Math. Struct. in Comp. Science (2009), *vol.* 19, *pp.* 1125–1139. © Cambridge University Press 2009
doi:10.1017/S0960129509990144

Efficient computation in rational-valued P systems

NADIA BUSI, MIGUEL A. GUTIÉRREZ-NARANJO,
and MARIO J. PÉREZ-JIMÉNEZ

Research Group on Natural Computing, Department of Computer Science and Artificial Intelligence, University of Sevilla, Spain
Email: {magutier,marper}@us.es

Received 17 Mar 2008; revised 14 January 2009

In memory of Nadia Busi

In this paper, we describe a new representation for deterministic rational-valued P systems that allows us to form a bridge between membrane computing and linear algebra. On the one hand, we prove that an efficient computation for these P systems can be described using linear algebra techniques. In particular, we show that the computation for getting a configuration in such P systems can be carried out by multiplying appropriate matrices. On the other hand, we also show that membrane computing techniques can be used to get the nth power of a given matrix.

1. Introduction

Since Gh. Păun introduced membrane computing in Păun (2000), a great diversity of P system variants have been presented and a big effort has gone into proving important theoretical properties of such devices. Most of these variants are computationally complete, that is, if a problem can be solved *algorithmically*, then a P system of that kind can be designed to solve the problem.

Such theoretical results are basic in the development of new computational paradigms. Moreover, an important research branch in theoretical research on membrane systems shows that whenever new membranes can be created along the computation, some P systems are able to solve **NP**-complete problems in linear time by trading space for time.

After an initial wave of theoretical results, in recent years there has been an increasing interest in practical applications within the P system community (see Ciobanu *et al.* (2006)). This paper joins this research line and proposes a new representation for rational-valued P systems that allows us to form a bridge between membrane computing and linear algebra, and enriches P systems with successful techniques. We will prove that an efficient computation can be performed for these P systems using linear algebra techniques.

Throughout this paper we consider deterministic P systems with restricted types of rules (evolution and communication rules without cooperation).

The paper is organised as follows. We first describe an example that shows the difficulty of finding effective solutions to large problems with current P system techniques. We then give a brief introduction to rational-valued P systems and present a new representation for configurations and rules in such a P system. We prove that computing a new configuration

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

from a given one can be performed by multiplying a vector by an appropriate matrix. In Section 5, we show that well-known linear algebra techniques can be applied to solve membrane computing problems using this new representation. Finally, we present some conclusions and describe some new open lines of research.

2. Motivation

We will consider the following toy world:

A factory produces a good a. It can be produced as a secondary product in the transformation of product b into c or c into b. Although, the factory needs energy, transport, storage and other production costs, they are not taken into account in this toy description. In order to settle the ideas, we consider that for each unit of raw material b the factory produces two units of a and one unit of c. Analogously, for each unit of c, two units of a and one unit of b are produced.

The factory is also able to recycle product a. In order to simplify the description, we will suppose that at each time unit, all objects a in the market are sent to the factory to be recycled. The factory recycles them and obtains the fourth part of each object a sent into the factory.

All objects a produced by the factory are sent to the market.

In this toy problem, we do not consider other factories. We only consider that there are certain numbers of objects coming out of the factory, which evolve according to some market rules. For example, in one time unit we will consider that in the market for each unit of b, one unit of a and four units of c are produced, and that each unit of c produces one unit of b.

Given a description of the stock in the factory and in the market for a time $t = 0$ (for example, one unit of b inside the factory and one unit b in the market), the problem is to know how many units of a will be in the market at some instant in the future, such as $t = 100$.

This toy problem can be easily translated into a P system. However, for the moment we will not worry about the model – technical considerations will be discussed in the next section.

We can consider our factory as a processor unit inside a market where three types of objects a, b and c are transformed or sent from or into the factory. So we can consider a membrane structure with only two membranes. The inner membrane, with label f, will represent the factory, and the outer membrane, the skin, will be labelled by s and will represent the market, that is, the membrane structure will be $[\,[\,]_f\,]_s$. The evolution of the goods described above can be easily translated into a set of P system rules:

Rule 1:	$[a]_f \to a\,[\,]_f$	**Rule 4:**	$a\,[\,]_f \to [a^{1/4}]_f$
Rule 2:	$[b \to a^2c]_f$	**Rule 5:**	$[b \to ac^4]_s$
Rule 3:	$[c \to a^2b]_f$	**Rule 6:**	$[c \to b]_s$

The description of the stock at $t = 0$ is the initial configuration of the P system. In this case $w_f = b$ and $w_s = b$.

This toy example shows how a system where the universe is split into regions with different evolution rules and communication among adjacent regions can be easily translated into a P system, where we focus on the evolution of the system instead of the results encoded by the halting computations. It can be obviously generalised to more complex situations. We can consider different departments in the factory or a market composed of more factories with a hierarchical structure of departments inside. We will

