  $\beta_i \rightarrow \beta_{i+1}$  of membrane 2 has been carried out). The skin region is empty.

The second stage begins with the execution of the  $(2k+4)$ -th step. During this stage we eliminate repeated copies of objects  $b_{ij}$  and  $d_{ij}$  in membrane 2, and we compute the period of each state (encoded in the second subscript of the objects  $d$ ). The rules of membrane 2 permit transforming two copies of the object  $b_{ij}$  and  $d_{ij}$  into one copy, and the period of each state  $e_i$  is calculated by means of the rules of type  $r_8$ . For that, we need at most  $\alpha = 2k + 4 + \lceil \lg_2 k \rceil + (k-1)(k+2)/2$  steps. This stage finalizes when the rule  $\beta_{\alpha+1} \rightarrow \delta$  dissolves membrane 2 in the  $(\alpha + 2)$ -th step. This stage is a non-deterministic one.

Finally, the third stage is the output phase, and begins with the execution of the  $(\alpha + 3)$ -step. In this stage the objects  $b_{ij}b_{ji}$  are transformed into the objects  $a_{ij}a_{ji}$  by means of the rule  $r_1$  (meaning that the states  $e_i$  and  $e_j$  belongs to the same equivalence class). When this rule cannot be applied, then the transient objects are expelled to the environment applying the rules of types  $r_2$  and  $r_3$ . After that, the rule  $r_4$  sends the recurrent states and their period to the external environment. The process finalizes when the rule  $r_5$  sends the absorbent states.

### 3.3 Formal Verification

Given a computation  $\mathcal{C}$  of the P system  $\Pi(P_k)$ , 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  $l \in \{1, 2, 3, 4\}$ , 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  $\mathcal{C}_m$ .

First of all, we show that during the first stage the objects  $s_{ijr}$  codify the existence of a path from  $e_i$  to  $e_j$  with length  $r$ , and the objects  $t_{ijur}$  codify the existence of a path from  $e_u$  to  $e_j$  with length  $r$  and with  $e_i$  next to last node.

**Lemma 1.** *For each  $r$  such that  $1 \leq r \leq k$  we have the following:*

(a) *If  $r = 1$ , then for each  $i, j$  such that  $1 \leq i, j \leq k$ , the object  $t_{iji0}$  belongs to  $C_1(3)$  if and only if there exists a path from  $e_i$  to  $e_j$  with length 1 and with  $e_i$  being next to last node.*

*If  $r > 1$ , then for each  $i, j, u$  such that  $1 \leq i, j, u \leq k$ , the object  $t_{iju(r-1)}$  belongs to  $C_{(2r-1)}(3)$  if and only if there exists a path from  $e_u$  to  $e_j$  with length  $r$  and with  $e_i$  being next to last node.*



- (b) For each  $i, j$  such that  $1 \leq i, j \leq k$ ,  $i \neq j$ , the object  $s_{ijr}$  belongs to  $C_{2r}(4)$  if and only if there exists a path from  $e_i$  to  $e_j$  with length  $r$ .

*Proof.* We prove the lemma by induction on  $r$ .

– Let us suppose that  $r = 1$ .

- (a) Let  $i, j$  be such that  $1 \leq i, j \leq k$ .

If  $t_{iju0} \in C_1(3)$ , having in mind the composition of the initial configuration, there exists objects  $s_{ii0}$  and  $t_{ij}$  in  $C_0(4)$ . So,  $p_{ij} = 1$  and  $(e_i, e_j)$  is an arc of the graph associated with the Markov chain. Hence, there exists a path from  $e_i$  to  $e_j$  with length 1 and with  $e_i$  next to last node. Conversely, if there exists a path from  $e_i$  to  $e_j$  with length 1 and with  $e_i$  next to last node, then  $p_{ij} = 1$ . So, the object  $t_{ij}$  belongs to  $C_0(4)$ . Having in mind that  $s_{ii0} \in C_0(4)$ , and applying the rules of type  $r_{14}$  we have  $t_{ijj0} \in C_1(3)$ .

- (b) Let  $i, j$  be such that  $1 \leq i, j \leq k$ ,  $i \neq j$ .

Let us suppose that  $s_{ij1} \in C_2(4)$ . Then that object has been produced by an object  $t_{ijj0}$  belongs to  $C_1(3)$  and applying the rules of type  $r_9$ . From (a) we deduce that there exists a path from  $e_i$  to  $e_j$  with length 1 (and with  $e_i$  next to last node).

