On the Efficiency of P Systems with Active Membranes and Two Polarizations

Artiom ALHAZOV[1,2], **Rudolf FREUND**[3]

[1] Research Group on Mathematical Linguistics
Rovira i Virgili University
Pl. Imperial Tarraco 1, 43005 Tarragona, Spain
E-mail: `artiome.alhazov@estudiants.urv.es`

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

[3] Faculty of Informatics
Vienna University of Technology
Favoritenstr. 9, A-1040 Wien, Austria
E-mail: `rudi@emcc.at`

Abstract

We present an algorithm for deterministically deciding `SAT` in linear time by P systems with active membranes using only two polarizations and rules of types (a), (c) and (e). Moreover, various restrictions on the general form of the rules are considered: global, non-renaming, independent of the polarization, preserving it, changing it, producing two membranes with different polarizations, having exactly one or two objects in (each membrane of) the right-hand side, improving results from [1]. Several problems related to different combinations of these restrictions are formulated, too.

1 Introduction

Membrane systems are biologically motivated theoretical models of distributed and parallel computing. The most interesting questions probably are completeness (solving every solvable problem) and efficiency (solving a hard problem in feasible time). We here address the latter problem, i.e., we shall give an algorithm how to decide `SAT` in linear time using only two polarizations in P systems with active membranes.

The question of removing the polarizations (charges $+, -, 0$ associated with the membranes) from P systems with active membranes without diminishing their computing power or their efficiency in solving computationally hard problems in a feasible time was formulated several times and was recently considered in various contexts (with the polarizations replaced by various other features, such as label changing – see, e.g., [2], [3]). Here, following [1], we present another way for improving previous results: the number of polarizations can be decreased to two, without introducing new features.

There are numerous results of solving such (mostly NP-complete) problems as `SAT`, `HPP`, Validity, Subset-Sum, Knapsack, Vertex Cover, Clique, `QBF-SAT` by P systems with active membranes with three polarizations (e.g., see [2], [3], [4], [5], [9], [10], [12], [13], [14], [16], [17], [18], [20], [21]). The ability of the systems to act depending on the membrane polarizations and to change them is a powerful control feature, the use of which is not necessary if one pays the price of changing membrane labels. Another result is solving `SAT` in a semi-uniform manner, without polarizations and without changing labels, but also using membrane dissolution and non-elementary membrane division. Here we show that two polarizations are enough even when restricting the types of rules to (a), (c), and (e). It remains as an open question whether polarizations can be completely removed, and we *conjecture* that the answer is negative.

Moreover, we consider a few restrictions on the general form of the rules, under which it is still possible to solve `SAT`. The motivations of considering these restrictions are of three kinds: bringing the construction closer to biological cells (making it as "realistic" as possible); building a normal form (as restrictive as possible), for the possible future direct simulation results; and finding out which aspects of active membranes are essential for the efficiency of P systems.

2 Prerequisites

The reader is assumed to be familiar with basic elements of formal language theory. For an alphabet V, by V^* we denote the free monoid generated by V under the operation of concatenation; the *empty string* is denoted by λ, and $V^*\setminus\{\lambda\}$ is denoted by V^+. By $\mathbf{N}$ we denote the set of positive integers, and $\mathbf{N}_0 := \mathbf{N} \cup \{0\}$ is the set of non-negative integers. In the following we will not distinguish between a vector $(y_1, ..., y_\beta) \in \mathbf{N}_0^\beta$, its representation by a multiset or its representation by a string with Parikh vector $(y_1, ..., y_\beta)$. For more notions as well as basic results from the theory of formal languages, the reader is referred to [6] and [19].

We also assume the reader to be familiar with the basic elements of membrane computing, e.g., from [15] (details can be found at `http://psystems.disco.unimib.it`), in particular, with P systems with active membranes.