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

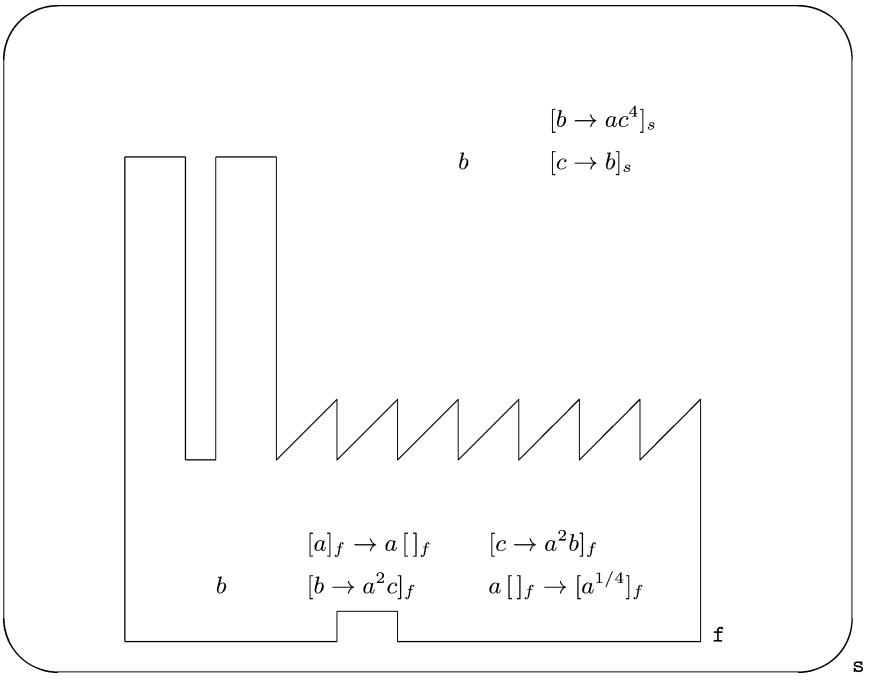


Fig. 1. An example

need a great number of membranes and rules, but the types of rules (rewriting rules inside the membranes and communication rules for the interchange of objects between adjacent regions) are the same.

Despite the advantages of describing the problem by means of a P system, it is not so clear that this description is useful for computing an effective solution to the problem. From a practical point of view, there are two possibilities within the current P system techniques for computing the disposition of the stocks at $t = 100$:

1 Carry out all the computation steps in turn until the configuration at $t = 100$ is reached. Although the new generation of software simulators has improved in efficiency compared with the earliest ones, the application of a large set of rules many times is not viable.

2 Produce a general scheme that provides the number of objects a in any configuration from the initial configuration, and prove that the general scheme gives the right answer, then obtain the solution for the case $t = 100$.
In our example, if we use $C^t(a, s)$ to denote the multiplicity of the object a in the membrane s (in the market) at time t, we have

$$C^0(a,s) = 0, \quad C^1(a,s) = 1, \quad C^2(a,s) = 2, \quad C^3(a,s) = \tfrac{25}{4},$$
$$C^4(a,s) = \tfrac{5}{2}, \quad C^5(a,s) = \tfrac{313}{16}, \quad C^6(a,s) = \tfrac{21}{8}, \quad C^7(a,s) = \tfrac{4537}{64},$$
$$C^8(a,s) = \tfrac{85}{32}, \quad C^9(a,s) = \tfrac{70585}{256} \quad C^{10}(a,s) = \tfrac{341}{128} \quad \ldots$$

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

We leave the problem of finding the general expression for the sequence $\{C^n(a,s) \mid n \in \mathbb{N}\}$ to the reader. However, in general, the task of finding the general term for such sequences is very hard.

We will come back to this problem in later sections, but before that, we will give some technical details for the P system model.

3. Rational-valued P systems

The extension of integer-valued P systems to rational-valued P systems is quite natural. A multiset over a set A is a mapping $m : A \to \mathbb{N}$ from A to the set of natural numbers. A rational-valued multiset over the set A is a mapping $m : A \to \mathbb{Q}$ from A to the set of rational numbers. The use of rational numbers in P systems is not new, and the biological motivation is that symbol a can represent a *mol* of molecules of a instead of a single molecule. In this way, it is reasonable to consider a non-integer portion of a. The remaining ingredients for P systems used in this paper are as follows:

A *rational-valued P system*[†] of degree m is a tuple of the form

$$\Pi = (\Gamma, H, \mu, w_1, \ldots, w_m, R)$$

where:

1 Γ is an alphabet of objects.
2 H is a finite set of labels.
3 μ is a *membrane structure* whose nodes are called membranes. These membranes are injectively labelled with labels from H.
4 $w_1, \ldots, w_m$ are *rational-valued multisets* over Γ associated with the membranes of μ.
5 R is a finite set of *rules*, using the following forms:
   (a) $[a \to v]_h$ where $h \in H$, $a \in \Gamma$ and v is a rational-valued multiset over Γ. These are *object evolution rules* associated with cells and depending only on the label of the cell.
   (b) $a[\,]_h \to [v]_h$ where $h \in H$, $a \in \Gamma$ and v is a rational-valued multiset over Γ. These are *send-in rules*. An object outside a membrane sends a multiset of objects into the membrane.
   (c) $[a]_h \to v[\,]_h$ where $h \in H$, $a \in \Gamma$ and v is a rational-valued multiset over Γ. These are *send-out rules*. An object inside a membrane sends a multiset of objects out of the membrane.

During the computation, the multisets associated with the membranes can change, but the alphabet Γ, the set of labels H, the membrane structure μ and the set of rules R are fixed. We will call the 4-tuple (Γ, H, μ, R) the *skeleton* of the P system.

