Computing Backwards with P Systems

Miguel A. Gutiérrez-Naranjo, Mario J. Pérez-Jiménez

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

Summary. Searching all the configurations C' such that produce a given configuration C, or, in other words, computing backwards in Membrane Computing is an extremely hard task. The current approximations are based in heavy hand-made calculus by considering the specific features of the given configuration. In this paper we present a general method for characterizing all the configurations C' such that produce a given configuration C in transition P systems without cooperation and without dissolution.

1 Introduction

Given a computational model with a universal clock, where the time is considered in a discrete way and the transition from a state to the next one is made by a set of rules, it is usual to wonder about the previous state of a given one, or in other words, to wonder about the possibility of computing backwards.

Note that the determinism of the model does not make the solution easier, since the determinism of the computation does not lead to the determinism of the reverse computation. One can go deterministically from S to S_0 and from S' to S_0, but given S_0, the reversed computation is not deterministic. A special situation is considered when the rules are *reversible*. In this case, it suffices to apply the reversed rules to S_1 according to the computational model to obtain the desired states[1].

In this paper we study the problem of characterizing the set of configurations of a P systems that produce a given configuration in one computational step. We study the case in which the P system is not necessarily deterministic and the rules are not reversible in general. We will consider a restricted version of transition P systems without cooperation where the membrane structure does not change along the computation.

The paper is organized as follows: first we expose an example that shows the necessity of finding a method for computing backwards, avoiding the heavy calculus

[1] This case is studied for P systems in [1].

based on specific features of the given configuration. Next, our P system model is briefly introduced and we present our representation for configurations and rules in such a P system. In Section 6 we prove our main result: The computation of all the configurations C' such that produce a given configuration C can be reduced to find solutions of a system of linear equations with values[2] in $\mathbb{N}$. In Section 7 we provide a general method of calculus based on our theorem. Finally, some conclusions and new open research lines are presented.

2 Motivation

The reader is assumed to be familiar with basic elements of membrane computing, e.g., from [4] Let us start with the P system Π with alphabet $\Gamma = \{a, b, c\}$, set of labels $H = \{e, s\}$, membrane structure $\mu = [\,[\,]_e\,]_s$ and set of rules R

Rule 1: $[\,a \to b^2c\,]_e$	**Rule 4:** $[\,b \to a\,]_s$
Rule 2: $[\,a\,]_e \to a\,[\,]_e$	**Rule 5:** $a\,[\,]_e \to [\,c\,]_e$
Rule 3: $[\,b \to c^2\,]_s$	**Rule 6:** $[\,c \to a\,]_e$

In Section 3, we will give a detailed description of the P system model studied in this paper, but by now it is enough to know that all the rules are applied in a non-deterministic maximal parallel way as usual in the general framework of Membrane Computing.

Let us consider now configuration $C' = [\,[a^2b]_e\,a^2c\,]_s$, i.e., the configuration in which the multiset placed in the membrane labeled by e is a^2b and the multiset in the membrane s is a^2c. Our problem is to find the configuration (or configurations) C such that we can go from C to C' in *one* computational step. In other words, we want to compute backwards from C and characterize *all* the configurations C such that produce C' in one computation step.

We can reason in the following way:

- We find two objects a in the membrane labeled by e in the configuration C'. Since rules 1 and 2 consume all the objects in the membrane e from the previous configuration C, we conclude that such pair of objects a must be produced by the application of rule(s) of Π. It is easy to check that only rule 6 produces objects a in membrane e, then the number of objects c in configuration C must be at least 2. If we look at the set of rules again, we observe that object c in membrane e only triggers rule 6. Hence, if the number of objects c in e is higher than 2 we conclude that the number of objects a in the membrane e in the configuration C must be greater than 2. Therefore, we conclude that the number of objects c in the membrane e in configuration C is equal to 2.
- We find one object b in the membrane labeled by e in configuration C'. The unique rule that can produce it is rule 1, but the application of the rule produces at least two objects b in membrane e. Then we conclude that rule 1 is not

[2] We represent by $\mathbb{N} = \{0, 1, 2, \dots\}$ the set of natural numbers.

applied. The occurrence of such object b can only be explained by considering its occurrence in configuration C. As one can check, no rule is triggered by object b in the membrane e, then the number of objects b in membrane e in the configuration C equals to 1.

- No object c are placed in the membrane e in C'. All such objects from the previous configuration C are consumed by rule 6, so no object c in the membrane e imply that rules 1 and 5 have not been triggered. From the previous paragraph, it is known that rule 5 has not been applied. Since all the objects a in membrane s send objects e into membrane c by means of rule 5 and the numbers of objects c in such membrane in configuration C' is zero, we conclude that in configuration C no objects a are placed in the membrane s.
- We find one object c in the membrane labeled by s in configuration C'. The unique rule that can produce it is rule 3, but the application of the rule produces at least two objects c in membrane s. Then we conclude that rule 3 is not applied. The occurrence of such object b can only be explained by considering its occurrence in configuration C. As one can check, no rule is triggered by the object c in the membrane s, then the number of objects c in membrane s in the configuration C equals 1.
- Finally, we find two objects a in the membrane labeled by s in the configuration C'. Since rule 5 consumes all the objects in the membrane e from the previous configuration C, we conclude that such objects a must be produced by the application of rule(s) of Π. Rules 2 and 4 produce objects a in membrane s. Rule 2 is triggered by an object a in the membrane e and rule 4 is triggered by an object b in membrane s. We can also check that all the objects b in s produce objects a. Nonetheless, an object a in the membrane e can trigger rules 1 and 2. Fortunately, we have seen that rule 1 is not triggered, so can conclude that all the objects a in membrane e trigger rule 2. We conclude that the number of objects a in membrane e in the configuration C and the number of objects b in the membrane s must be less than or equal to 2 and the sum of both numbers must be equal to 2.