If there exists a path from  $e_i$  to  $e_j$  with length 1, then from (a) we deduce that the object  $t_{ijj0}$  belongs to  $C_1(3)$ . Applying the rule of type  $r_9$  we obtain that  $s_{ij1} \in C_2(4)$ .

– Let  $r \geq 1$  and  $r < k$  and let us suppose that conditions (a) and (b) hold for  $r$ . Let us show that these conditions hold for  $r + 1$ .

- (a) Let  $i, j, u$  be such that  $1 \leq i, j, u \leq k$ .

If the object  $t_{ijur}$  belongs to  $C_{2r+1}(3)$ , then in the  $(2r + 1)$ -th step the rules of type  $r_{14}$  has been applied in membrane 4, in order to produce the object  $t_{ijur}$ . Then, the objects  $s_{uir}$  and  $t_{ij}$  must belongs to  $C_{2r}(4)$ . By the induction hypothesis there exists a path from  $e_u$  to  $e_i$  of length  $r$ . Having in mind that  $t_{ij} \in C_{2r}(4)$ , it follows that  $(e_i, e_j)$  is an arc of the graph associated. Consequently there exists a path from  $e_u$  to  $e_j$  with length  $r + 1$  with  $e_i$  next to last node.

Let us suppose that there exists a path from  $e_u$  to  $e_j$  with length  $r + 1$  with  $e_i$  next to last node. Then there is a path from  $e_u$  to  $e_i$  of length  $r$ . By the induction hypothesis, the object  $s_{uir}$  belongs to  $C_{2r}(4)$ . Moreover,  $p_{ij} = 1$  because  $(e_i, e_j)$  is an arc of the graph associated, so  $t_{ij}$  belongs to  $C_{2r}(4)$ . Applying the rules of type  $r_{14}$ , we have  $t_{ijur} \in C_{2r+1}(3)$ .

- (b) Let  $i, j$  be such that  $1 \leq i, j \leq k$ ,  $i \neq j$ .

If the object  $s_{ij(r+1)}$  belongs to  $C_{2r+2}(4)$ , then there exists  $u$  ( $1 \leq u \leq k$ ) such that the object  $t_{ujir}$  belongs to  $C_{2r+1}(3)$ . By the induction hypothesis, there exists a path from  $e_i$  to  $e_j$  of length  $r + 1$  with  $e_i$  next to last node. Then, there exists a path from  $e_i$  to  $e_j$  of length  $r + 1$ .

Conversely, let us suppose that there exists a path from  $e_i$  to  $e_j$  of length  $r + 1$ . Let  $u$  be such that  $e_u$  is the next to last node of this path. By induction hypothesis, we have  $t_{ujir} \in C_{2r+1}(3)$ . Applying the rules of type  $r_9$  we obtain that the object  $s_{ij(r+1)}$  belongs to  $C_{2r+2}(4)$ .  $\square$

**Lemma 2.** For each  $r$  such that  $1 \leq r \leq k$  we have the following:

- (a) There are  $i, j, u$  such that  $1 \leq i, j, u \leq k$ ,  $t_{iju(r-1)} \in C_{2r-1}(3)$ ,  $s_{ijr} \in C_{2r}(4)$ .
- (b) For all  $i, j, u, z$  such that  $1 \leq i, j, u, z \leq k$ , we have:

$$t_{ijuz} \notin C_{2r}(3), s_{ijz} \notin C_{2r-1}(4), t_{ij}^{p_{ij}(k-1)} \in C_{2r}(4)$$

*Proof.* By induction on  $r$ . First of all, recall that

$$C_0(4) = \{s_{ii0}t_{ij}^{p_{ij}(k-1)} : 1 \leq i, j \leq k\}, C_0(3) = \{c_0\}.$$

Let  $i, j$  be such that  $1 \leq i, j \leq k$  and  $p_{ij} = 1$ . Applying the rules of type  $r_{14}$  at the initial configuration we have  $t_{iji0} \in C_1(3)$ . Then, applying the rules of type  $r_9$  in the second step we have  $s_{ij1} \in C_2(4)$ . Moreover, each object  $t_{ij}$  that has evolved in the first step, returns to membrane 4 in the next step. So,  $t_{ij}^{p_{ij}(k-1)} \in C_2(4)$ , for all  $i, j$  ( $1 \leq i, j \leq k$ ).