[†] We make an abuse of notation here. In general, the expression 'rational-valued P system' denotes any P system in which the multisets can take rational values rather than just natural values. In this paper, however, we use 'rational-valued P system' for a specific variant with a membrane structure that does not change during the computation, and where the rules used consider communication and evolution of the objects. Other variants of P systems, using such further rules as division, dissolution, and so on, can also be rational-valued.

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

Rules are applied according to the following principles:

— All rules are applied in parallel: the rules are applied in one step to all objects to which they can be applied. An object in a membrane can be used by only one rule, chosen non-deterministically, but any object that can evolve by a rule of any form should evolve.
— The objects to evolve in a step and the rules by which they evolve are chosen non-deterministically, but in such a way that in each region we have a maximally parallel application of rules.
— All objects and membranes not specified in a rule and that do not evolve are passed unchanged to the next step.

We have here a double parallelism: one at the level of each membrane (the rules are used in parallel) and one at the level of the system (all membranes evolve concomitantly).

Although rational-valued P systems are defined in a general way, in this paper we only consider deterministic P systems, that is, at any step, at most one rule can be applied to each object.

4. A new point of view

The key idea of the present paper is to consider a new representation for the configurations and rules of a P system. The starting point is the representation used in Cordón-Franco *et al.* (2005), but we introduce several changes.

First, our elementary objects are pairs of the type $(a, h) \in \Gamma \times H$, meaning that the object $a \in \Gamma$ is in the membrane (labelled by) $h \in H$.

Roughly speaking, transitions in P systems are performed by rules in which the occurrence of an element a_0 in a membrane h_0 produces the occurrence of β_1 copies of the element a_1, β_2 copies of the element a_2, and so on, in a membrane h_1.

More formally, the rules in the P system model presented above can be reformulated as follows:

$$(a_0, h_0) \to (a_1, h_1)^{\beta_1}(a_2, h_1)^{\beta_2} \dots (a_n, h_1)^{\beta_n}.$$

If $h_0 \neq h_1$, we have a communication rule. In this case, both membranes must be adjacent (one membrane is the father of the other one). If h_0 is the father of h_1, we have a send-in communication rule. If the opposite holds, we have a send-out communication rule. On the other hand, if $h_0 = h_1$, we have an *evolution* rule. For each $i \in \{1, \dots, n\}$, β_i represents the multiplicity of (a_i, h_1) on the right-hand side (RHS) of the rule.

Note that according to this new representation, if all the left-hand sides (LHS) of the rules are different, the P system is deterministic. We will consider P systems such that for any object a and any membrane (labelled by) h we have a unique rule such that the LHS is (a, h), adding the *identity rule* $(a, h) \to (a, h)$ if necessary. With the identity rules, we obtain P systems whose computations may never stop. In this paper we are only interested in the evolution of computation in time and not in halting conditions.

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

4.1. *Orders and rules*

The second basic idea in the new representation consists of setting a total order in the set $\Gamma \times H$. In the rest of this paper, in order to simplify the notation, given an alphabet Γ and a set of labels H, we will use d to denote the cardinal $\Gamma \times H$. Let us consider a total order $\mathcal{O}$ on the set $\Gamma \times H$, $\mathcal{O} : \{1, \ldots, d\} \to \Gamma \times H$. Using this order, we will represent $\Gamma \times H$ as the finite sequence $\{\gamma_1, \ldots, \gamma_d\}$, where γ_i is the ith pair of $\Gamma \times H$ in the order $\mathcal{O}$. Using this order, each rule

$$(a_0, h_0) \to (a_1, h_1)^{\beta_1}(a_2, h_1)^{\beta_2} \ldots (a_n, h_1)^{\beta_n}$$

can be represented as

$$\gamma_i \to \gamma_1^{\alpha_1}\gamma_2^{\alpha_2} \ldots \gamma_d^{\alpha_d}$$

where $(a_0, h_0) = \gamma_i$ and for all $i \in \{1, \ldots, d\}$:

— if there exist $j \in \{1, \ldots, n\}$ such that $\gamma_i = (a_j, h_1)$, then $\alpha_i = \beta_j$;
— otherwise, $\alpha_i = 0$.

The use of an order on $\Gamma \times H$ leads us to a more homogeneous representation of the rule $\gamma_i \to \gamma_1^{\alpha_1}\gamma_2^{\alpha_2} \ldots \gamma_d^{\alpha_d}$. It can be represented by a pair $\langle i, \vec{v} \rangle$ where i is a natural number between 1 and d, encoding the LHS of the rule, and $\vec{v}$ is a rational valued vector of dimension d.

Definition 4.1. Consider a rational-valued P system Π with Γ the alphabet, H the set of labels and $\Gamma \times H$ the ordered set $\{\gamma_1, \ldots, \gamma_d\}$. The algebraic representation of the rule

$$\gamma_i \to \gamma_1^{\alpha_1}\gamma_2^{\alpha_2} \ldots \gamma_d^{\alpha_d}$$

is the pair $(i, \vec{v})$ where $\vec{v} = (\alpha_1, \ldots, \alpha_d)$.

Note that given an order $\{\gamma_1, \ldots, \gamma_d\}$ on $\Gamma \times H$, any pair $\langle n, \vec{v} \rangle$, where n is a natural number between 1 and d, and $\vec{v}$ is a rational-valued vector of dimension d, defines a unique rule and *vice versa*. In addition, each rule has a unique algebraic representation.

