Grammars volume 7(2004), 141–159

Polarizationless P Systems with Active Membranes

Artiom Alhazov[1,2], Linqiang Pan[1,3†]
[1]Research Group on Mathematical Linguistics
Rovira i Virgili University
Pl. Imperial Tàrraco 1, 43005 Tarragona, Spain
E-mail: {artiome.alhazov@estudiants, lp@fll}.urv.es

[2]Institute of Mathematics and Computer Science
Academy of Science of Moldova
Str. Academiei 5, Chişinău, MD 2028, Moldova
E-mail: artiom@math.md

[3]Department of Control Science and Engineering
Huazhong University of Science and Technology
Wuhan 430074, Hubei, People's Republic of China
E-mail: lqpan@mail.hust.edu.cn

Abstract

The subject of this paper is the continuation of the studies of P systems with active membranes without polarizations with the label-changing capacity of some rules. Rewriting and communication rules that do not change membrane labels can be applied either sequentially or in a maximally parallel way. We consider several classes of P systems and study their generative power.

Particularly interesting, P systems with only evolution rules used sequentially and changing labels compute exactly the Parikh sets of matrix languages; the universality is reached by P systems with evolution rules and communication rules used sequentially. By direct constructions, we also prove that SAT can be solved in linear time by systems with evolution rules changing labels, communication, and membrane division. Several open problems are also formulated.

1 Introduction

Division of membranes (inspired from division of biological cells) is the most studied feature allowing to create an exponential workspace in a linear time, and to solve "difficult" problems, typically **NP**-complete problems, in polynomial (often, linear) time (see [8, 9, 11]). Recently, also **PSPACE**-complete problems were attacked in this way (see [14, 1]).

†Corresponding author.

The computation of P systems (with symbol-objects) is the evolution of *multisets* (objects with associated multiplicities). Basically, P systems with active membranes are the *multiset-rewriting systems*, distributed within a tree-like membrane structure. The process continues until the system *halts* (no rules are applicable). The region outside the membrane system is called the *environment*; the outermost membrane is called the *skin*, and any membrane not containing other membranes is called *elementary*.

Informally speaking, six types of rules are used in P systems with active membranes: (a) multiset rewriting, (b) introducing objects into membranes, (c) sending objects out of membranes, (d) dissolving membranes, (e) dividing elementary membranes, (f) dividing non-elementary membranes.

All these rules are associated with membranes, which have a label (from a finite set of possible labels) and an "electrical charge" (which can be $+, -$, or 0). Moreover, rules of types (b), (c), (e) and (f) can change the polarization of the associated membranes, but not their labels.

In [2] one starts from the observation that the electrical charges mentioned above are not quite realistic biologically and one addresses the natural question of considering polarizationless membrane systems and their power and efficiency properties, for instance, universality and capability to solve `SAT` in linear time. In [2], one gives a partial answer to these problems.

A simpler problem is as follows: what else can be added to P systems with active membranes such that by removing the polarizations we still get universality and polynomial solutions to **NP**-complete problems? For this problem, in [2] one considers the changing of the label of a membrane. This idea comes already from [9], although not introduced there with this goal: allowing the membrane division rules to introduce membranes with new labels, not necessarily with the same label as the divided membrane.

In P systems with active membranes, the evolution rules (those of type (a)) are typically considered as only using objects and are applied maximally parallel, while the communication, dissolution, and division rules as using both objects and membranes and are applied sequentially (applying the latter rules in parallel might lead to a *conflict* as a result of a simultaneous application of rules changing membrane polarizations (or labels) in a different way).

However, if we forbid the communication operations to change labels, then we could regard them as not using membranes and apply them in parallel, like the evolution rules, as is done, for instance, in the symport/antiport P systems in [7] and in P systems with boundary rules in [3].

On the other hand, we can allow also rules of type (a) to change the polarization or the label of the membrane – and then such rules will be applied in sequentially, not to lead to label conflicts.

In this way, we get three criteria to classify the rules: changing or not polarizations, changing or not labels of membranes, using the rules in parallel or sequentially (the study of parallel application of rules changing membrane labels or polarizations without conflicts is an open topic). On the other hand, we have rules of six forms, each one being of several possible types with respect to the previous classification. A lot

of classes of P systems are obtained by combining these possibilities, which is a small "jungle" worth exploring. In [2], one reports some results of three types: simulation lemmas among different classes of P systems, universality results (as a consequence of possible simulation lemmas or directly proven), efficiency results (see Tables 2,3).

In this paper, we continue to study the P systems with active membranes without polarizations, where some rules can change the membrane labels. Also, the rewriting and communication rules not changing membrane labels can be applied either sequentially or in parallel. We consider the generative power of several classes of P systems, and obtain some results about the universality of P systems. Particularly interesting, when evolution rules are used sequentially changing labels without polarization, the Parikh sets of matrix languages can be computed by P systems with only evolution rules; the universality is reached by P systems with evolution rules and communication rules sequentially used without polarization. By direct constructions, we also prove that `SAT` can be solved in linear time by systems with evolution rules changing labels, communication, and membrane division without using polarizations.

2 Preliminaries

We assume the reader to be familiar with the fundamentals of complexity theory and formal language theory, for instance, from [6, 13, 12], as well as with the basics of membrane computing, for instance, from [9] (see `http://psystems. disco.unimib.it` for details and recent results). We only mention that RE, CF, REG denote the family of recursively enumerable, context-free, and regular languages, respectively, and that for a family FL, we denote by $PsFL$ the family of Parikh sets of languages in FL; as usual, the Parikh mapping associated with an alphabet V is denoted by Ψ_V.

$ET0L$ is the family generated by extended tabled $0L$ systems. Such a system is defined by a set N of nonterminal symbols, an axiom $S \in N \cup T$, a set T of terminal symbols, and a set H of tables. Each table is a set of context-free rules.

For $w, z \in (N \cup T)^*$ we write $w \Longrightarrow z$ if z can be obtained from w by applying some table $h \in H$: to each symbol of w, a rule from h is applied. The language generated by G is defined by $L(G) = \{w \in T^* \mid S \Longrightarrow^* w\}$.

For every language $L \in ET0L$ there exists an extended tabled $0L$ system in a normal form: the system has two tables, and the nonterminals are only trivially rewritten.

We will now introduce the notion of matrix grammars, used below in proofs.

A *matrix grammar with appearance checking* is a computationally universal rewriting system. Details can be found in [4]. For each matrix grammar there is an equivalent matrix grammar in the binary normal form (this is true both for grammars with appearance checking and without appearance checking; in the latter case, the set F and the matrices of type 3 described below are missing).