Bearing in mind these considerations, there are three configurations C such that produce C' in one computation step:

- $C_1 = [\,[\,bc^2\,]_e\, b^2c\,]_s$, i.e., $w_e = bc^2$ and $w_s = b^2c$. It is easy to check that by applying the rules 4 and 6 we obtain the configuration $C' = [\,[a^2b\,]_e\, a^2c\,]_s$.
- $C_2 = [\,[\,abc^2\,]_e\, bc\,]_s$, i.e., $w_e = abc^2$ and $w_s = bc$. In this case, C' is obtained by applying the rules 2, 4 and 6.
- $C_3 = [\,[\,a^2bc^2\,]_e\, c\,]_s$, i.e., $w_e = a^2bc^2$ and $w_s = c$. In this case, C' is obtained by applying the rules 2 and 6.

A question arises in a natural way: Could this reasoning be automatic? In other words, given a P system and a configuration C', is there an algorithm such that outputs the set $\mathcal{C}$ of configurations C and produce C' in one computational step?

We can even go beyond. We wonder if there exists an algorithm such that it takes a P system Π as input and it outputs a mapping $\mathcal{R}_\Pi$ which, for *every*

configuration C' of Π, $\mathcal{R}_\Pi(C')$ is the set of all computations C such that C' is obtained from C in one computational step. In this paper, we will give a positive answer to both questions. Before, we need to recall the connections between P systems and Linear Algebra.

3 The P System Model

Throughout this paper, we will consider a restricted form of transition P systems without dissolution and without output membrane. Considering an output membrane is irrelevant for our study, since we are not interested in the objects placed in a particular membrane, but in the computation process itself. We also restrict the type of rules. Cooperation is not allowed and then rules are triggered by only one object.

Namely, along this paper a P system of degree m is a tuple

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

where:

- Γ is an alphabet whose elements are called *objects*;
- H is the set of m labels and m is called the *degree* of Π.
- μ is the membrane structure of the P system; membranes are bijectively labelled with the elements of H;
- $w_{h_1}, \ldots, w_{h_m}$ are strings that represent multisets over Γ associated with each membrane of μ;
- $R = \{R_1, \ldots, R_m\}$ is the set of sets of rules, where R_i with $i \in \{1, \ldots, m\}$ is a finite set of *evolution rules* over Γ. The type of evolution rules of R_i depends on the membrane structure μ. Let $j_1, \ldots, j_r$ be the labels of membranes immediately inside the membrane i. An evolution rule of R_i is of the form $a \to v$, where a is an object from Γ and v is an string over Γ^i_{tar}, where $\Gamma^i_{tar} = \Gamma \times TAR_i$, for $TAR_i = \{here, out\} \cup \{in_{j_k} \mid k \in \{1, \ldots, r\}\}$.

The symbols *here*, *out* and in_{j_k} are called *target commands*. The rules are applied in a non-deterministic maximally parallel way. Given a rule $a \to v$, the effect of applying this rule in a compartment i is to remove the object a and to insert the objects specified by v in the regions designated by the target commands associated with the objects from v. In particular,

- if v contains $(a, here)$, the object a will be placed in the same region where the rule is applied;
- if v contains (a, out), the object a will be placed in the compartment that surrounds the region where the rule is applied;
- if v contains (a, in_j), the object a will be placed in compartment j, provided that j is immediately inside i.

In one step, each object in a membrane can only be used for one rule (non deterministically chosen when there are several possibilities), but any object which can evolve by a rule of any form must do it. All the elements which are not involved in any of the rules to be applied remain unchanged. Several rules can be applied to different objects in the same cell simultaneously.

Along 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 constant. We will call the 4-uple (Γ, H, μ, R) the *skeleton* of the P system.

Notice that the P system presented in Section 2 is a particular case of this P system model with a slight change of notation in the rules

1. Notation $[a \to v]_h$ where $h \in H$, $a \in \Gamma$ and v is a string over Γ is a short notation to indicate that the rule $a \to (v_1, here) \dots (v_n, here)$ belongs to the set of rules R_h, with $v = v_1 \dots v_n$.
2. Notation $a[\,]_h \to [v]_h$ where $h \in H$, $a \in \Gamma$ and v is a string over Γ is a short notation to indicate that the rule $a \to (v_1, in_h) \dots (v_n, in_h)$ belongs to the set of rules R_{h^*}, with h^* the label of the membrane surrounding the membrane h and $v = v_1 \dots v_n$.
3. Notation $[a]_h \to v[\,]_h$ where $h \in H$, $a \in \Gamma$ and v is a string over Γ is a short notation to indicate that the rule $a \to (v_1, out) \dots (v_n, out)$ belongs to the set of rules R_h, with $v = v_1 \dots v_n$.

4 Changing the Point of View

The key idea of the present paper is to consider an algebraic representation for the configurations and the rules of a P system. The starting point is the representation used in [2], but we introduce several changes.

First, our elementary objects are pairs of type $(a, h) \in \Gamma \times H$ meaning that object $a \in \Gamma$ is placed in the membrane (labeled 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 element a_1 in membrane h_1, β_2 copies of element a_2 in membrane h_2, etc.

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

$$(a_0, h_0) \to (a_1, h_1)^{\beta_1} (a_2, h_2)^{\beta_2} \dots (a_n, h_n)^{\beta_n}$$

Note that, for all $i \in \{1, \dots, n\}$, if $h_0 = h_i$ then, (a_i, h_i) is equivalent to the pair $(a_i, here)$. Otherwise, if $h_0 \neq h_i$ both membranes must be adjacent (one membrane is the father of the other one). If h_0 is the father of h_i, then the pair (a_i, h_i) is equivalent to (a_i, in_{h_i}). Finally, if h_i is the father of h_0, then the pair (a_i, h_i) is equivalent to (a_i, out). For each $i \in \{1, \dots, n\}$, β_i represents the multiplicity of (a_i, h_i) in the right-hand side (RHS) of the rule.