Having in mind that in the first step all objects  $s_{ii0}$  are consumed, we have  $s_{ijz} \notin C_1(4)$ , for all  $i, j, z$  ( $1 \leq i, j, z \leq k$ ) Hence,  $t_{ijuz} \notin C_2(3)$ , for all  $i, j, u, z$  ( $1 \leq i, j, u, z \leq k$ ).

Assuming the result holds for  $r < k$  ( $r \geq 1$ ), we prove the result holds for  $r+1$ .

By the induction hypothesis, there exist  $i, j$  ( $1 \leq i, j \leq k$ ) such that  $s_{ijr} \in C_{2r}(4)$ . But there is  $u$  ( $1 \leq u \leq k$ ) such that  $t_{uj} \in C_{2r}(4)$ ; applying the rules of type  $r_{14}$  we have we have  $t_{ijur} \in C_{2r+1}(3)$ . Then, applying the rules of type  $r_9$  in the next step we have  $s_{uj(r+1)} \in C_{2r+2}(4)$ . Moreover, each object  $t_{ij}$  that has evolved in the  $r$ -th step, returns to membrane 4 in the next step. So,  $t_{ij}^{p_{ij}(k-1)} \in C_{2r+2}(4)$ , for all  $i, j$  ( $1 \leq i, j \leq k$ ).

Having in mind that in the  $r$ -th step all objects  $s_{ijr}$  which belong to  $C_{2r}(4)$  have evolved, we have  $s_{ijz} \notin C_{2r+1}(4)$ , for all  $i, j, z$  ( $1 \leq i, j, z \leq k$ ) Hence,  $t_{ijuz} \notin C_{2r+2}(3)$ , for all  $i, j, u, z$  ( $1 \leq i, j, u, z \leq k$ ).  $\square$

**Proposition 2.** For each  $i, j$  such that  $1 \leq i, j \leq k$  we have the following:

- (1) If  $i \neq j$ , then the following assertions are equivalent:
  - (a) There exists a path from  $e_i$  to  $e_j$ .
  - (b) The object  $b_{ij}$  belongs to  $C_{2k+2}(3)$ .
  - (c) The object  $b_{ij}$  belongs to  $C_{2k+3}(2)$ .
- (2) The following conditions are equivalent
  - (a) There exists a path from  $e_i$  to  $e_i$  with length  $j$ .
  - (b) The object  $d_{ij}$  belongs to  $C_{2k+2}(3)$ .
  - (c) The object  $d_{ij}$  belongs to  $C_{2k+3}(2)$ .

*Proof.* Let  $i, j$  be such that  $1 \leq i, j \leq k$ .

- (1) Let  $i \neq j$  and let us suppose that there exists a path from  $e_i$  to  $e_j$ . Let  $r \geq 1$  be the length of that path  $r$ , and let  $e_u$  be the next to last node of that path. From Lemma 1, we have  $t_{uji(r-1)} \in C_{2r-1}(3)$ . Applying the rules of type  $r_9$  or  $r_{10}$  we obtain that  $b_{ij} \in C_{2r}(3)$ . Hence  $b_{ij} \in C_{2k+2}(3)$ .

Conversely, let us suppose that  $b_{ij} \in C_{2k+2}(3)$ . Then, from Lemma 2 there exists  $r$  ( $1 \geq r \leq k$ ) such that  $t_{uji(r-1)} \in C_{2r-1}(3)$ . From Lemma 1 we deduce that there exists a path from  $e_i$  to  $e_j$ .

Obviously,  $b_{ij} \in C_{2k+2}(3) \iff b_{ij} \in C_{2k+3}(2)$ .

(2) Let us suppose that there exists a path from  $e_i$  to  $e_i$  of length  $j$ . Then, there exists a state  $e_u$  and a path from  $e_i$  to  $e_u$  of length  $j - 1$ , and with  $(e_u, e_i)$  being an arc of the associated graph. From Lemma 1, the object  $t_{u^{ii(j-1)}}$  belongs to  $C_{2j-1}(3)$ . Applying the rules of type  $r_{11}$  or  $r_{12}$  we have  $d_{ij} \in C_{2j}(3)$ . Hence,  $d_{ij} \in C_{2k+2}(3)$ .

Conversely, let us suppose that  $d_{ij} \in C_{2k+2}(3)$ . Then, from Lemma 2 there exists  $r$  ( $1 \geq r \leq k$ ) such that  $t_{u^{ii(j-1)}} \in C_{2r-1}(3)$ . From Lemma 1 we deduce that there exists a path from  $e_i$  to  $e_i$  with length  $j$ .