A matrix grammar $G = (N, T, S, M, F)$ in the *binary normal form* if $N = N_1 \cup N_2 \cup \{S, \#\}$, with these three sets mutually disjoint, and the matrices in M are in one of the following forms:

1. $(S \to XA)$, with $X \in N_1, A \in N_2$,
2. $(X \to Y, A \to x)$, with $X, Y \in N_1, A \in N_2, x \in (N_2 \cup T)^*, |x| \leq 2$,
3. $(X \to Y, A \to \#)$, with $X, Y \in N_1, A \in N_2$,
4. $(X \to \lambda, A \to x)$, with $X \in N_1, A \in N_2$, and $x \in T^*, |x| \leq 2$.

Moreover, there is only one matrix of type 1 (that is why one uses to write it in the form $(S \to X_{init}A_{init})$, in order to fix the symbols X, A present in it), and F consists exactly of all rules $A \to \#$ appearing in matrices of type 3; $\#$ is a trap-symbol, because once introduced, it is never removed. A matrix of type 4 is used only once, in the last step of a derivation.

For $w, z \in (N \cup T)^*$ we write $w \Longrightarrow z$ if there is a matrix in $m \in M$ such that applying once each rule of m to w one can obtain z. A rule can be skipped if it is in F and it is not applicable.

The language generated by G is defined by $L(G) = \{w \in T^* \mid S \Longrightarrow^* w\}$. The family of languages of this form is denoted by MAT_{ac}. If the set F is empty, then the grammar is said to be without appearance checking; the corresponding family of languages is denoted by MAT.

It is known that $CF \subset MAT \subset MAT_{ac} = RE$, $PsREG = PsCF \subset PsMAT \subset PsRE$, and that $CS - MAT \neq \emptyset$, $PsCS - PsMAT \neq \emptyset$ (for instance, the one-letter languages in MAT are known to be regular, [5]).

3 P Systems with Active Membranes

We will now define the model which we work with: P systems with active membranes. A P system *with active membranes* (and electrical charges) is a construct

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

where:

1. $m \geq 1$ (the initial degree of the system);
2. O is the alphabet of *objects*;
3. H is a finite set of *labels* for membranes;
4. μ is a *membrane structure*, consisting of m membranes, labeled (not necessarily in a one-to-one manner) with elements of H;
5. $w_1, \ldots, w_m$ are strings over O, describing the multisets of objects (every symbol in a string represents one copy of the corresponding object) placed in the m regions of μ;
6. R is a finite set of *developmental rules*, of the following forms:

(a_p) $[\ a \to v\ ]_h^e$,
for $h \in H, e \in \{+,-,0\}, a \in O, v \in O^*$
(object evolution rules, associated with membranes and depending on the label and the charge of the membranes, but not directly involving the membranes, in the sense that the membranes are neither taking part in the application of these rules nor are they modified by them);

(b_s) $a[\ \ ]_h^{e_1} \to [\ b\ ]_h^{e_2}$,
for $h \in H, e_1, e_2 \in \{+,-,0\}, a, b \in O$
(communication rules; an object is introduced in the membrane, possibly modified during this process; also the polarization of the membrane can be modified, but not its label);

(c_s) $[\ a\ \ ]_h^{e_1} \to [\ \ \ ]_h^{e_2} b$,
for $h \in H, e_1, e_2 \in \{+,-,0\}, a, b \in O$
(communication rules; an object is sent out of the membrane, possibly modified during this process; also the polarization of the membrane can be modified, but not its label);

(d_s) $[\ a\ \ ]_h^e \to b$,
for $h \in H, e \in \{+,-,0\}, a, b \in O$
(dissolving rules; in reaction with an object, a membrane can be dissolved, while the object specified in the rule can be modified);

(e_s) $[\ a\ \ ]_h^{e_1} \to [\ b\ \ ]_h^{e_2} [\ c\ \ ]_h^{e_3}$,
for $h \in H, e_1, e_2, e_3 \in \{+,-,0\}, a, b, c \in O$
(division rules for elementary membranes; in reaction with an object, the membrane is divided into two membranes with the same label, possibly of different polarizations; the object specified in the rule is replaced in the two new membranes by possibly new objects, and the remaining objects are duplicated);

(f_s) $[\ [\ \ \ ]_{h_1}^{e_1} \cdots [\ \ \ ]_{h_k}^{e_1}\ [\ \ \ ]_{h_{k+1}}^{e_2} \cdots [\ \ \ ]_{h_n}^{e_2}\ ]_{h_0}^{e_0}$
$\to [\ [\ \ \ ]_{h_1}^{e_3} \cdots [\ \ \ ]_{h_k}^{e_3}\ ]_{h_0}^{e_5}\ [\ [\ \ \ ]_{h_{k+1}}^{e_4} \cdots [\ \ \ ]_{h_n}^{e_4}\ ]_{h_0}^{e_6}$,
for $k \geq 1, n > k, h_i \in H, 0 \leq i \leq n$, and $e_0, \ldots, e_6 \in \{+,-,0\}$ with $\{e_1, e_2\} = \{+,-\}$; if the membrane with the label h_0 contains other membranes than those with the labels $h_1, \ldots, h_n$ specified above, then they must have neutral charges in order to make this rule applicable; these membranes are duplicated and then are part of the contents of both new copies of the membrane h_0
(division of non-elementary membranes; this is possible only if a membrane contains two immediately lower membranes of opposite polarization, + and −; the membranes of opposite polarizations are separated in the two new membranes, but their polarization can change; always, all membranes of opposite polarizations are separated by applying this rule. We omit the details here).

These rules correspond to rules $(a), (b), (c), (d), (e), (f)$ in [9]. (Note that, following [2], we have omitted the label of the left parenthesis from a pair of parentheses

which identifies a membrane.) The rules of type (a_p) are applied in the parallel way (all objects which can evolve by such rules have to evolve), while the rules of types $(b_s), (c_s), (d_s), (e_s), (f_s)$ are used sequentially, in the sense that one membrane can be used by at most one rule of these types at a time. In total, the rules are used in the non-deterministic maximally parallel manner: all objects and all membranes which can evolve, should evolve.

The result of a halting computation is the vector of natural numbers describing the multiplicity of objects expelled into the environment during the computation; the set of vectors computed in this way by all possible halting computations of Π is denoted by $Ps(\Pi)$. A P system is called *deterministic* if there is a single computation. A P system is called *confluent* if either all of its computations do not halt, or all of its computations reach the same halting configuration.