The second basic idea in the representation appears in [3] too. It consists on settling a total order in the set $\Gamma \times H$. Along the paper, in order to simplify the notation, given an alphabet Γ and a set of labels H, d will denote the cardinal $\Gamma \times H$. Let us consider a total order $\mathcal{O}$ on the set $\Gamma \times H$, $\mathcal{O} : \{1, \ldots, d\} \to \Gamma \times H$. By using this order, we will represent $\Gamma \times H$ as the finite sequence $\langle \gamma_1, \ldots, \gamma_d \rangle$, where γ_i is the i-th pair of $\Gamma \times H$ in the order $\mathcal{O}$.

By using this order, each rule

$$(a_0, h_0) \to (a_1, h_1)^{\beta_1} (a_2, h_2)^{\beta_2} \ldots (a_n, h_n)^{\beta_n}$$

can be represented as

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

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

- If there exists $j \in \{1, \ldots, n\}$ such that $\gamma_i = (a_j, h_j)$ then $\alpha_i = \beta_j$.
- Otherwise $\alpha_i = 0$.

We will say that $\gamma \to \gamma_1^{\alpha_1} \gamma_2^{\alpha_2} \ldots \gamma_d^{\alpha_d}$ is the *pairwise* representation of the rule.

The use of an order on $\Gamma \times H$ leads us to a more homogeneous representation of rule $\gamma \to \gamma_1^{\alpha_1} \gamma_2^{\alpha_2} \ldots \gamma_d^{\alpha_d}$. It can be represented by a pair $\langle \gamma, \vec{v} \rangle$ where γ (the LHS of the rule) belongs to $\Gamma \times H$, and $\vec{v}$ is a vector of dimension d whose arguments are in $\mathbb{N}$. Formally, we have the following definition:

Definition 1. *Let us consider a P system Π with Γ the alphabet and H the set of labels. Let $\Gamma \times H$ be the ordered set $\langle \gamma_1, \ldots, \gamma_d \rangle$. The algebraic representation of the rule*

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

is the pair $(\gamma, \vec{v})$ where $\vec{v} = (\alpha_1, \ldots, \alpha_d)$. We will say that $\vec{v}$ represents the right-hand side of the rule r_i.

Remark 1: Given an order $\langle \gamma_1, \ldots, \gamma_d \rangle$ on $\Gamma \times H$, a pair $\langle \gamma, \vec{v} \rangle$ where $\gamma \in \Gamma \times H$ and $\vec{v}$ is a vector of dimension d (with values in $\mathbb{N}$) defines a unique rule and vice-versa, each rule having a unique algebraic representation.

Remark 2: If the P system is not deterministic, then there exists at least one $\gamma \in \Gamma \times H$ such that there exists two different vectors $\vec{v}_1$ and $\vec{v}_2$ such that pairs $\langle \gamma, \vec{v}_1 \rangle$ and $\langle \gamma, \vec{v}_2 \rangle$ represent two different rules.

Let us see an example of this algebraic representation.

Example 1. Let us consider the skeleton of the P system considered in Section 2 with $\Gamma = \{a, b, c\}$, $H = \{e, s\}$, $\mu = [\,[\,]_e\,]_s$ and R the set of rules

Rule 1: $[\,a \to b^2 c\,]_e$	**Rule 4:** $[\,b \to a\,]_s$
Rule 2: $[\,a\,]_e \to a\,[\,]_e$	**Rule 5:** $a\,[\,]_e \to [\,c\,]_e$
Rule 3: $[\,b \to c^2\,]_s$	**Rule 6:** $[\,c \to a\,]_e$

The set of objects is $\Gamma = \{a, b, c\}$ and the set of labels is $H = \{e, s\}$. Let us consider the following total order in $\Gamma \times H$

$$\langle (a,e),(b,e),(c,e),(a,s),(b,s),(c,s)\rangle$$

The six rules of the P system can be settled as

$$\begin{array}{ll} r_1\colon (a,e)\to (b,e)^2(c,e) & r_4\colon (b,s)\to (a,s)\\ r_2\colon (a,e)\to (a,s) & r_5\colon (a,s)\to (c,e)\\ r_3\colon (b,s)\to (c,s)^2 & r_6\colon (c,e)\to (a,e)\end{array}$$

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

$$\begin{array}{ll} \textbf{Rule 1: } \langle (a,e),(0,2,1,0,0,0)\rangle & \textbf{Rule 4: } \langle (b,s),(0,0,0,1,0,0)\rangle\\ \textbf{Rule 2: } \langle (a,e),(0,0,0,1,0,0)\rangle & \textbf{Rule 5: } \langle (a,s),(0,0,1,0,0,0)\rangle\\ \textbf{Rule 3: } \langle (b,s),(0,0,0,0,0,2)\rangle & \textbf{Rule 6: } \langle (c,e),(1,0,0,0,0,0)\rangle\end{array}$$

4.1 Configurations

A *configuration* of such a P system is the description of the multiset placed in the membranes of the P system in a given instant. Formally, given a P system with working alphabet Γ and set of labels H, a configuration C is a multiset over $\Gamma\times H$, $C:\Gamma\times H\to\mathbb{N}$, and we denote by $C(a,m)$ the multiplicity of object a in the membrane labeled by m of that configuration. The support of C, $supp(C)$, is defined as $supp(C)=\{(a,m)\in\Gamma\times H\,|\,C(a,m)\neq 0\}$ and, as usual in multisets theory, C will be represented as $\{(a,m)^{C(a,m)}\,|\,(a,m)\in supp(\mathcal{C})\}$. For example, the configuration of our example $[\,[\,b\,]_e\,c^3\,]_s$ can be represented as $\{(b,e),(c,s)^3\}$.