Example 4.1. Consider the P system described in Section 2. The set of objects is $\Gamma = \{a, b, c\}$ and the set of labels is $H = \{f, s\}$. Consider the following total order in $\Gamma \times H$:

$$\{(a, f), (b, f), (c, f), (a, s), (b, s), (c, s)\}$$

The six rules of the P system

Rule 1:	$[a]_f \to a\,[\,]_f$	**Rule 4:**	$a\,[\,]_f \to [a^{1/4}]_f$
Rule 2:	$[b \to a^2 c]_f$	**Rule 5:**	$[b \to ac^4]_s$
Rule 3:	$[c \to a^2 b]_f$	**Rule 6:**	$[c \to b]_s$

can be written as

r_1:	$(a, f) \to (a, s)$	r_4:	$(a, s) \to (a, f)^{1/4}$
r_2:	$(b, f) \to (a, f)^2(c, f)$	r_5:	$(b, s) \to (a, s)(c, s)^4$
r_3:	$(c, f) \to (a, f)^2(b, f)$	r_6:	$(c, s) \to (b, s)$.

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

Using the previous total order in $\Gamma \times H$, these rules have the following algebraic representation:

Rule 1:	$\langle 1,(0,0,0,1,0,0)\rangle$	**Rule 4:**	$\langle 4,(\frac{1}{4},0,0,0,0,0)\rangle$
Rule 2:	$\langle 2,(2,0,1,0,0,0)\rangle$	**Rule 5:**	$\langle 5,(0,0,0,1,0,4)\rangle$
Rule 3:	$\langle 3,(2,1,0,0,0,0)\rangle$	**Rule 6:**	$\langle 6,(0,0,0,0,1,0)\rangle$.

4.2. *Configurations*

A *configuration* of such a rational-valued P system is the description of the multiset placed in the membranes of the P system at a given moment. Formally, given a rational-valued P system with working alphabet Γ and set of labels H, a configuration C is a rational-valued multiset over $\Gamma \times H$, and we use $C(a,m)$ to denote the multiplicity of the object a in the membrane labelled by m of that configuration. The support of C, $supp(C)$, is defined as $supp(C) = \{(a,m) \in \Gamma \times H \mid C(a,m) \neq 0\}$ and, as usual in multisets theory, C will be represented as $\{(a,m)^{C(a,m)} \mid (a,m) \in supp(C)\}$. For example, the initial configuration of our example $[\,[b]_f\, b\,]_s$ can be represented as $\{(b,f),(b,s)\}$. This representation is adequate because the membrane structure does not change during the evolution.

In order to formalise the concept of computation with this new representation, we will fix some notation. Consider the order $\{\gamma_1,\ldots,\gamma_d\}$ on $\Gamma \times H$. For each i $(1 \leqslant i \leqslant d)$ we will use r_i to denote the unique rule having the pair γ_i as the LHS of the rule. We also use RHS_i to denote the right-hand side of the rule r_i, and for all $\sigma \in \Gamma \times H$, we use $|RHS_i(\sigma)|$ to denote the multiplicity of σ in RHS_i.

Example 4.2. Consider the P system described in Section 2, and let

$$((a,f),(b,f),(c,f),(a,s),(b,s),(c,s))$$

be the total order on $\Gamma \times H$. Since (b,s) is the fifth element in the order, we have $r_5 : (b,s) \to (a,s)(c,s)^4$, $RHS_5 = (a,s)(c,s)^4$ and $|RHS_5(c,s)| = 4$.

In order to obtain a new configuration C' from a given configuration C and from the set of rules $\{r_i \mid\ 1 \leqslant i \leqslant d\}\}$, we need to describe the multiplicity of any $\sigma \in \Gamma \times H$ in C'. In such a multiplicity, each rule $r_i : \gamma_i \to RHS_i$ adds the multiplicity of σ on the right-hand side of the rule multiplied by the multiplicity of γ_i in the configuration C. Formally, for every r $(1 \leqslant r \leqslant d)$, we have

$$C'(\gamma_r) = \sum_{i=1}^{d} C(\gamma_i) \cdot |RHS_i(\gamma_r)|.$$

Example 4.3. In our example, the initial configuration is $\mathscr{C}_0 = \{(b,s),(b,f)\}$. In order to know the multiplicity of the pair (a,s) in the configuration C_1, it suffices to compute

$$\begin{aligned} C'((a,s)) &= C(\gamma_1)\cdot|RHS_1(a,s)| + C(\gamma_2)\cdot|RHS_2(a,s)| + C(\gamma_3)\cdot|RHS_3(a,s)| \\ &+ C(\gamma_4)\cdot|RHS_4(a,s)| + C(\gamma_5)\cdot|RHS_5(a,s)| + C(\gamma_6)\cdot|RHS_6(a,s)| \\ &= 0\cdot 1 + 1\cdot 0 + 0\cdot 0 + 0\cdot 0 + 1\cdot 1 + 0\cdot 0 \\ &= 1. \end{aligned}$$

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

Given the idea of fixing an order on $\Gamma \times H$, the representation of a configuration by means of a vector is quite natural.

Definition 4.2. Consider a rational-valued P system Π with Γ the alphabet, H the set of labels and $\Gamma \times H$ the ordered set $\{\gamma_1, \ldots, \gamma_d\}$. An algebraic representation of a configuration $C : \Gamma \times H \to \mathbb{R}$ is a rational-valued vector

$$\vec{C} = (C(\gamma_1), \ldots, C(\gamma_d)).$$

That is, the jth element in $\vec{C}$ is a rational number representing the multiplicity of the jth element of $\Gamma \times H$.