Obviously,  $d_{ij} \in C_{2k+2}(3)$  if and only if  $d_{ij} \in C_{2k+3}(2)$ .  $\square$

**Proposition 3.** *If  $\alpha = 2k + 4 + \lceil \lg_2 k \rceil + (k - 1)(k + 2)/2$ , then:*

$$C_{\alpha+1}(2) = \{b_{ij} : 1 \leq i, j \leq k, i \neq j, \text{ there is a path from } e_i \text{ to } e_j\} \cup \\ \{d_{ip} : 1 \leq i, p \leq k, p \text{ is the period of the state } e_i\} \cup \{\beta_{\alpha+1}\}.$$

$$C_{\alpha+2}(1) = \{b_{ij} : 1 \leq i, j \leq k, i \neq j, \text{ there is a path from } e_i \text{ to } e_j\} \cup \\ \{d_{ip} : 1 \leq i, p \leq k, p \text{ is the period of the state } e_i\}.$$

*Proof.* Applying repeatedly the rules  $\beta_i \rightarrow \beta_{i+1}$  ( $0 \leq i \leq \alpha$ ) starting from the initial configuration, we have  $\beta_{\alpha+1} \in C_{\alpha+1}(2)$ .

From Proposition 2 we deduce that in membrane 2 of the configuration  $C_{2k+3}$  we have objects  $b_{ij}$ , with different multiplicities, such that there is a path from the state  $e_i$  to state  $e_j$ , and objects  $d_{ij}$ , with multiplicity 1, such that there is a path from the state  $e_i$  to state  $e_i$  with length  $j$ . Then, applying the rules of type  $r_6$  in, at most,  $\lceil \lg_2 k \rceil$  steps, we get that the multiplicity of each object is 1. Simultaneously, applying the rules of type  $r_7$  and (8) in at most  $\lceil \lg_2 k \rceil + (k - 1)(k + 2)/2$  steps we produce the objects  $d_{ip}$ , where  $p$  is the greatest common divisor of  $\{d_{ij} : d_{ij} \in C_{2k+3}(2)\}$ .

In the step  $\alpha + 2$ , membrane 2 is dissolved by executing the rule  $\beta_{\alpha+1} \rightarrow \delta$ .  $\square$

**Theorem 3.** *Let  $C_f$  be the final configuration of the computation  $\mathcal{C}$  of the system  $\Pi(P_k)$ . Then:*

- (a) *The state  $e_i$  is transient with period  $p$  if and only if  $T_{ip} \in C_f(\text{env})$ .*
- (b) *The state  $e_i$  is recurrent (and not absorbent) with period  $p$  if and only if  $R_{ip} \in C_f(\text{env})$ .*
- (c) *The state  $e_i$  is absorbent (with period 1) if and only if  $A_{i1} \in C_f(\text{env})$ .*

*Proof.* (a) Let us suppose that  $e_i$  is a transient state. If the equivalence class of  $e_i$  has more than one element, then we can apply the rules of type  $r_1$  in membrane 1 of the configuration  $C_{\alpha+2}$  producing objects  $a_{ij}$  and  $a_{ji}$ . In this case, there exists  $r$  ( $1 \leq r \leq k$ ) such that the object  $b_{ir}$  belongs to  $C_{\alpha+2}(1)$  but  $b_{ri} \notin C_{\alpha+2}(1)$ . So, in the  $(\alpha + 4)$ -th step the object  $\gamma_i$  is produced applying the rules of type  $r_2$ . Then in the next step (and using the object  $d_{ip}$ ) we obtain that  $T_{ip} \in C_{\alpha+5}(\text{env})$ , where  $p$  is the period of  $e_i$  (from Proposition 3).

If the equivalence class of  $e_i$  is a singleton, then the rules of type  $r_1$  cannot be applied in the configuration  $C_{\alpha+2}$ . So, we apply the rules of type  $r_2$  producing

the object  $\gamma_i$  that in the next step produces (together with the object  $d_{ip}$ ) the object  $T_{ip}$  in the environment (that is,  $T_{ip} \in C_{\alpha+4}(env)$ ).