From the idea of setting an order on $\Gamma\times H$, the representation of a configuration via a vector is quite natural.

Definition 2. *Let us consider a P system Π with Γ the alphabet, H the set of labels and order $\langle\gamma_1,\ldots,\gamma_d\rangle$ on $\Gamma\times H$. An algebraic representation of a configuration $C:\Gamma\times H\to\mathbb{N}$ is a vector*

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

that is, the j-th element in $\vec{C}$ is a number representing the multiplicity of the j-th element of $\Gamma\times H$.

Let us remark that, if the order on $\Gamma\times H$ is set, then there exists a bijective correspondence between a configuration C and its algebraic representation $\vec{C}$.

Example 2. As we saw before, the initial configuration $[\,[\,b\,]_e\,c^3\,]_s$ can be expressed as the multiset $C=\{(b,e),(c,s)^3\}$. If we consider order

$$\langle (a,e),(b,e),(c,e),(a,s),(b,s),(c,s)\rangle$$

then the algebraic representation of the configuration is $\vec{C}=(0,1,0,0,0,3)$.

In order to formalize the concept of computation with this new representation, we will fix some notations. We denote by RHS_r the right-hand side of rule r and for all $\sigma \in \Gamma \times H$, $|RHS_r(\sigma)|$ denotes the multiplicity of σ in the multiset RHS_r.

Example 3. Let us consider the pairwise representation of the rule $r_1 : (a,e) \to (b,e)^2(c,e)$, then $RHS_{r_1} = (b,e)^2(c,2)$ and $|RHS_{r_1}|(b,e) = 2$.

Definition 3. *Let us consider an alphabet Γ, a set of labels H and the set of rules R of a P system. We will denote by $\mathcal{LHS}(R)$ the set of all the pairs from $\Gamma \times H$ that are the left-hand side of a rule from R. Formally*

$$\mathcal{LHS}(R) = \{\gamma \in \Gamma \times H \mid \exists r \in R\,(\gamma = LHS(r))\}$$

Example 4. Let us consider $\Gamma = \{a,b,c\}$, $H = \{e,s\}$ and R the set of rules

$$\begin{array}{lll} r_1\colon (a,e) \to (c,e)^2 & r_2\colon (a,e) \to (a,s) & r_3\colon (b,e) \to (c,e) \\ r_4\colon (a,s) \to (b,s) & r_5\colon (a,s) \to (b,s)(c,s)^2 & \end{array}$$

In this case $\mathcal{LHS}(R) = \{(a,e),(b,e),(a,s)\}$.

Definition 4. *Let us consider an alphabet Γ and a set of labels H of a P system Π and let $R = \langle r_1,\ldots,r_p\rangle$ be an enumeration of its set of rules with $r_j = (LHS(r_j),\vec{v_j})$. Let $C : \Gamma \times H \to \mathbb{N}$ be a configuration of Π.*

A partition *of C with respect to R is a p-uple*

$$\mathcal{P} = \langle (r_1,k_1),\ldots,(r_p,k_p)\rangle$$

such that for all $j \in \{1,\ldots,p\}$, $k_j \geq 0$ and for all $\gamma \in \mathcal{LHS}(\mathcal{R})$

$$\sum_{LHS(r_j)=\gamma} k_j = C(\gamma)$$

Example 5. Let us consider an alphabet $\Gamma = \{a,b,c\}$ a set of labels $H = \{e,s\}$, $\mu = [\,[\,]_e\,]_s$ and R the set of rules from the example 4

$$\begin{array}{lll} r_1\colon (a,e) \to (c,e)^2 & r_2\colon (a,e) \to (a,s) & r_3\colon (b,e) \to (c,e) \\ r_4\colon (a,s) \to (b,s) & r_5\colon (a,s) \to (b,s)(c,s)^2 & \end{array}$$

Let us consider a configuration with algebraic representation $\vec{C} = \langle 3,0,1,7,4,1\rangle$ associated with order $\langle (a,e),(b,e),(c,e),(a,s),(b,s),(c,s)\rangle$ of $\Gamma \times H$. In this case, one possible partition of C with respect to R is

$$\mathcal{P} = \langle (r_1,2),(r_2,1),(r_3,0),(r_4,2),(r_5,5)\rangle$$

the number associated to each rule is a natural number and $\mathcal{LHS}(R) = \{(a,e),(b,e),(a,s)\}$, so in order to check that $\mathcal{P}$ is a partition it suffices to check

$$\begin{array}{l} \sum_{LHS(r_j)=(a,e)} k_j = k_1 + k_2 = 2 + 1 = 3 = C(a,e) \\ \sum_{LHS(r_j)=(b,e)} k_j = k_3 = 0 = C(b,e) \\ \sum_{LHS(r_j)=(a,s)} k_j = k_4 + k_5 = 2 + 5 = 7 = C(a,s) \end{array}$$

The different possible partitions capture the idea of different choice of rules in the case of non-deterministic P system. Notice that in the case of a deterministic P system, there exists only one partition

$$\mathcal{P} = \langle (r_1, C(LHS(r_1))), (r_2, C(LHS(r_2))), \ldots, (r_p, C(LHS(r_p))) \rangle$$

In order to obtain a new configuration C' from a given configuration C and from the set of rules $\{r_1, \ldots, r_p\}$, we need to describe the multiplicity of any $\sigma \in \Gamma \times H$ in C'. For the calculus of such multiplicity we need

- A partition $\mathcal{P} = \langle (r_1, k_1), \ldots, (r_p, k_p) \rangle$ of C with respect to R.
- The set $\mathcal{LHS}(R)$

In such multiplicity, each rule $r_i : \gamma_i \to RHS_{r_i}$ adds the multiplicity of σ in the right hand side of the rule multiplied by the value k_i in the partition $\mathcal{P}$. If the object is not consumed by any rule, we also add the multiplicity in the original configuration.