By $PsOP_m(r)$ we denote the family of sets $Ps(\Pi)$ computed as described above by P systems with at most m membranes using rules of types listed in r. We write $m = *$ if the number of membranes is unbounded. Details can be found in [9]. Finally, we will use the superscript *int* to represent the P systems with internal output (the result is obtained in a specified region, rather than in the environment).

3.1 Polarizations, Labels, Sequential, and Parallel

As we have seen in the Introduction, it is a natural question to consider P systems without membrane polarizations. Let us consider now rules – of types $(a_p), (b_s), \cdots, (f_s)$ – without polarizations. They are of the following forms (we add the subscript 0 to the previous letters identifying the six types of rules; as above, O is the alphabet of objects and H is the set of labels of membranes):

(a_{0p}) $[\ a \to v\ ]_h$, where $a \in O, v \in O^*$, and $h \in H$,

(b_{0s}) $a[\ ]_h \to [\ b\ ]_h$, where $a, b \in O$ and $h \in H$,

(c_{0s}) $[\ a\ ]_h \to [\ ]_h b$, where $a, b \in O$ and $h \in H$,

(d_{0s}) $[\ a\ ]_h \to b$, where $a, b \in O$ and $h \in H$,

(e_{0s}) $[\ a\ ]_h \to [\ b\ ]_h [\ c\ ]_h$, where $a, b, c \in O$ and $h \in H$,

(f_{0s}) $[\ [\ ]_i [\ ]_j\ ]_k \to [\ [\ ]_i\ ]_k [\ [\ ]_j\ ]_k$, where i, j, k are labels.

For the rules of dividing non-elementary membranes, we here only consider the simple form without polarizations as above shown. The meaning of such a rule is that if two membranes with labels i, j are placed inside a membrane with label k, then the membrane k is divided so that one of the new membranes k contains membrane i and the other one contains membrane j; all membranes and objects placed inside membranes i and j, as well as all membranes and objects from membrane k placed outside membranes i and j, are reproduced in the new copies of membrane k. As usual, the membranes different from i, j, k (those not involved in this rule) evolve in the standard non-deterministic maximally parallel manner.

Rules for dividing non-elementary membranes are already known to be very powerful – illustration can be found in, e.g., [1, 14]. As we see in [2] (Theorem 9 in [2]), they are powerful even in the restricted case where no polarization is used and the labels of membranes are not changed.

Using the P systems without polarizations, we can compute at least the Parikh sets of matrix languages (generated by grammars without appearance checking) (Theorem 2 in [2]). To get universality of P systems without polarizations, in [2] we allow the communication rules and the membrane division rules to change the label of a membrane. Specifically, these rules are of the following forms (here we add a prime to the previous letters identifying the types of rules to indicate the fact that the labels can be changed):

(b'_{0s}) $a[\ ]_{h_1} \to [\ b\ ]_{h_2}$, where $a, b \in O$ and $h_1, h_2 \in H$,

(c'_{0s}) $[\ a\ ]_{h_1} \to [\ ]_{h_2} b$, where $a, b \in O$ and $h_1, h_2 \in H$,

(e'_{0s}) $[\ a\ ]_{h_1} \to [\ b\ ]_{h_2} [\ c\ ]_{h_3}$, where $a, b, c \in O$ and $h_1, h_2, h_3 \in H$.

In P systems with active membranes, the evolution rules (those of type (a_p)) are typically considered as only using objects, hence can be applied in parallel. We can also allow these rules to change the label of the membrane – and then such a rule should be applied in a sequential manner, not to lead to label conflicts. We write such a rule in the form (a'_{0s}) $[\ a\ ]_{h_1} \to [\ v\ ]_{h_2}$; here subscript s means sequential application. For uniformity, we write the evolution rules without polarization not changing label and sequentially used in the form (a_{0s}) $[\ a\ ]_h \to [\ v\ ]_h,\ a \in O,\ v \in O^*,\ h \in H$.

On the other hand, if we forbid the communication operations to change labels (type (b_{0s}) or (c_{0s})), then we could regard them as not using membranes and apply them in parallel, like the evolution rules, as is done, for instance, in the symport/antiport P systems in [7] and in P systems with boundary rules in [3]. In [2], one considers rules of the following types (subscript p means parallel application):

(b_{0p}) $a[\ ]_h \to [\ b\ ]$, where $a, b \in O$ and $h \in H$,

(c_{0p}) $[\ a\ ]_h \to [\ ]\, b$, where $a, b \in O$ and $h \in H$.

(Note that we have omitted the label of the membrane on the right hand side of the rule, as the label could be changed by another rule in the same step.) In total, only one of the rules of types $(f_{0s}), (d_{0s}), (\alpha_{0s}), (\alpha'_{0s})$, $\alpha \in \{a, b, c, e\}$, can be applied for any membrane at any time, but all rules of types (a_{0p}), (b_{0p}), (c_{0p}) can be applied in parallel. The polarizationless rules mentioned are summarized in Table 1.

4 Generative Power

What is the power of P systems using the rules mentioned in subsection 3.1? In this section, we give a partial answer to this problem. Specifically, we consider $PsMAT$ (the Parikh sets of languages generated by matrix grammars without appearance checking) and $PsET0L$ (the Parikh sets of $ET0L$ languages).

	Changing Labels ($'_{0s}$)	Sequential ($_{0s}$)	Parallel ($_{0p}$)
(a)	$(a'_{0s})[\ a\ ]_h \to [\ v\ ]_{h_1}$	$(a_{0s})[\ a\ ]_h \to [\ v\ ]_h$	$(a_{0p})[\ a \to v\ ]_h$
(b)	$(b'_{0s})a[\ \ ]_h \to [\ b\ ]_{h_1}$	$(b_{0s})a[\ \ ]_h \to [\ b\ ]_h$	$(b_{0p})a[\ \ ]_h \to [\ b\ ]$
(c)	$(c'_{0s})[\ a\ ]_h \to [\ \ ]_{h_1}b$	$(c_{0s})[\ a\ ]_h \to [\ \ ]_h b$	$(c_{0p})[\ a\ ]_h \to [\ \ ]b$
(d)	Does not exist	$(d_{0s})[\ a\ ]_h \to b$	Does not exist
(e)	$(e'_{0s})[\ a\ ]_h$ $\to [\ b\ ]_{h_1}[\ c\ ]_{h_2}$	$(e_{0s})[\ a\ ]_h$ $\to [\ b\ ]_h[\ c\ ]_h$	Does not exist
(f)	Does not exist	$(f_{0s})[\ [\ \ ]_i[\ \ ]_j\ ]_k$ $\to [\ [\ \ ]_i\ ]_k[\ [\ \ ]_j\ ]_k$	Does not exist

Table 1: Rules without polarizations

4.1 The Power of Changing Labels by Evolution

The power of systems with rules of type (a'_{0s}) is still to be studied. As we will see immediately, P systems with rules of this type alone (no communication/dissolution/division) can characterize all the Parikh sets of matrix languages. Because no communication rule is available, we have to use internal output (denoted by superscript *int*).