Reciprocally, let us suppose that  $T_{ip} \in C_{\alpha+4}(env)$ ). Then, the object  $\gamma_i$  must be generated in order to can apply the rules of types  $r_2$  and/or  $r_3$ . If only the rules of type  $r_2$  are applied, then there are  $j, j'$  ( $1 \leq j, j' \leq k$ ) such that  $e_j$  is accessible from  $e_i$  and  $e_i$  is accessible from  $e_{j'}$ , and  $e_i, e_{j'}$  are communicating states. Hence,  $e_i$  is a transient state whose equivalence class has more than one object. If the rules of type  $r_3$  are applied, then  $e_i$  is a transient state whose equivalence class is a singleton.

(b) Let us suppose that the state  $e_i$  is recurrent (and not absorbent) with period  $p$ . Then, the equivalence class of  $e_i$  has more than one object and there is no transient state belongs to that class. So, the rules of type  $r_1$  will be applied in the configuration  $C_{\alpha+2}$  and the object  $\gamma_i$  cannot be produced. Hence, applying the rules of type  $r_4$  in the next configuration we have  $R_{ip} \in C_{\alpha+4}(env)$ .

Reciprocally, if  $R_{ip} \in C_f(env)$  then the rule  $a_{ij}d_{ip} \rightarrow (R_{ip}, out)$  has been applied (for some  $j, p$  with  $1 \leq j, p \leq k$ ). For that, the objects  $a_{ij}$  and  $a_{ji}$  have been produced and the object  $\gamma_i$  has not been generated. Consequently, the state  $e_i$  is recurrent and its equivalence class has more than one object (that is, it is not an absorbent state).

(c) Let us suppose that the object  $e_i$  is absorbent (consequently its period is 1). In this case, its equivalence class is a singleton. So, there is no  $j$  ( $1 \leq j \leq k$ ) such that  $b_{ij}$  and  $b_{ji}$  belongs to  $C_{\alpha+2}(1)$ . Then the rules of type  $r_1$  are not applicable for  $i$ . Applying the rule  $d_{i1} \rightarrow (A_{i1}, out)$  we obtain that  $A_{i1} \in C_{\alpha+3}(env)$ .

Reciprocally, if  $A_{i1} \in C_f(env)$ , then the object  $d_{i1}$  belongs to membrane 1 in the next to last configuration. So, the objects  $a_{ij}$  has not been produced. Then, the state is recurrent and its equivalence class has only one object.  $\square$

*Remark 2.* Let us note that the number of steps of the computation of the P system  $P_k$  is either  $\alpha + 3$  or  $\alpha + 4$ . That is, the number of steps is quadratic in the number of states of the Markov chain.

## 4 Conclusions

One of the central issues in Markov chain theory is the asymptotic long-term behavior of Markov chains.

Due to different results concerning the existence (and the uniqueness) of a stationary distribution, the problem of classification of states is an important one in the mathematical study of Markov chains and related stochastic processes.

In this paper we give an efficient (*semi-uniform*) solution of the problem of classification in the framework of the cellular computing with membranes. The solution is semi-uniform because for each adjacency matrix of the directed graph associated with a Markov chain, a specific P system with external output is designed. The solution is efficient, because it is quadratic in the number of states of the Markov chain. Furthermore, the amount of resources initially required to construct the system is polynomial 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 (in this case a problem of classification, not a decision problem), following a specific methodology. 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 case where it is well known that the mechanisms of verification are often a very hard task.

## Acknowledgement

The third author wishes to acknowledge the support of the project TIN2005-09345-C04-01 of the Ministerio de Educación y Ciencia of Spain, co-financed by FEDER funds, and of the Project of Excellence TIC 581 of the Junta de Andalucía.

## References

1. M. Cardona, M.A. Colomer, J. Miró, A. Zaragoza, A step towards DNA computation model. Submitted, 2006.
2. M. Cardona, M.A. Colomer, M.J. Pérez-Jiménez, A. Zaragoza, Handling Markov chains with membrane computing. *Lecture Notes in Computer Science*, 4135 (2006), 72–85.
3. O. Häggström, *Finite Markov Chains and Algorithmic Applications*. London Mathematical Society, Cambridge University Press, 2003.
4. R. Nelson, *Probability, Stochastic Processes, and Queueing Theory: The Mathematics of Computer Performance Modeling*. Springer-Verlag, New York, 1995.
5. A.N. Shiriyayev. *Probability*. GTM 95, Springer, 1984.
6. 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.
7. Gh. Păun, *Membrane Computing. An Introduction*. Springer-Verlag, Berlin, 2002.
8. Gh. Păun, G. Rozenberg, A guide to membrane computing. *Theoretical Computer Science*, 287 (2002), 73–100.
9. ISI web page: <http://esi-topics.com/erf/october2003.html>