Formally, for every $\sigma \in \Gamma \times H$ we have:

$$C'(\sigma) = \begin{cases} \sum_{i=1}^{i=p} k_i \cdot |RHS_{r_i}(\sigma)| & \text{if } \sigma \in \mathcal{LHS}(R) \\ \sum_{i=1}^{i=p} k_i \cdot |RHS_{r_i}(\sigma)| + C(\sigma) & \text{if } \sigma \notin \mathcal{LHS}(R) \end{cases}$$

Example 6. Let us come back again to our P system Π with alphabet $\Gamma = \{a, b, c\}$, set of labels $H = \{e, s\}$, membrane structure $\mu = [\,[\,]_e\,]_s$ and the set of rules R

Rule 1: $[\,a \to b^2c\,]_e$	**Rule 4:** $[\,b \to a\,]_s$
Rule 2: $[\,a\,]_e \to a\,[\,]_e$	**Rule 5:** $a\,[\,]_e \to [\,c\,]_e$
Rule 3: $[\,b \to c^2\,]_s$	**Rule 6:** $[\,c \to a\,]_e$

Let us consider configuration $C_1 = [\,[\,bc^2\,]_e\, b^2c\,]_s$, i.e., $w_e = bc^2$ and $w_s = b^2c$. It is easy to check that by applying rules 4 and 6 we obtain configuration $C' = [\,[a^2b]_e\, a^2c\,]_s$. Such configuration can also be obtained by considering the multiplicity of each pair in $\Gamma \times H$ and using the previous formula. First we consider the partition $\mathcal{P} = \langle (r_1, 0), (r_2, 0), (r_3, 0), (r_4, 2), (r_5, 0), (r_6, 2) \rangle$ and $\mathcal{LHS}(R) = \{(a, e), (b, s), (a, s), (c, e)\}$. Then, for example,

$$C'(a, s) = k_1 \cdot 0 + k_2 \cdot 1 + k_3 \cdot 0 + k_4 \cdot 1 + k_5 \cdot 0 + k_6 \cdot 0 = 2 \cdot 1 = 2$$
$$C'(b, e) = k_1 \cdot 2 + k_2 \cdot 0 + k_3 \cdot 0 + k_4 \cdot 0 + k_5 \cdot 0 + k_6 \cdot 0 + C(b, e) = 0 \cdot 2 + 1 = 1$$

the remaining multiplicities in configuration C' can be obtained in a similar way.

5 Matrix Associated with the Skeleton

After defining the algebraic representation of rules and configurations, we will define a numerical matrix associated with the skeleton of a P system. The next definition of *extended* set of rules will be used in the definition of the matrix.

Definition 5. *Let Γ be the alphabet, H the set of labels and R the set of rules of a P system where R is a set of rules in its pairwise form. The extended set of rules of R in this skeleton, R^* is the set of rules R together with the identity rule $\gamma \to \gamma$ for all the $\gamma \in \Gamma \times H$ such that there is no rule in R with γ in its left-hand side.*

Considering identity rules, we obtain P systems whose computations never stop. In this paper, we are interested only in the evolution of computation in time and not in halting conditions. Let us remark two important considerations related with the extended set of rules:

- If R^* is the extended set of rules of R, then $\mathcal{LHS}(R^*) = \Gamma \times H$.
- Consequently, if C is a configuration of a P system Π with $\langle \gamma_1, \ldots, \gamma_d \rangle$ an order on $\Gamma \times H$ and $\mathcal{P}^* = \langle (r_1, k_1), \ldots, (r_p, k_p) \rangle$ is a partition of a configuration C of a P system with respect to its extended set of rules, then configuration C' that can be obtained from C in one computation step following such partition is $C'(\gamma_j) = \sum_{i=1}^{i=p} k_i \cdot |RHS_{r_i}(\gamma_j)|$ for all $j \in \{1, \ldots, d\}$.

Example 7. Let us consider again the skeleton of example 1, and its set of rules,

$$\begin{array}{ll} r_1\colon (a,e) \to (b,e)^2(c,e) & r_4\colon (b,s) \to (a,s) \\ r_2\colon (a,e) \to (a,s) & r_5\colon (a,s) \to (c,e) \\ r_3\colon (b,s) \to (c,s)^2 & r_6\colon (c,e) \to (a,e) \end{array}$$

Note that the pairs γ from $\Gamma \times H$ such that there is no rule in R with γ as its left-hand side are (b,e) and (c,s), therefore to obtain R^* we have to add to R the rules

$$r_7\colon (b,e) \to (b,e) \qquad r_8\colon (c,s) \to (c,s)$$

Obviously, the set of rules R^* has also an algebraic representation

$$\begin{array}{ll} \textbf{Rule 1:} \langle (a,e), (0,2,1,0,0,0) \rangle & \textbf{Rule 5:} \langle (a,s), (0,0,1,0,0,0) \rangle \\ \textbf{Rule 2:} \langle (a,e), (0,0,0,1,0,0) \rangle & \textbf{Rule 6:} \langle (c,e), (1,0,0,0,0,0) \rangle \\ \textbf{Rule 3:} \langle (b,s), (0,0,0,0,0,2) \rangle & \textbf{Rule 7:} \langle (b,e), (0,1,0,0,0,0) \rangle \\ \textbf{Rule 4:} \langle (b,s), (0,0,0,1,0,0) \rangle & \textbf{Rule 8:} \langle (c,s), (0,0,0,0,0,1) \rangle \end{array}$$

With the help of the concept of extended set of rules, we define the matrix associated with a skeleton.