Theorem 4.1 $PsMAT = PsOP_1^{int}(a'_{0s})$.

Proof. (i) Let us consider a matrix grammar (without appearance checking) $G = (N, T, S, M)$ in the binary normal form, hence with $N = N_1 \cup N_2 \cup \{S\}$, and with matrices of the types 1, 2, 4 as shown in Section 2. The terminal matrices $(X \to \lambda, A \to x)$ are replaced by $(X \to f,\ A \to x)$, for f being a new symbol.

We construct the P system

$$\begin{aligned} \Pi &= (O, H, [\ \ ]_{X_{init}}, A_{init}\$, R), \\ O &= N_2 \cup T \cup \{\$, \#\}, \\ H &= N_1 \cup \{f\}, \end{aligned}$$

and the set R contains the following rules (together with the rules we give explanations about their use and the functioning of the system Π)

1. $[\ A\ ]_X \to [\ x\ ]_Y$,
   for each matrix $(X \to Y, A \to x)$,
   $X \in N_1$, $Y \in N_1 \cup \{f\}$, $x \in (N_2 \cup T)^*, |x| \leq 2$.

The control symbols are stored as membrane labels, the contents of the only region is the multiset of objects corresponding to the sentential form together with object \$. The matrices are simulated accordingly.

2. $[\ \$\ ]_X \to [\ \#\ ]_f$, $X \in N_1$,

If no terminal matrix was applied (membrane label other than f) and no matrix is any more applicable, then the computation should be discarded, so the trap symbol is introduced.

3. $[\ \$\ ]_f \to [\ \lambda\ ]_f$,

When a terminal matrix is applied, \$ should be erased before the computation can finish.

4. $[\ A\ ]_f \to [\ \#\ ]_f$, $A \in N_2 \cup \{\#\}$.

If any nonterminal symbol is present in the sentential form after the terminal matrix is introduced, then the trap symbol is introduced, leading to an endless computation.

From the previous explanations it is easy to see that $\Psi_T(L(G)) = Ps(\Pi)$, which proves the inclusion $PsMAT \subseteq PsOP_1^{int}(a'_{0s})$.

(ii) Now we prove $PsOP_1^{int}(a'_{0s}) \subseteq PsMAT$.

Let us consider a P system $\Pi = (O, H, [\ ]_{\alpha_0}, w_0, R)$ with rules of type (a'_{0s}).

We will use the notations $H' = \{\alpha' \mid \alpha \in H\}$, $H'' = \{\alpha'' \mid \alpha \in H\}$, $O' = \{a' \mid a \in O\}$. We construct the matrix grammar

$$G = (O \cup H \cup H'', O' \cup H', S, M),$$

and M contains the following matrices:

1. $(S \to \alpha_0 w_0)$.

This matrix simulates the initial state of the P system.

2. $(\alpha \to \beta, a \to v)$, for rule $[\ a\ ]_\alpha \to [\ v\ ]_\beta \in R$, where $a \in O$, $v \in O^*$, $\alpha, \beta \in H$.

Each rule in R is simulated by a matrix of the above form.

3. $(\alpha \to \alpha'')$, $\alpha \in H$.
4. $(\alpha'' \to \alpha'', a \to a')$, $\alpha \in H$, $a \in O$.
5. $(\alpha'' \to \alpha')$, $\alpha \in H$.

The rules of types 3, 4, 5 are used to terminate the evolution of the P system. In this way, $L(G) = \{\alpha' w' \mid [\ w_0\ ]_{\alpha_0} \Rightarrow^* [\ w\ ]_\alpha\}$ is the language of (descriptions of) configurations of Π (reachable from the starting configuration). Let us define a regular language

$$U = H'(O')^* - \{\alpha' z_1 a' z_2 \mid [\ a\ ]_\alpha \to [\ v\ ]_\beta \in R, z_1, z_2 \in (O')^*\}.$$

Then the language $L(G) \cap U$ corresponds to the halting computations.

Note that in the language $L(G) \cap U$, α' corresponds to the label of the membrane, not the object. We define a morphism $h : O' \cup H' \longrightarrow O$ in the following way: $h(a') = a$ for $a \in O$, $h(\alpha') = \lambda$ for $\alpha \in H$. Then we have $Ps(\Pi) = \Psi_O(h(L(G) \cap U))$.

Because the family of matrix languages is effectively closed under intersection with regular languages and under morphisms, we obtain $\Psi_O(h(L(G) \cap U)) \in PsMAT$, and thus $Ps(\Pi) \in PsMAT$. □

Remark 4.1 *Note that in the above theorem we used only one membrane. It is easy to see that* $PsOP_1^{int}(a'_{0s}) = PsOP_*^{int}(a'_{0s})$*:*

Since there is no communication between the regions, the computations in all regions are independent. Given an arbitrary P system Π', construct a P system Π, consisting of the output region only, with the initial objects and rules associated to it. Then, if $L(\Pi') \neq \emptyset$, then $L(\Pi') = L(\Pi)$. The case of the empty language happens if the system never halts, possibly because of the evolution in some non-output region.

4.2 The Power of Changing Labels by Communication

Theorem 4.2 $PsET0L \subseteq PsOP_2(a_{0p}, c'_{0s})$, $PsET0L \subseteq PsOP_1^{int}(a_{0p}, c'_{0s})$.

Proof. Let us consider an $ET0L$ system in the normal form $G = (V, T, w, R_1, R_2)$. We construct the P system

$$\begin{aligned} \Pi &= (O, H, [\,[\,\,]_0\,]_s, w_s = \lambda, w_0 = wg\$, R), \\ O &= V \cup \{g, \$, \#\}, \\ H &= \{0, 1, 2, 3, s\}, \end{aligned}$$

and the set R contains the following rules (together with the rules we give explanations about their use and the functioning of the system Π)

1. $[\,\$\,]_0 \to [\,\,]_i\$$, $i \in \{1, 2\}$.
2. $[\,g \to g\$\,]_0$.

The object \$ is sent out of the internal membrane, changing the label of the internal membrane from 0 to 1 or 2, which corresponds to choosing table of rules. At the same time, the object g evolves to produce a new auxiliary object \$.