Example 4.4. As we have already seen, the initial configuration $[\,[b]_f\, b\,]_s$ can be expressed as the multiset $C = \{(b, f), (b, s)\}$. If we consider the order

$$\{(a, f), (b, f), (c, f), (a, s), (b, s), (c, s)\},$$

then the algebraic representation of the configuration is $\vec{C} = (0, 1, 0, 0, 1, 0)$.

Note that there exists a bijective correspondence between a configuration C and its algebraic representation $\vec{C}$.

4.3. *Matrices associated with sets of rules*

Having defined the algebraic representation of rules and a configuration, we will now define a matrix to compute a new configuration from a given one.

Definition 4.3. Consider a rational-valued P system Π with Γ the alphabet, H the set of labels and $\Gamma \times H$ the ordered set $\{\gamma_1, \ldots, \gamma_d\}$. Let $R = \{\langle 1, \vec{v_1}\rangle, \ldots \langle d, \vec{v_d}\rangle\}$ be the set of algebraic representations of the rules of Π. The $d \times d$ matrix M_Π whose rows are the vectors $\vec{v_1}, \ldots, \vec{v_d}$, respectively, is the matrix associated with Π.

Notice that the matrix associated with a given P system depends only on the skeleton of the P system and not on a specific initial configuration.

Example 4.5. In our example, the matrix associated with the P system Π is

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

The definition of these algebraic objects allows us to define an algebraic method for obtaining a new configuration from a given one.

Theorem 4.1. Let Π a rational-valued P system Π with Γ the alphabet, H the set of labels and $\Gamma \times H$ the ordered set $\{\gamma_1, \ldots, \gamma_d\}$. Let M_Π be its associated matrix. Given

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

a configuration C, the next configuration C' of the P system can be obtained as the configuration associated with the algebraic configuration $\vec{C} \cdot M_\Pi$.

Proof. It suffices to show that for every r $(1 \leqslant r \leqslant d)$, the multiplicity of γ_r in C' coincides with the rth component of the vector $\vec{C} \cdot M_\Pi$.

Let C be the given configuration and $\{\gamma_i \to RHS_i \mid\ 1 \leqslant i \leqslant d\}$ be the set of rules of the P system. Then, the configuration obtained from C by the application of the rules is the new configuration C', verifying that for every r $(1 \leqslant r \leqslant d)$

$$C'(\gamma_r) = \sum_{i=1}^{d} C(\gamma_i) \cdot |RHS_i(\gamma_r)|.$$

On the other hand, the rth component of the vector $\vec{C} \cdot M_\Pi$ is

$$(\vec{C} \cdot M_\Pi)_r = \sum_{j=1}^{d} \vec{\mathscr{C}}_j \cdot (M_\Pi)_{jr}$$

where $\vec{\mathscr{C}}_j$ is the jth component of the vector $\vec{C}$ and $(M_\Pi)_{jr}$ is the jth element in the rth row of the matrix M_Π. Let us analyse such a $\vec{\mathscr{C}}_j$ and $(M_\Pi)_{jr}$.

— By definition, the jth component of the vector $\vec{C}$ is the multiplicity of γ_j in the configuration C, that is, $C(\gamma_j)$.
— The jth element in the rth row of the matrix M_Π is the multiplicity of γ_r in the RHS of the rule r_j, that is, $|RHS_j(\gamma_r)|$.

Therefore, for every r $(1 \leqslant r \leqslant d)$, we have

$$(\vec{C} \cdot M_\Pi)_r = \sum_{j=1}^{d} \vec{\mathscr{C}}_j \cdot (M_\Pi)_{jr} = \sum_{i=1}^{d} C(\gamma_i) \cdot |RHS_i(\gamma_r)| = C'(\gamma_r). \qquad \square$$

Example 4.6. In our example we consider the initial configuration C_0 with $w_f = w_s = 1$: $[\,b\,]_f\, b\,]_s$. Applicable rules are rule **2:** $[b \to a^2c]_f$ and rule **5:** $[b \to ac^4]_s$, so

$$C_1 = [\,[\,a^2c]_f\, ac^4\,]_s.$$

In order to obtain the next configuration we should consider rules **1:** $[a]_f \to a\,[\,]_f$, **3:** $[c \to a^2b]_f$, **4:** $a\,[\,]_f \to [a^{1/2}]_f$ and **6:** $[c \to b^5]_f$. We obtain

$$C_2 = [\,[\,a^{\frac{9}{4}}b]_f\, a^2b^4\,]_s$$

Using the algebraic representation, $\vec{C}_0 = (0,1,0,0,1,0)$ and considering the matrix M_Π from Example 4.5, we get

$$\vec{C}_1 = \vec{C}_0 \cdot M_\Pi = (2,0,1,1,0,4)$$

and

$$\vec{C}_2 = \vec{C}_1 \cdot M_\Pi = \left(\frac{9}{4},1,0,2,4,0\right).$$

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

5. From membrane computing to linear algebra

From Theorem 4.1 we deduce that in order to know the nth configuration of a P system Π from a given configuration C_0, it is only necessary to consider the algebraic expression of C_0, $\vec{C}_0$ and the application matrix M_Π, and compute

$$\vec{C}_n = \vec{C}_0 \cdot M_\Pi^n$$

where M_Π^n is the nth power of the matrix M_Π. Then, we just have to translate $\vec{C}_n$ into C_n to get the desired configuration.