Definition 6. *Let us consider skeleton $Sk = (\Gamma, H, \mu, R)$ of a P system and let $\langle r_1, \ldots, r_p \rangle$ be an enumeration of the extended set of rules R^* of R in its algebraic form. The matrix associated with skeleton Sk, M_{Sk} is the matrix whose rows are vectors $\vec{v_1}, \ldots, \vec{v_p}$, where for each i with $1 \leq i \leq p$, $\vec{v_i}$ is the vector which represents the right-hand side of rule r_i.*

Before showing an example, some remarks are necessary.

- The matrix associated with a skeleton depends on the skeleton, as well as on the enumeration of the rules of the extended set and the order on $\Gamma \times H$. A different enumeration produces a different order in the rows of the matrix.
- In case of deterministic P systems, the number of rules in the extended set, p, and the number of pairs in $\Gamma \times H$, d are the same and we have a square matrix[3]. In general, M_{Sk} is a $d \times p$ matrix with $d \leq p$.

Example 8. If we consider the skeleton of example 7 and the enumeration of the eight rules of the extended set R^* and the usual order on $\Gamma \times H$, $\langle (a,e),(b,e),(c,e),(a,s),(b,s),(c,s)\rangle$

Rule 1: $\langle (a,e),(0,2,1,0,0,0)\rangle$ **Rule 5:** $\langle (a,s),(0,0,1,0,0,0)\rangle$
Rule 2: $\langle (a,e),(0,0,0,1,0,0)\rangle$ **Rule 6:** $\langle (c,e),(1,0,0,0,0,0)\rangle$
Rule 3: $\langle (b,s),(0,0,0,0,0,2)\rangle$ **Rule 7:** $\langle (b,e),(0,1,0,0,0,0)\rangle$
Rule 4: $\langle (b,s),(0,0,0,1,0,0)\rangle$ **Rule 8:** $\langle (c,s),(0,0,0,0,0,1)\rangle$

we have the following matrix

$$M_{Sk} = \begin{pmatrix} 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

6 Computing Backwards

The definition of these algebraic objects allows us to define an algebraic method to characterize the set of configurations $\mathcal{C}$ which can produce a given configuration C_0 in one computation step. First, we need to find the solutions of a system of linear equations.

Definition 7. *Let Π be a P system, $\langle r_1,\ldots,r_p\rangle$ a enumeration of its set of extended rules, M_{Sk} the matrix associated with the skeleton of Π based on that enumeration of R^* and let $\vec{C_0}$ be the vectorial representation of a configuration C_0. We will define the* solution set *of M_{Sk} and $\vec{C_0}$ and we will denote it by $SOL(M_{Sk},\vec{C_0})$ the set of real-valued vectors $\vec{x}$ with dimension p such that $\vec{C_0} = \vec{x}\cdot M_{Sk}$.*

Notice that according to the definition, $SOL(M_{Sk},\vec{C_0})$ can be the empty set. It is well known in Linear Algebra that if the range of the matrix M_{Sk} and the

[3] This kind of matrices were studied in [3].

range of the matrix M_{Sk} augmented with the vector of coefficients $\vec{C_0}$ is not the same, then the system of equations has no solution.[4]

$SOL(M_{Sk}, \vec{C_0})$ is a manifold of dimension p minus the range of the matrix M_{Sk} embedded in a vectorial space of dimension p, but the study of the algebraic properties of such manifold is out of the scope of this paper.

Example 9. Let us come back to our main example. If we take the matrix M_{Sk} from example 8, configuration $C' = [\,[\,a^2b\,]_e\, a^2c\,]_s$ from Section 2 and algebraic representation $\vec{C'} = (2,1,0,2,0,1)$, then in order to get $SOL(M_{Sk}, \vec{C'})$ we need to solve the system

$$(2,1,0,2,0,1) = (x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8) \begin{pmatrix} 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

or equivalently,

$$\begin{aligned} x_6 &= 2 & \qquad x_2 + x_4 &= 2 \\ 2x_1 + x_7 &= 1 & \qquad 2x_3 + x_8 &= 1 \\ x_1 + x_5 &= 0 \end{aligned}$$

Then, $SOL(M_{Sk}, \vec{C'})$ is the following 3-dimensional manifold embedded in an 8-dimensional vectorial space

$$SOL(M_{Sk}, \vec{C'}) = \{(\alpha, \beta, \gamma, 2-\beta, -\alpha, 2, 1-2\alpha, 1-2\gamma) \,:\, \alpha, \beta, \gamma \in \mathbb{R}\,\}$$

Definition 8. *Let Π be a P system and an order $\langle \gamma_1, \ldots, \gamma_d \rangle$ on $\Gamma \times H$, $\langle r_1, \ldots, r_p \rangle$ a enumeration of its set of extended rules, M_{Sk} the matrix associated with the skeleton of Π based on that enumeration of R^* and let $\vec{C}$ be the vectorial representation of a configuration C. We define the* constructor mapping *as*

$$\psi_\Pi : SOL(M_{Sk}, \vec{C}) \to \mathbb{R}^d$$

such that for all $(x_1, \ldots, x_p) \in SOL(M_{Sk}, \vec{C'})$, $\psi_\Pi((x_1, \ldots, x_p)) = (y_1, \ldots, y_d)$ verifying for all $i \in \{1, \ldots, d\}$,

$$y_i = \sum_{\gamma_i = LHS(r_k)} x_k$$

[4] This result is called the Rouche-Frobenius theorem, especially in the Spanish speaking world. This is almost certainly because the Spanish mathematician Julio Rey Pastor referred to the theorem by this name.