3. $[\,a \to v\,]_i$, $a \in V, v \in V^*$, for rule $a \to v \in R_i$ of G, $i \in \{1, 2\}$.
4. $[\,g \to g\$\,]_i$, $i \in \{1, 2\}$.
5. $[\,\$\,]_i \to [\,\,]_j\$$, $i, j \in \{1, 2\}$.

All the objects from V in the internal membrane evolve in parallel according to the rules from the same table R_i. The object \$ is sent out of the internal membrane, changing the label of it from i to j, which corresponds to choosing the table for the next derivation. At the same time, the object g evolves to produce a new auxiliary object \$.

6. $[\,g\,]_i \to [\,\,]_3\$$, $i \in \{1, 2\}$.

7. $[\ a \to \#\ ]_3$, for all $a \in V - T$.
8. $[\ \# \to \#\ ]_3$.
9. $[\ a\ ]_3 \to [\ \ ]_3 a$, for all $a \in T$.
10. $[\ a\ ]_s \to [\ \ ]_s a$, for all $a \in T$.

After using the rule $[\ g\ ]_i \to [\ \ ]_3\$$, no rules corresponding to the rules from the tables can be used in the internal membrane. At that time, if there is any symbol from $V-T$ in the internal membrane, then the trap-object # is introduced, which leads to an endless computation. If no symbol from $V-T$ appears in the internal membrane, then after any terminal symbol is sent out the system by rules of type 9 and 10, the computation stops.

From the previous explanations it is easy to see that $\Psi_T(L(G)) = Ps(\Pi)$, which proves the inclusion $PsET0L \subseteq PsOP_2(a_{0p}, c'_{0s})$. The statement $PsET0L \subseteq PsOP_1^{int}(a_{op}, c'_{0s})$ follows from the construction above by removing the skin membrane and removing rules 9 and 10. □

Theorem 4.3 *$PsMAT \subseteq PsOP_2(a_{0s}, c'_{0s})$, $PsMAT = PsOP_1^{int}(a_{0s}, c'_{0s})$.*

Proof. (i) $PsMAT \subseteq PsOP_2(a_{0s}, c'_{0s})$, $PsMAT \subseteq PsOP_1^{int}(a_{0s}, c'_{0s})$.

Let us consider a matrix grammar (without appearance checking) $G = (N, T, S, M)$ in the binary normal form, hence with $N = N_1 \cup N_2 \cup \{S\}$, and with matrices of the types 1, 2, 4 introduced in Section 2. We assume all matrices are injectively labeled with elements of a set B. The terminal matrices $(X \to \lambda, A \to x)$ are replaced by $(X \to f, A \to x)$, for f being a new symbol (the label of the matrix remains the same).

We construct the P system

$$\begin{aligned} \Pi &= (O, H, [\ [\ \ ]_{X_{init}}\ ]_s, \lambda, A_{init}\$, R), \\ O &= T \cup N_2 \cup \{\$, \$'\}, \\ H &= N_1 \cup \{Y_m \mid Y \in N_1, m \in B\} \cup \{f\}, \end{aligned}$$

and the set R contains the following rules:

1. $[\ \$\ ]_X \to [\ \ ]_{Y_m}\$$.
2. $[\ A\ ]_{Y_m} \to [\ x\$\$'\ ]_{Y_m}$.
3. $[\ \$'\ ]_{Y_m} \to [\ \ ]_Y\$'$, for some matrix $m : (X \to Y, A \to x)$, where $X \in N_1$, $Y \in N_1 \cup \{f\}$, $x \in (N_2 \cup T)^*, \mid x \mid \leq 2$.

These rules are used to simulate the matrix $m : (X \to Y, A \to x)$. The first rule of the matrix is simulated by the change of the label of the membrane, and the correctness of this operation is obvious (one cannot simulate one rule of the matrix without simulating at the same time also the other rule).

4. $[\ A\ ]_f \to [\ \#\ ]_f$, for $A \in N_2$.

5. $[\ \#\ ]_f \to [\ \#\ ]_f$.

After using a terminal matrix of G, no symbol from N_2 should be present in the system, otherwise the trap-object # is introduced and the computation never stops.

6. $[\ a\ ]_f \to [\ \ ]_f a$, for all $a \in T$.
7. $[\ a\ ]_s \to [\ \ ]_s a$, for all $a \in T$.

After a terminal matrix is applied, any terminal symbol is sent out.

From the previous explanations it is easy to see that $\Psi_T(L(G)) = Ps(\Pi)$, which proves the inclusion $PsMAT \subseteq PsOP_2(a_{0s}, c'_{0s})$. The statement $PsMAT \subseteq PsOP_1^{int}(a_{0s}, c'_{0s})$ follows from the construction above by removing the skin membrane and removing rules 6 and 7.

(ii) Now we prove $PsOP_1^{int}(a_{0s}, c'_{0s}) \subseteq PsMAT$. Since the objects sent into the environment cannot influence any more the internal configuration of the system (no rules of type (b'_{0s}) are allowed), and do not constitute the result (the output is internal), applying any rule $[\ a\ ]_\alpha \to [\ \ ]_\beta b$ of type (c'_{0s}) is equivalent to applying the rule $[\ a\ ]_\alpha \to [\ \lambda\ ]_\beta$ of type (a'_{0s}), and hence any P system with rules (a_{0s}, c'_{0s}) and internal result can be reduced to an equivalent P system with rules (a'_{0s}). Therefore, $PsOP_1^{int}(a_{0s}, c'_{0s}) \subseteq PsOP_1^{int}(a'_{0s}) = PsMAT$ (equality true by Theorem 4.1), which concludes the proof. □

In the two theorems above, the rules c'_{0s} were used, sending objects out of the membrane and changing its label. In the next two theorems, we will instead use the rules b'_{0s}, bringing objects into the membrane and changing its label. The Parikh sets of all $ET0L$ languages can be generated if we use the evolution rules applied in a parallel way (a_{0p}), and the Parikh sets of all matrix languages can be generated if we use the evolution rules applied sequentially (a_{0s}). All these results use two membranes, and the output is internal.

Theorem 4.4 $PsET0L \subseteq PsOP_2^{int}(a_{0p}, b'_{0s})$.