The translation of the membrane computing techniques into linear algebra opens up a new perspective, since the use of rational number and simple arithmetic operations such as addition and multiplication can elucidate the evolution of the P systems.

However, this solution is not suitable for real-world problems since the dimension of the matrix is $d \times d$, where d is the product of the number of objects of the alphabet and the number of membranes. Computing the nth power of the $d \times d$ matrix M_Π is a very hard task whenever n is large enough. Using the Coppersmith–Winograd algorithm (Coppersmith and Winograd 1990), this problem can be solved in time $O(n \cdot d^{2.376})$.

Fortunately, this problem has been studied deeply in linear algebra and a successful solution has been found: given a square matrix A, it is possible to find two square matrices P and B such that $A = P \cdot B \cdot P^{-1}$ with B *as simple as possible*[†]. In the best case, B is a diagonal matrix.

In this way, given a square matrix A and its associated matrices P and B, if $A = P \cdot B \cdot P^{-1}$, we have $A^n = P \cdot B^n \cdot P^{-1}$, for all $n \in \mathbb{N}$. The computation of B and P is hard if the order of A is high, but it is justified if the power n is large enough.

Example 5.1. Given our application matrix M_Π, we consider the matrices

$$P = \begin{pmatrix} 0 & 0 & \frac{15}{2} & -\frac{15}{2} & 0 & 0 \\ 0 & -1 & -10 & 30 & 1 & 0 \\ 0 & 1 & -10 & 30 & 1 & 0 \\ 0 & 0 & -\frac{15}{4} & -\frac{15}{4} & 0 & 0 \\ -2 & 0 & -\frac{1}{2} & \frac{1}{2} & 0 & 2 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

The nth power of B is

$$B^n = \begin{pmatrix} (-2). & 0 & 0 & 0 & 0 & 0 \\ 0 & (-1)^n & 0 & 0 & 0 & 0 \\ 0 & 0 & (-\frac{1}{2})^n & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2^n} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2^n \end{pmatrix}$$

[†] Such matrix is called the *Jordan canonical form* of the matrix, see Horn and Johnson (1985) for details.

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

Bearing in mind that P^{-1} is the matrix

$$P^{-1} = \begin{pmatrix} -\frac{1}{60} & 0 & 0 & \frac{2}{15} & -\frac{1}{4} & \frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{15} & 0 & 0 & -\frac{2}{15} & 0 & 0 \\ -\frac{1}{15} & 0 & 0 & -\frac{2}{15} & 0 & 0 \\ \frac{8}{3} & \frac{1}{2} & \frac{1}{2} & \frac{8}{3} & 0 & 0 \\ \frac{1}{60} & 0 & 0 & \frac{2}{15} & \frac{1}{4} & \frac{1}{2} \end{pmatrix}$$

if we use $v_4(n)$ to denote the fourth component of $\vec{C}_n$ and considering $\vec{C}_n = \vec{C}_0 \cdot P \cdot B^n \cdot P^{-1}$, then

$$v_4(n) = \frac{8}{3} + \frac{1}{15}\left(-\frac{1}{2}\right)^n - \frac{1}{2^{n-2}} + \frac{1}{3}(-1)^n \frac{1}{2^{n-2}} - \frac{1}{15}\frac{1}{2^n} + \frac{2^{n+2}}{15} - \frac{1}{15}(-1)^n 2^{n+2}.$$

Finally, $v_4(n)$ can be expressed in a simpler way as

$$v_4(n) = \begin{cases} \frac{1}{15}\,\frac{1}{2^n}(40\,2^n + 2^{2n+3} - 82) & \text{if } n \text{ is odd} \\ \frac{8}{3}\left(1 - \frac{1}{2^n}\right) & \text{if } n \text{ is even.} \end{cases}$$

For the problem in Section 2, we wanted to know the amount of good a in the market at time $t = 100$. Using the general expression above, we can now say that $v_4(100)$ is very close to $\frac{8}{3}$: more precisely, $v_4(100) = \frac{8}{3}\left(1 - \frac{1}{2^{100}}\right)$.

6. From linear algebra to membrane computing

In the previous section we saw that the nth power of a matrix can help us describe the configuration of a rational-valued P system after n transition steps. In this section we will study the reverse question: can we use membrane computing techniques to compute the nth power of the matrix? We will now show that we can give an affirmative answer to this question.

The first step is to obtain a rational-valued P system associated with a given rational-valued square matrix.

Theorem 6.1. Let M be a rational-valued square matrix of dimension $d \times d$. We can construct a rational-valued P system Π such that M is the matrix associated with Π.

Proof. As we pointed out earlier, the matrix associated with a given P system depends only on the skeleton of the P system and not on a specific initial configuration. So, in order to prove the theorem, it is enough to give the alphabet, the set of labels and the rules of a P system. Consider:

— the ordered alphabet $\Gamma = \{a_1, \ldots, a_d\}$;
— the set of labels $H = \{s\}$, that is, the P system has only a membrane, the skin;
— the membrane structure $[\,]_s$.

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

Consider the order $\{(a_1, s), \ldots, (a_d, s)\}$ on $\Gamma \times H$ associated with the total order in Γ, and the set of rules R obtained from the set of algebraic representations $\{(i, \vec{v}_i) \mid 1 \leqslant i \leqslant d\}$ (where $\vec{v}_i$ is the ith row of M) after removing identity rules.