Notice that the set $SOL(M_{Sk}, \vec{C})$ depends on the way in which the set of extended rules is enumerated, but $\psi_\Pi(SOL(M_{Sk}, \vec{C}))$ is independent of such enumeration. Obviously, if all the coordinates of $\vec{x} \in SOL(M_{Sk}, \vec{C}')$ are natural numbers, then all the coordinates of $\psi(\vec{x})$ are also natural numbers.

Example 10. Following with the set $SOL(M_{Sk}, \vec{C}')$ from Example 9 and order $\langle((a,e),(b,e),(c,e),(a,s),(b,s),(c,s)\rangle$ on $\Gamma \times H$, we have

$$\begin{array}{l} y_1 = \sum_{(a,e)=LHS(r_k)} x_k = x_1 + x_2 = \alpha + \beta \\ y_2 = \sum_{(b,e)=LHS(r_k)} x_k = x_7 = 1 - 2\alpha \\ y_3 = \sum_{(c,e)=LHS(r_k)} x_k = x_6 = 2 \\ y_4 = \sum_{(a,s)=LHS(r_k)} x_k = x_5 = -\alpha \\ y_5 = \sum_{(b,s)=LHS(r_k)} x_k = x_3 + x_4 = 2 + \gamma - \beta \\ y_6 = \sum_{(c,s)=LHS(r_k)} x_k = x_8 = 1 - 2\gamma \end{array}$$

Therefore $\psi_\Pi(SOL(M_{Sk}, \vec{C}))$ is a 3-dimensional manifold embedded in an 6-dimensional vectorial space

$$\psi_\Pi(SOL(M_{Sk}, \vec{C})) = \{(\alpha + \beta, 1 - 2\alpha, 2, -\alpha, 2 + \gamma - \beta, 1 - 2\gamma) \,|\, \alpha, \beta, \gamma \in \mathbb{R}\}$$

Finally, we only consider the elements of $SOL(M_{Sk}, \vec{C})$ such that all its coordinates are natural numbers. We will prove below that the image of such vectors by means of the constructor mapping represent the searched configurations.

Definition 9. *Let Π be a P system, $\langle r_1, \ldots, r_p\rangle$ a enumeration of its set of extended rules, M_{Sk} the matrix associated with the skeleton of Π based on that enumeration of R^* and let $\vec{C}$ be the vectorial representation of a configuration C. We define*

- $NSOL(M_{Sk}, \vec{C})) = \{(x_1, \ldots, x_p) \in SOL(M_{Sk}, \vec{C})) \,|\, \forall i \in \{1, \ldots, p\}\, (x_i \in \mathbb{N})\}$
- *A* constructed configurations C_1 *of Π is a configuration such that $\vec{C_1} \in \psi_\Pi(NSOL(M_{Sk}, \vec{C}))$.*

Example 11. If we take $\psi_\Pi(SOL(M_{Sk}, \vec{C}))$ from Example 10

$$\psi_\Pi(NSOL(M_{Sk}, \vec{C})) = \left\{ (\alpha + \beta, 1\text{–}2\alpha, 2, -\alpha, 2 + \gamma\text{–}\beta, 1\text{–}2\gamma) \,|\, \begin{array}{l} \alpha, \beta, \gamma \in \mathbb{R} \\ \alpha + \beta \in \mathbb{N} \\ 1 - 2\alpha \in \mathbb{N} \\ -\alpha \in \mathbb{N} \\ 2 + \gamma - \beta \in \mathbb{N} \\ 1 - 2\gamma \in \mathbb{N} \end{array} \right\}$$

The set $\psi_\Pi(NSOL(M_{Sk}, \vec{C}))$ has only three elements

$$\vec{C_1} = (0, 1, 2, 0, 2, 1) \quad \vec{C_2} = (1, 1, 2, 0, 1, 1) \quad \vec{C_3} = (2, 1, 2, 0, 0, 1)$$

which correspond to the three configurations obtained in Section 2. Next we prove that the result holds in the general case.

Theorem 1. *Let Π be a P system with skeleton $Sk = (\Gamma, H, \mu, R)$ and let C be a configuration of Π. Let $\langle \gamma_1, \ldots, \gamma_d \rangle$ be an order on $\Gamma \times H$ and $\langle r_1, \ldots, r_p \rangle$ an enumeration of the extended set of rules R^* of R. Let M_{Sk} be the matrix associated with the skeleton Sk following such order and enumeration. Then, the configuration C_1 produces C in one computation step if and only if $\vec{C_1} \in \psi_\Pi(NSOL(M_{Sk}, \vec{C}))$.*

Proof. Let us consider a configuration C_1 such that $\vec{C_1} \in \psi_\Pi(NSOL(M_{Sk}, \vec{C}))$. Such configuration is a multiset C_1 on the set $\Gamma \times H$ such that for all $i \in \{1, \ldots, n\}$, $C_1(\gamma_i) \in \mathbb{N}$.

$\vec{C_1} \in \psi_\Pi(NSOL(M_{Sk}, \vec{C}))$ if and only if there exist $(x_1, \ldots, x_p) \in SOL(M_{Sk}, \vec{C})$ with $x_i \in \mathbb{N}$ for all $i \in \{1, \ldots, p\}$ such that $\psi_\Pi(x_1, \ldots, x_n) = (C_1(\gamma_1), \ldots, C_1(\gamma_d))$. By definition of the constructor mapping $\psi_\Pi : SOL(M_{Sk}, \vec{C}) \to \mathbb{R}^d$ we have for all $i \in \{1, \ldots, d\}$,

$$C_1(\gamma_i) = \sum_{\gamma_i = LHS(r_k)} x_k$$

On the other hand, we also know that $(x_1, \ldots, x_p) \in SOL(M_{Sk}, \vec{C})$, i.e.,

$$(C(\gamma_1), \ldots, C(\gamma_d)) = (x_1, \ldots, x_d) \cdot M_{Sk}$$

By construction of the matrix M_{Sk}, the previous equality means that for all $i \in \{1, \ldots, n\}$,

$$C(\gamma_i) = \sum_{j=1}^{p} x_j \cdot |RHS_{r_j}(\gamma_i)|$$

To sum up, $\vec{C_1} \in \psi_\Pi(NSOL(M_{Sk}, \vec{C}))$ if and only if there exist $(x_1, \ldots, x_p)$ such that for all $i \in \{1, \ldots, p\}$