Proof. Let us consider an ET0L system in the normal form $G = (V, T, w,\ R_1, R_2)$. We construct the P system

$$\begin{aligned} \Pi &= (O, H, [\ [\ \ ]_0\ ]_s, w_s = g, w_0 = w, R), \\ O &= V \cup \{g, \$, \#\}, \\ H &= \{0, 1, 2, 3, s\}, \end{aligned}$$

and the set R contains the following rules:

1. $[\ g \to g\$\ ]_s$.
2. $\$\ [\ \ ]_0 \to [\ \$\ ]_i$, $i \in \{1, 2\}$.

In the first step, the object g produces the object \$. In the second step, the object \$ is sent in the internal membrane, changing the label of the internal membrane from 0 to 1 or 2, which corresponds to choosing the table of rules. At the same time, the object g in the skin membrane evolves to produce a new auxiliary object \$.

3. $[\ a \to v\ ]_i$, $a \in V, v \in V^*$, for rule $a \to v \in R_i$ of G, $i \in \{1,2\}$.
4. $[\ \$ \to \lambda\ ]_i$, $i \in \{1,2,3\}$.
5. $\$[\ \ ]_i \to [\ \$\ ]_j$, $i,j \in \{1,2\}$.

All the objects from V in the internal membrane evolve in parallel according to the rules from the same table R_i, and the object \$ in the internal membrane is erased. The object \$ in the skin membrane is sent in the internal membrane, changing the label of this membrane from i to j, which corresponds to choosing the table for the next derivation. At the same time, the object g in the skin membrane evolves to produce a new auxiliary object \$.

6. $g[\ \ ]_i \to [\ \$\ ]_3$, $i \in \{1,2\}$.
7. $[\ a \to \#\ ]_3$, for all $a \in V - T$.
8. $[\ \# \to \#\ ]_3$.

After using the rule $g[\ \ ]_i \to [\ \$\ ]_3$, no rules which correspond to the rules from the tables can be used in the internal membrane. At that time, if there is any symbol from $V-T$ in the internal membrane, then the trap-object # is introduced, leading to an endless computation. If no symbol from $V-T$ appears in the internal membrane, then in the next step the object \$ is erased, and the computation stops.

From the previous explanations it is easy to see that $\Psi_T(L(G)) = Ps(\Pi)$, which proves the inclusion $PsET0L \subseteq PsOP_2^{int}(a_{0p}, b'_{0s})$. □

4.3 Universality with Evolution Changing Labels

Using only rules of type (a'_{0s}), we have $PsMAT = PsOP_1^{int}(a'_{0s})$ (Theorem 4.1 in Section 4.1). If rules of types (b_{0s}), (c_{0s}) are added, then we can get the universality.

Theorem 4.5 $PsOP(a'_{0s}, b_{0s}, c_{0s}) = PsRE$.

Proof. Consider a matrix grammar $G = (N, T, S, M, F)$ with appearance checking, in the binary normal form, hence with $N = N_1 \cup N_2 \cup \{S, \#\}$ and with the matrices of the four forms introduced in Section 2. Assume that all matrices of type 3 are injectively labeled with elements of a set B. Replace the rule $X \to \lambda$ from matrices of type 4 by $X \to f$, where f is a new symbol.

We construct the P system of degree 2

$$\begin{aligned} \Pi &= (O, H, [\ [\ \ ]_0\ ]_{X_{init}}, w_0 = \lambda, w_{X_{init}} = \$A_{init}, R), \\ O &= T \cup N_2 \cup \{\$, \$', \$'', \#\}, \\ H &= N_1 \cup \{X_m \mid X \in N_1, m \in B\} \cup \{0, f\}, \end{aligned}$$

and the set R containing the following rules. We present them in blocks as used for simulating matrices of G, thus also having clear the way the system Π works.

The simulation of a matrix $(X \to Y, A \to x)$, with $X \in N_1, Y \in N_1 \cup \{f\}$, is done in one step, using the next rule:

1. $[\ A\ ]_X \to [\ x\ ]_Y$.

The first rule of the matrix is simulated by the change of the label of the skin membrane, and the second rule of the matrix is simulated by the evolving of the objects. The correctness of this operation is obvious.

The simulation of a matrix $m : (X \to Y, A \to \#)$, with $X, Y \in N_1$ and $A \in N_2$, is done in four steps, using the next rules:

2. $[\ \$\ ]_X \to [\ \$'\ ]_{Y_m}$,
3. $\$'[\quad]_0 \to [\ \$'\ ]_0$,
   $[\ A\ ]_{Y_m} \to [\ \#\ ]_f$,
4. $[\ \$'\ ]_0 \to [\quad]_0\$''$,
5. $[\ \$''\ ]_{Y_m} \to [\ \$\ ]_Y$.

The membrane with label X is used by object \$, changing the label to Y_m. In the next step, if any copy of A is present, then it introduces the trap-object # changing the label to f and the computation never stops. If no $B^{(i)}$ is present, then the objects $\$'$ evolve, returning the label of the skin membrane to Y and evolving back to the object \$, for iterating the procedure.

We also consider the following rules:

6. $[\ A\ ]_f \to [\ \#\ ]_f$, for all $A \in N_2$,
7. $[\ \# \to \#\ ]_f$,
8. $[\ a\ ]_f \to [\quad]_f a$, for all $a \in T$.

The equality $\Psi_T(L(G)) = Ps(\Pi)$ easily follows from the above explanations. □

5 Efficiency

In the brute-force algorithms to solve `SAT` as those in [8, 9], the rules of type (a_p) are used in parallel. As it was noticed in [2], this can also be done without using polarizations and without using the label changing possibilities, but evolution rules used in parallel are needed. Somewhat surprisingly, if evolution rules are used in a sequential mode, P systems with evolution changing labels, and with membrane division of type (e_{0s}) and communication rules of type (c_{0s}) turn out to be able to solve `SAT` in linear time ($\leq k_1 m + k_2 n + k_3$, where m is the number of clauses, n is a number of variables, and k_1, k_2, k_3 are constants) in a *deterministic* way.

Consider a decisional problem X. A family $\Pi_X = (\Pi_X(1), \Pi_X(2), \cdots)$ of P systems (with active membranes in our case) is called *semi-uniform* (*uniform*) if its elements are constructible in polynomial time starting from $X(n)$ (from n, respectively), where $X(n)$ denotes the instance of size n of X. We say that X can be solved in polynomial (linear) time by the family Π_X if the system $\Pi_X(n)$ will always stop in a polynomial (linear, respectively) number of steps, sending out the object `yes` if and only if the

instance $X(n)$ has a positive answer. For more detailed information about complexity classes for P systems, please refer to [10, 9].

The solution we give to SAT in this framework is semi-uniform in the sense that the P system we construct starts from a given instance of the problem; however, the construction takes a polynomial time, hence it is "honest" from this point of view.