By construction, M is the matrix associated with any rational-valued P system with skeleton (Γ, H, μ, R). □

Given a rational-valued square matrix M of dimension $d \times d$, the P system $\Pi(M)$ constructed from M according to the previous theorem is not unique. Moreover, there can exist two or more such P systems with different skeletons, as the following example shows.

Example 6.1. Consider again the square matrix from Example 4.5:

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

As we saw earlier, it is the associate matrix of the P system

$$\Pi = (\Gamma, H, \mu, w_f, w_s, R)$$

where $\Gamma = \{a, b, c\}$, $H = \{f, s\}$, $\mu = [\,[\,]_f\,]_s$, $w_f = b$, $w_s = b$ and R is the set of rules

Rule 1:	$[a]_f \to a\,[\,]_f$	**Rule 4:**	$a\,[\,]_f \to [a^{1/4}]_f$
Rule 2:	$[b \to a^2 c]_f$	**Rule 5:**	$[b \to a c^4]_s$
Rule 3:	$[c \to a^2 b]_f$	**Rule 6:**	$[c \to b]_s$.

On the other hand, following the construction of the theorem, we also have, for example, the P system

$$\Pi' = (\Gamma, H, \mu, w_s, R)$$

where $\Gamma = \{a_1, \ldots, a_6\}$, $H = \{s\}$, $\mu = [\,]_s$, $w_s = a_1$ and R is the set of rules

Rule 1:	$[a_1 \to a_4]_s$	**Rule 4:**	$[a_4 \to a_1^{1/4}]_s$
Rule 2:	$[a_2 \to a_1^2 a_3]_s$	**Rule 5:**	$[a_5 \to a_4 a_6^4]_s$
Rule 3:	$[a_3 \to a_1^2 a_2]_s$	**Rule 6:**	$[a_6 \to a_5]_s$.

We will now define a concept that is basic for the computation of the power of a matrix using P systems. We have seen that, in general, we can find several skeletons of P systems associated with a given matrix. In order to obtain the computation, we need to consider an initial configuration together with the skeleton of the P system. Given the skeleton of a P system, the next definition shows a set of initial configurations that will be helpful for our purposes.

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

Definition 6.1. Let Γ be the alphabet and H be the set of labels of a given P system, and let $\{\gamma_1,\ldots,\gamma_d\}$ be an order on $\Gamma \times H$. We will say that the set of configurations $\{C_0^1,\ldots,C_0^d\}$ is the *basis* for this order on $\Gamma \times H$ whenever for every $i \in \{1,\ldots,d\}$, the algebraic representation of C_0^i is a vector of dimension d with all its components equal to zero except the ith component, which is equal to one.

Example 6.2. Consider the ordered alphabet $\Gamma = \{a,b,c\}$ and $H = \{f,s\}$ with the order $\{(a,f),(b,f),(c,f),(a,s),(b,s),(c,s)\}$. Then the associated basis is $\mathscr{B} = \{C_0^1,\ldots,\mathscr{C}_0^6\}$ where $C_0^1 = \{(a,f)\}$, $C_0^2 = \{(b,f)\}$, $C_0^3 = \{(c,f)\}$, $C_0^4 = \{(a,s)\}$, $\mathscr{C}_0^5 = \{(b,s)\}$ and $C_0^6 = \{(c,s)\}$.

Finally, the next theorem provides a method for computing the power of a rational-valued square matrix using membrane computing techniques.

Theorem 6.2. Let M be a rational-valued square matrix of dimension d. Let (Γ,H,μ,R) be the skeleton of a P system Π such that M is the matrix associated with Π. Let $\{\gamma_1,\ldots,\gamma_d\}$ be an order on $\Gamma \times H$ and let $\mathscr{B} = \{C_0^1,\ldots,C_0^d\}$ be the basis for this order on $\Gamma \times H$. Consider $n \in \mathbb{N}$ and for every i $(1 \leqslant i \leqslant d)$ let C_n^i be the configuration obtained in the P system with skeleton (Γ,H,μ,R) and initial configuration C_0^i after n steps of transitions. Then the nth power of M is the matrix having $\vec{C}_n^1,\ldots,\vec{C}_n^d$ as rows, where $\vec{C}_n^i$ is the algebraic representation of C_n^i for every i $(1 \leqslant i \leqslant d)$.

Proof. We must prove that $\vec{C}_n^i = (M^n)_i$ for every i $(1 \leqslant i \leqslant d)$, where $(M^n)_i$ is the ith row of M^n.

From Theorem 4.1 we know that for any configuration $\mathscr{C}_0$, the next configuration C_1 is the configuration associated with the algebraic configuration $\vec{C}_0 \cdot M$, and then the nth configuration obtained from C_0 is the configuration associated with the algebraic configuration $\vec{C}_0 \cdot M^n$.

This holds for any configuration, and thus, in particular, for the element of the basis. Therefore, for each $i \in \{1,\ldots,d\}$,

$$\vec{C}_n^i = \vec{C}_0^i \cdot M^n.$$

However, $\vec{C}_0^i \cdot M^n$ is the ith row of M^n, since $\vec{C}_0^i$ is a vector with all its components equal to zero, apart from the ith component, which is equal to 1. □

Corollary 6.1. P systems can be used to compute the n-power of a $d \times d$ matrix in time $\Theta(n \cdot d)$.