(a) $x_i \in \mathbb{N}$
(b) $C_1(\gamma_i) = \sum_{\gamma_i = LHS(r_k)} x_k$
(c) $C(\gamma_i) = \sum_{j=1}^{p} x_j \cdot |RHS_{r_j}(\gamma_i)|$

Since R^* is a set of extended rules, $\mathcal{LHS}(R^*)$ is the set $\Gamma \times H$. Bearing this equality in mind, properties (a) and (b) claim that $\mathcal{P}^* = \langle (r_1, x_1), \ldots, (r_p, x_p) \rangle$ is a partition of C_1 with respect to R^* and property (c) claims that the configuration C can be obtained from C_1 by using the partition $\mathcal{P}^*$.

On the other hand, if C_1 produces C in one computation step, then there exist a vector $(x_1, \ldots, x_n)$ such that $\langle (r_1, x_1), \ldots, (r_p, x_p) \rangle$ is a partition of C_1 with respect to R^* verifying properties (a), (b) and (c) and therefore $\vec{C_1} \in \psi_\Pi(NSOL(M_{Sk}, \vec{C}))$.

7 A General Method

After the proof of Theorem 1, we come back to the questions asked at the end of Section 2. We wondered if there exists an algorithm such that it takes a P system

Π as input and it outputs a mapping $\mathcal{R}_\Pi$ which, for *every* configuration C' of Π, $\mathcal{R}_\Pi(C')$ is the set of all computations C such that C' is obtained from C in one computational step. A method for computing such algorithm is the following:

Given a P system Π with skeleton $Sk = (\Gamma, H, \mu, R)$,

1. Fix an order $\langle \gamma_1, \ldots, \gamma_d \rangle$ for $\Gamma \times H$.
2. Consider the pairwise representation of the rules in R according to such order.
3. Consider the extended set of rules R^* from R and fix an enumeration $\langle r_1, \ldots, r_p \rangle$ of the rules from R^* in its algebraic representation.
4. Define matrix M_{Sk} following the orders $\langle \gamma_1, \ldots, \gamma_d \rangle$ and $\langle r_1, \ldots, r_p \rangle$.

Matrix M_{Sk} is the same for all configurations. Next we provide a method for finding all the configurations C' such that C' produce a given configuration C in one computation step.

Given a configuration C of Π

1. Obtain the algebraic representation $\vec{C}$ of C according to the order $\langle \gamma_1, \ldots, \gamma_d \rangle$.
2. Find all the vectors $\vec{x}$ with natural coordinates such that $\vec{C} = \vec{x} \cdot M_{Sk}$. The set of all these vectors is called $NSOL(M_{Sk}, \vec{C})$.
3. For each $\vec{x} \in NSOL(M_{Sk}, \vec{C})$, we consider $C_{\vec{x}} = (y_1 \ldots, y_d)$ where, for all $i \in \{1, \ldots, n\}$
$$y_i = \sum_{\gamma_i = LHS(r_k)} x_k$$
4. The set $\{C_{\vec{x}} \,|\, \vec{x} \in NSOL(M_{Sk}, \vec{C})\}$ is the set of the algebraic representations of all the configurations such that produce C in one computation step.

8 Conclusions and Future Work

In this paper, we provide a general method for finding all the configurations that produce a given one in one computational step. For that purpose, we have used an algebraic representation of rules and configurations and a matrix associated with the skeleton of the P systems.

The key step of the algorithm is to find all the vectors of natural numbers that are solutions of a system of linear equations. In such a system, the number of equations is the number of objects in the alphabet multiplied by the number of labels. The number of variables in the system is the cardinal of the set of extended rules which is at least the same as the number of equations and has no upper bound.

The problem of finding the solutions with natural values of a system of linear equations is a heavy problem, specially if we consider a high number of variables and equations (which is the usual case for P systems). Nonetheless, currently there exist some powerful software tools able to deal with large numerical matrices and solve the corresponding systems under the restriction of finding natural-valued vectors.

In this way, we hope that this method can be useful for researchers interested in computing backwards in Membrane Computing, since it can consider the problem of finding the previous configurations as a problem of Integer Programming.

Finally, this work can be extended in several ways. Not only by going deeper in the concept of computing backwards along a computation (and not only in one step) but exploring if these ideas can be extended to other P system models.

Acknowledgment

The authors acknowledge the support of the project TIN2006-13425 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the Project of Excellence with *Investigador de Reconocida Valía* of the Junta de Andalucía, grant P08-TIC-04200.

References

1. O. Agrigoroaiei, G. Ciobanu: Dual P Systems. *Proceedings of Ninth Workshop on Membrane Computing* (P. Frisco, D. Corne, Gh. Păun, eds.), Technical report HW-MACS-TR-0061, School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, UK, July 2008, 45–58.
2. A. Cordón-Franco, M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez: Exploring Computation Trees Associated with P Systems. In *Membrane Computing*, LNCS 3365, Springer, 2005, 278–286.
3. M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez: Efficient Computation in Rational-Valued P Systems. Submitted, 2009.
4. Gh. Păun: *Membrane Computing. An Introduction*. Springer, Berlin, 2002.