Theorem 5.1 *P systems with rules of types* $(a'_{0s}), (c_{0s}), (e_{0s})$*, constructed in a semi-uniform manner, can solve* SAT *in linear time in a deterministic way.*

Proof. Let us consider a propositional formula in a conjunctive normal form:

$$\begin{aligned} \beta &= C_1 \wedge \cdots \wedge C_m, \\ C_i &= y_{i,1} \vee \cdots \vee y_{i,l_i},\ 1 \le i \le m, \text{ where} \\ y_{i,k} &\in \{x_j, \neg x_j \mid 1 \le j \le n\},\ 1 \le i \le m, 1 \le k \le l_i. \end{aligned}$$

The instance β of SAT will be encoded in the rules of the P system by multisets v_j and v'_j of symbols, corresponding to the clauses satisfied by *true* and *false* assignment of x_j, respectively:

$$\begin{aligned} v_j &= \{c_i \mid x_j \in \{y_{i,k} \mid 1 \le k \le l_i\}, 1 \le i \le m\}, 1 \le j \le n, \\ v'_j &= \{c_i \mid \neg x_j \in \{y_{i,k} \mid 1 \le k \le l_i\}, 1 \le i \le m\}, 1 \le j \le n. \end{aligned}$$

We construct the P system

$$\begin{aligned} \Pi &= (O, H, \mu, w_0, w_{m+2}, R), \text{ with} \\ O &= \{d_i \mid 1 \le i \le n+1\} \cup \{t_i, f_i \mid 1 \le i \le n\} \\ &\cup \{c_i \mid 0 \le i \le m\} \cup \{\$, r, \mathtt{yes}, \mathtt{no}\}, \\ \mu &= [\,[\ ]_0[\ ]_{m+4}\,]_{m+2}, \\ w_0 &= d_1, \\ w_{m+4} &= d_1, \\ w_{m+2} &= \lambda, \\ H &= \{0, \cdots, m+4\}, \end{aligned}$$

and the following rules (we accompany them with explanations about their use):

Generation phase

G1 $[\ d_i\ ]_0 \to [\ t_i\ ]_0[\ f_i\ ]_0,\ 1 \le i \le n.$

In membrane 0, we subsequently choose each variable x_i, $1 \le i \le n$, and both values *true* and $false$ are associated with it, in form of objects t_i and f_i, which are separated in two membranes with label 0. The division of membrane 0 is triggered by the objects a_i, which are introduced by the next rules from group G2; this is important in interleaving the use of these rules (hence the division of membrane 0).

G2 $[\ t_i\ ]_0 \to [\ v_i d_{i+1}\ ]_0,$
$[\ f_i\ ]_0 \to [\ v'_i d_{i+1}\ ]_0,\ 1 \le i \le n.$

In membrane 0, we subsequently look for the clauses satisfied by the truth assignments of each variable x_i, $1 \leq i \leq n$. At the same time, the objects d_{i+1} are introduced for the next round of division. After $2n$ steps, if there is at least one membrane with label 0 which contains all the symbols $c_1, \cdots, c_m$, this means that the truth assignment from that membrane satisfies all clauses, hence it satisfies formula β. Otherwise (if in no membrane with label 0 we get all objects $c_1, c_2, \ldots, c_m$), the formula β is not satisfiable.

Checking phase

C1 $[\ d_{n+1}\ ]_0 \rightarrow [\ r\ ]_1$.

At step $2n$, the objects d_{n+1} appear in every membrane with label 0. In the next step, the object d_{n+1} evolves to r, changing the label of the membranes from 0 to 1.

C2 $[\ c_i\ ]_i \rightarrow [\ \$\ ]_{i+1}$, $1 \leq i \leq m$.

In the membranes with label i, if there exist objects c_i, then the objects c_i evolve to \$, changing the label to $i+1$. Note that only one copy of c_i (if there are many copies) evolves in each membrane. Note also that the objects c_i can only evolve when the membrane has label i, which means that the objects $c_1, c_2, \cdots, c_{i-1}$ already existed in the corresponding membrane. In this way, after m steps we can find whether there is a membrane which contained $c_1, c_2, \cdots, c_m$. These are exactly the membranes with label $m+1$.

C3 $[\ r\ ]_{m+1} \rightarrow [\ \ ]_{m+1} r$.

The object r exits to the skin membrane. Note that the skin membrane may have exponentially many copies of the object r.

C4 $[\ r\ ]_{m+2} \rightarrow [\ \texttt{yes}\ ]_{m+3}$.

If the problem has a solution, then one copy of object r evolves to `yes`, changing the label of the skin membrane from $m+2$ to $m+3$.

Output phase

O1 $[\ d_i\ ]_{m+4} \rightarrow [\ d_{i+1}\ ]_{m+4}$, $1 \leq i \leq 2n+m+2$.

O2 $[\ d_{2n+m+3}\ ]_{m+4} \rightarrow [\ \ ]_{m+4}\texttt{no}$.

In the meantime, inside the membrane with label $m+4$ object d_1 evolves into d_{2n+m+3} and then an object `no` is sent to the skin at step $2n+m+3$.

O3 $[\ \texttt{yes}\ ]_{m+3} \rightarrow [\ \ ]_{m+3}\texttt{yes}$.

O4 $[\ \texttt{no}\ ]_{m+2} \rightarrow [\ \ ]_{m+2}\texttt{no}$.

If β has a solution, then object `yes` is ejected into the environment. If formula β is not satisfiable, then the label of the skin membrane remains $m+2$ and object `no` is sent into the environment. In either case, the computation stops at step $2n+m+4$ of the computation.