Example 6.3. Consider the rational-valued square matrix of dimension 3×3

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

and consider the skeleton (Γ,H,μ,R) where $\Gamma = \{a,b,c\}$, $H = \{s\}$, $\mu = [\,]_s$, and R is the set of rules

$$\mathbf{R}_1 \equiv [a \to a^2c]_s \quad \mathbf{R}_2 \equiv [b \to ab]_s \quad \mathbf{R}_3 \equiv [c \to a^3bc]_s.$$

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

Consider the order $((a,s),(b,s),(c,s))$ and the basis $\mathscr{B} = \{C_0^1, C_0^2, C_0^3\}$ with $C_0^1 = \{(a,s)\}$, $C_0^2 = \{(b,s)\}$ and $\mathscr{C}_0^3 = \{(c,s)\}$. The first three steps of the computation, starting with the elements of the basis, are as follows:

$$\begin{array}{llll} C_0^1 = [a]_s & C_1^1 = [a^2c]_s & \mathscr{C}_2^1 = [a^7bc^3]_s & C_3^1 = [a^{24}b^4c^{10}]_s \\ C_0^2 = [b]_s & C_1^2 = [ab]_s & \mathscr{C}_2^2 = [a^3bc]_s & C_3^2 = [a^{10}b^2c^4]_s \\ C_0^3 = [c]_s & C_1^3 = [a^3bc]_s & \mathscr{C}_2^3 = [a^{10}b^2c^4]_s & C_3^3 = [a^{34}b^6c^{14}]_s. \end{array}$$

Since $\vec{C}_3^1 = (24,4,10)$, $\vec{C}_3^2 = (10,2,4)$ and $\vec{C}_3^3 = (34,6,14)$, according to Theorem 6.2, we have

$$M^3 = \begin{pmatrix} 24 & 4 & 10 \\ 10 & 2 & 4 \\ 34 & 6 & 14 \end{pmatrix}$$

7. Conclusions and future work

In the early years of membrane computing research, a big effort was made to prove the computational completeness and computational efficiency of the different models of P systems. Recently, a wide research line has opened up looking for solutions to real-world problems. Membrane computing provides a framework flexible enough to adapt to many different problems, as shown in Ciobanu *et al.* (2006), and this increasing extension to other scientific areas requires ever more efficient techniques.

In this paper we have presented a new point of view on the representation of P systems. The consideration of rational-valued P systems is not new, and the representation of configurations and rules using object–label pairs has been the basis for other successful results (see Gutiérrez-Naranjo *et al.* (2006) or Gutiérrez-Naranjo *et al.* (2005)). The main novelty in the current paper is the use of a total order on the set of object–label pairs. This order has also allowed us to represent the multiplicities of the objects in the different membranes as a vector on the right-hand side of the rules. We have shown that the *associated matrix* is a powerful tool for computing the configuration after n transition steps and, being a rational-valued matrix, we can operate on it with all the well-known linear algebra techniques. In particular, the Jordan matrix computation can be applied to solve real-world problems described in terms of membrane computing.

The use of this new representation opens up a wide range of possibilities by translating linear algebra results into P systems. From the membrane computing point of view, the question is whether this representation can be extended to other P system models, and, in particular, to non-deterministic models and/or to models where the membrane structure can change during the computation.

Furthermore, we have shown that membrane computing techniques can be used to compute the nth power of a given matrix. The current state of P system simulators cannot compete with dedicated software for computation with matrices, but if the research leads us to an implementation of P systems (*in vivo*, *in vitro* or *in electronic media*), our results provide algorithms for a realistic computation of the powers of matrices using membrane computing devices.

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press

Acknowledgement

The authors acknowledge the support of both the project TIN2006-13425 of the Ministerio de Educación y Ciencia of Spain, co-financed by FEDER funds, and the project of excellence TIC-581 of the Junta de Andalucía.

References

Ciobanu, G., Păun, Gh. and Pérez-Jiménez, M. J. (eds.) (2006) *Applications of Membrane Computing*, Springer-Verlag.

Coppersmith, D. and Winograd, S. (1990) Matrix multiplication via arithmetic progressions. *Journal of Synbolic Computation* **9** 251–280.

Cordón-Franco, A., Gutiérrez-Naranjo, M. A., Pérez-Jiménez, M. J. and Riscos-Núñez, A. (2005) Exploring Computation Trees Associated with P Systems. In: Membrane Computing. *Springer-Verlag Lecture Notes in Computer Science* **3365** 278–286.

Gutiérrez-Naranjo, M. A., Pérez-Jiménez, M. J., Riscos-Núñez, A. and Romero-Campero, F. J. (2005) P Systems with Active Membranes, without Polarizations and with Dissolution: A Characterization of P. In: Unconventional Computation. *Springer-Verlag Lecture Notes in Computer Science* **3699** 105–116.

Gutiérrez-Naranjo, M. A., Pérez-Jiménez, M. J., Riscos-Núñez, A. and Romero-Campero, F. J. (2006) On the Power of Dissolution in P Systems with Active Membranes. In: Membrane Computing. *Springer-Verlag Lecture Notes in Computer Science* **3850** 224–240.

Horn, R. A. and Johnson, C. R. (1985) *Matrix Analysis*, Cambridge University Press.

Păun, Gh. (2000) Computing with membranes. *Journal of Computer and System Sciences* **61** (1) 108–143.

https://doi.org/10.1017/S0960129509990144 Published online by Cambridge University Press