It is easy to see that the system Π can be constructed in a polynomial time starting from β, and the satisfiability of formula β was decided in linear time. This concludes the proof. □

α'_{0s}	$PsMAT$	$PsET0L$	$PsRE$
-	$\subset PsOP(a_{0p}, b_{0s}, c_{0s}, d_{0s}, e_{0s})$ [2]	Open	Open
(a'_{0s})	$= PsOP_1(a'_{0s})$		$= PsOP_2(a'_{0s}, b_{0s}, c_{0s})$
(b'_{0s})		$\subseteq PsOP_2^{int}(a_{0p}, b'_{0s})$	$= PsOP(a_{0p}, b'_{0s}, c_{0s})$ [2]
(c'_{0s})	$\subseteq PsOP_2(a_{0s}, c'_{0s})$ $= PsOP_1^{int}(a_{0s}, c'_{0s})$	$\subseteq PsOP_2(a_{0p}, c'_{0s})$ $\subseteq PsOP_1^{int}(a_{0p}, c'_{0s})$	$= PsOP(a_{0p}, b_{0s}, c'_{0s})$ [2]
(e'_{0s})			$= PsOP(a_{0p}, c_{0s}, e'_{0s})$ [2] $= PsOP^{int}(a_{0p}, e'_{0s})$ [2]

Table 2: Generative power depending on rules changing labels

6 Final Remarks

With the goal of removing the polarization from P systems with active membranes, in this paper and in [2] the possibility is investigated to allow instead to change the labels of membranes, and the attempt was successful in the case of evolution rules (of type (a'_{0s})), in the case of rules for sending objects out of a membrane (of type (c'_{0s})) and in the case of rules for dividing membranes (of type (e'_{0s})) – losing however the determinism. The case of using rules of type (b'_{0s}) (introducing objects into membranes) for changing the labels has remained *open* in what concerns the simulation results – as well as the possibility to solve `SAT` in polynomial time – but not in what concerns the universality.

The use of non-polarized membranes suggests further possibilities in what concerns the application of rules. As it is shown in Section 3.1, we get three criteria to classify the rules: changing or not polarizations, changing or not labels of membranes, using the rules in parallel or sequentially. On the other hand, we have rules of six forms, each one being of several possible types with respect to the previous classification. A lot of classes of P systems are obtained by combining these possibilities. In this paper and in [2], this "jungle" of types of systems was explored for results of three types: simulation lemmas among different classes of P systems, universality results (as a consequence of possible simulation lemmas or directly proved), efficiency results. Now we summarize the efficiency results and the generative power results from these two papers in Tables 2,3 ([2] denotes the corresponding results from the paper [2]).

We close this section by pointing out some open problems, some of which were scattered in [2].

1. Can a P system with active membranes and polarizations be simulated by a system without polarizations but allowed to change the label of membranes when introducing objects into them?

2. Can `SAT` be solved in polynomial time by P systems without polarizations but allowed to change the label of membranes when introducing objects into them?

3. What is the power of P systems without polarizations and label changing?

α'_{0s}	Deterministic Uniform	Deterministic Semi-Uniform	Confluent Uniform
-		$(a_{0p}, c_{0s}, d_{0s}, e_{0s}, f_{0s})$ [2]	
(a'_{0s})		$(a'_{0s}, c_{0s}, e_{0s})$	
(c'_{0s})	$(a_{0p}, c'_{0s}, e_{0s})$ [2]		
(e'_{0s})	$(a_{0p}, b_{0p}, c_{0s}, e'_{0s})$ [2]		$(a_{0p}, b_{0s}, c_{0s}, e'_{0s})$ [2]

Table 3: Efficiency depending on rules changing labels (Solving `SAT` in linear time)

Specifically: what is the size of the family $PsOP(a_{0p}, b_{0s}, c_{0s}, d_{0s}, e_{0s})$? Is it equal to $PsRE$? If not, then what about its relation with $PsET0L$?

4. Rules of type (α_s) or (α'_{0s}) are stronger than rules of type (α_{0s}), where $\alpha \in \{b, c, e\}$. Are these relations "stronger/weaker" proper?

5. The universality of P systems with parallel communication (i.e., with rules of types (b_{0p}) and/or (c_{0p})) remains as an open question.

6. Are the inclusions $PsMAT \subseteq PsOP_2(a_{0s}, c'_{0s})$, $PsET0L \subseteq PsOP_2(a_{0p}, c'_{0s})$, $PsET0L \subseteq PsOP_1^{int}(a_{0p}, c'_{0s})$, $PsET0L \subseteq PsOP_2^{int}(a_{0p}, b'_{0s})$ proper?

Acknowledgements. The first author is supported by the project TIC2002-04220-C03-02 of the Research Group on Mathematical Linguistics, Tarragona. The first author acknowledges IST-2001-32008 project "MolCoNet" and also the Moldovan Research and Development Association (MRDA) and the U.S. Civilian Research and Development Foundation (CRDF), Award No. MM2-3034 for providing a challenging and fruitful framework for cooperation.

The second author is supported by grant DGU-SB2001-0092 from Spanish Ministry for Education, Culture, and Sport, National Natural Science Foundation of China (Grant No. 60373089), and Huazhong University of Science and Technology Foundation.

References

[1] A. Alhazov, C. Martín-Vide, L. Pan, Solving a **PSPACE**-Complete Problem by P Systems with Restricted Active Membranes, *Fundamenta Informaticae*, 58, 2, 2003, 66-77.

[2] A. Alhazov, L. Pan, Gh. Păun, Trading Polarizations for Labels in P Systems with Active Membranes, submitted, 2003.

[3] F. Bernardini, V. Manca, P Systems with Boundary Rules, *Membrane Computing* (Gh. Păun, G. Rozenberg, A. Salomaa, C. Zandron, Eds), Proc. WMC Curtea de Argeş, 2002, *LNCS* 2597, Springer-Verlag, 2003, 107–118.

[4] J. Dassow, Gh. Păun, *Regulated Rewriting in Formal Language Theory*, Springer-Verlag, Berlin, 1989.

[5] D. Hauschild, M. Jantzen, Petri Nets Algorithms in the Theory of Matrix Grammars, *Acta Informatica* 31, 1994, 719–728.

[6] Ch.P. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, MA, 1994.

[7] A. Păun, Gh. Păun, The Power of Communication: P Systems with Symport/Antiport, *New Generation Computers* 20, 3, 2002, 295–306.

[8] Gh. Păun, P Systems with Active Membranes: Attacking **NP**-Complete Problems, *Journal of Automata, Languages and Combinatorics*, 6, 1, 2001, 75–90.

[9] Gh. Păun, *Computing with Membranes: An Introduction*, Springer-Verlag, Berlin, 2002.

[10] M. J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini, The Polynomial Complexity Class with Active Membranes, submitted, 2003.

[11] M. J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini, *Teoría de la Complejidad en Modelos de Computatión Celular con Membranas*, Editorial Kronos, Sevilla, 2002.

[12] G. Rozenberg, A. Salomaa, Eds, *Handbook of Formal Languages* (3 volumes), Springer, Berlin, 1997.

[13] A. Salomaa *Formal Languages*, Academic Press, New York, 1973.

[14] P. Sosík, Solving a **PSPACE**-Complete Problem by P Systems with Active Membranes, *Proceedings of the Brainstorming Week on Membrane Computing* (M. Cavaliere, C. Martín-Vide, Gh. Păun, Eds), Report GRLMC 26/03, 2003, 305–312.