For the sake of completeness, we recall the definition of P systems with active membranes for the case when only rules of types (a) to (e) are used; in a more general way, as in the original definition, we allow the polarizations to be arbitrary non-negative integers:

A *P system system with active membranes* (of degree $m \geq 1$) is a construct of the form

$$\Pi = (O, E, \mu, w_1, \cdots, w_m, e_1, \cdots, e_m, R),$$

where O is the alphabet of objects, $E = \{0, \cdots, n-1\}$ with $n \geq 1$ is the set of electrical charges (polarizations), μ is the membrane structure (with m membranes, bijectively labelled with $1, 2, \cdots, m$; by H we denote the set of labels $\{1, 2, \cdots, m\}$),

$w_1, \cdots, w_m$ are strings over O indicating the multisets of objects at the beginning present in the m regions of μ, $e_1, \cdots, e_m$ are the polarizations at the beginning assigned to the membranes $1, \cdots, m$, and R is a finite set of rules of the following forms:

(a) $[\ a \to v\ ]_h^i,\ a \in O,\ v \in O^*,\ h \in H,\ i \in E$

(evolution rules, used in parallel in the region of membrane h, provided that the polarization of the membrane is i);

(b) $a[\ \ ]_h^i \to [\ b\ ]_h^j,\ a, b \in O,\ h \in H,\ i, j \in E$

(communication rules, sending an object into a membrane, possibly changing the polarization of the membrane);

(c) $[\ a\ ]_h^i \to [\ \ ]_h^j b,\ a, b \in O,\ h \in H,\ i, j \in E$

(communication rules, sending an object out of a membrane, possibly changing the polarization of the membrane);

(d) $[\ a\ ]_h^i \to b,\ a, b \in O,\ h \in H,\ i \in E$

(membrane dissolution rules; in reaction with an object, the membrane is dissolved);

(e) $[\ a\ ]_h^i \to [\ b\ ]_h^j[\ c\ ]_h^k,\ a, b, c \in O,\ h \in H,\ i, j, k \in E$

(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, and the object specified in the rule is replaced in the two new membranes by possibly new objects).

The rules of types (b), (c), (d), and (e) are considered as involving the membrane, hence, we assume at most one of such a rule to be used for each membrane in a given step; the use of rules is maximally parallel, with the rules chosen in a non-deterministic manner.

An output is associated with a halting computation – and only with halting computations – in the form of the objects sent into the environment during the computation. When using a P system Π for decision problems, we also specify an input membrane i_0, where we put the input to be analysed is put in addition to the axiom w_{i_0}; in sum, we then write

$$\Pi = (O, E, \mu, w_1, \cdots, w_m, e_1, \cdots, e_m, R, i_0).$$

3 Solving `SAT` in Linear Time

Throughout this section we use the following notation for instances of the `SAT` problem:

We consider a propositional formula in conjunctive normal form:

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

i.e., n is the number of variables and m is the number of clauses, hence, to β the size (n,m) is associated. For arbitrary $(n,m) \in \mathbf{N}^2$, we denote the family of `SAT` problems of size (n,m) by $\texttt{SAT}(n,m)$.

3.1 Using global rules

As it was shown in [1], $\texttt{SAT}(n,m)$ can be decided in linear time (linear with respect to n and m, i.e., the algorithm has time complexity $O(n+m)$) by a uniform family of P systems with two polarizations, only using rules of types (a), (c), and (e). Throughout this section we will always restrict ourselves to restricted variants of these types of rules.

We first recall the theorem from [1], giving the construction of the proof and short explanations as well as repeating the example that illustrates the corresponding construction.

Theorem 1. $\texttt{SAT}(n,m)$ *can be deterministically decided in linear time (linear with respect to n and m) by a uniform family of P systems with active membranes with two polarizations and global rules of types* (a), (c), *and* (e).

Proof. An instance β of the $\texttt{SAT}(n,m)$ problem as described above is encoded as a multiset over

$$V(n,m) = \{x_{i,j,j}, x'_{i,j,j} \mid 1 \leq i \leq m, 1 \leq j \leq n\}.$$

The object $x_{i,j,j}$ ($x'_{i,j,j}$) represents the variable x_j appearing in the clause C_i without (with) negation. Thus, the input multiset is

$$\begin{aligned} w &= \{x_{i,j,j} \mid x_j \in \{y_{i,k} \mid 1 \leq k \leq l_i\}, 1 \leq i \leq m, 1 \leq j \leq n\} \\ &\cup \{x'_{i,j,j} \mid \neg x_j \in \{y_{i,k} \mid 1 \leq k \leq l_i\}, 1 \leq i \leq m, 1 \leq j \leq n\}, \end{aligned}$$

which is placed into membrane 2 in addition to the initial symbol d_0 in the P system $\Pi(n,m)$ we will construct for any given $(n,m) \in \mathbf{N}^2$:

$$\begin{aligned} \Pi(n,m) &= (O(n,m), \{0,1\}, [_1\ [_2\ ]_2\ ]_1, t_0, d_0, 0, 0, R, 2), \\ O(n,m) &= \{x_{i,j,k}, x'_{i,j,k} \mid 1 \leq i \leq m, 0 \leq k \leq j \leq n\} \cup \{z, o, \texttt{yes}, \texttt{no}\} \\ &\cup \{c_{i,j} \mid 0 \leq i \leq m, 0 \leq j \leq n\} \cup \{c_i \mid 0 \leq i \leq m\} \\ &\cup \{d_i \mid 0 \leq i \leq n+1\} \cup \{e_i \mid 0 \leq i \leq m+1\} \\ &\cup \{t_i \mid 0 \leq i \leq n+2m+4\}; \end{aligned}$$

R contains the following rules (grouped by sub-tasks; see [1] for more explanations and details):

Global control in skin membrane

- $[\ t_i \rightarrow t_{i+1}\ ]^0$, $0 \leq i \leq n+2m+2$.

Generation phase

- $[\ d_i\ ]^e \rightarrow [\ d_{i+1}\ ]^0 [\ d_{i+1}\ ]^1$, $e \in \{0,1\}$, $0 \leq i < n-1$;

- $[\ x_{i,j,k} \to x_{i,j,k-1}\ ]^e$,
  $[\ x'_{i,j,k} \to x'_{i,j,k-1}\ ]^e$, $e \in \{0,1\}$, $1 \le i \le m$, $1 \le k \le j \le n$;
- $[\ x_{i,j,0} \to \lambda\ ]^0$,
  $[\ x_{i,j,0} \to c_{i,j}\ ]^1$,
  $[\ x'_{i,j,0} \to c_{i,j}\ ]^0$,
  $[\ x'_{i,j,0} \to \lambda\ ]^1$, $1 \le i \le m$, $1 \le j \le n$;
- $[\ c_{i,j} \to c_{i,j+1}\ ]^e$, $e \in \{0,1\}$, $1 \le i \le m$, $1 \le j < n$;
- $[\ d_n \to d_{n+1} z\ ]^1$,
  $[\ d_n \to d_{n+1}\ ]^0$.

During each of the first n steps, every elementary membrane is duplicated, in order to examine all possible truth assignments to the variables $x_1, \cdots, x_n$.

In step i of the generation phase, one of the membranes resulting from the application of the rule

$$[\ d_i\ ]^e \to [\ d_{i+1}\ ]^0 [\ d_{i+1}\ ]^1$$

gets polarization 0, corresponding to assigning the truth value `false` to x_i (and in this case the clauses where $\neg x_i$ appears are satisfied), and the other membrane gets polarization 1, corresponding to assigning the truth value `true` to x_i (and in this case those clauses where x_i appears without negation are satisfied). Due to the application of the rules

$$[\ x_{i,j,0} \to \lambda\ ]^0, [\ x_{i,j,0} \to c_{i,j}\ ]^1, [\ x'_{i,j,0} \to c_{i,j}\ ]^0, [\ x'_{i,j,0} \to \lambda\ ]^1,$$

only those variables "survive" which correspond to the correct truth assignment at the moment the last index has reached the ground level 0.

After the end of this first phase of the algorithm, 2^n elementary membranes (each of them with label 2) have been produced, each of them containing d_{n+1} and objects $c_{i,n}$ for all clauses C_i that are satisfied. Every membrane with polarization 1 also contains an object z. This procedure described so far in total takes $n+1$ step.

Transition phase

- $[\ z\ ]^1 \to [\ \ ]^0 o$;
- $[\ d_{n+1} \to e_1\ ]^e$, $e \in \{0,1\}$;
- $[\ c_{i,n} \to c_i\ ]^e$, $e \in \{0,1\}$, $1 \le i \le n$.

By the application of the rule $[\ z\ ]^1 \to [\ \ ]^0 o$ the polarization of the membranes polarized by 1 is reset to zero again by passing through the surrounding membrane, thereby also yielding the "garbage" symbol o within the skin membrane. After this single step of the transition phase all the elementary membranes now have the polarization 0 and contain e_1 as well as c_i for every satisfied clause C_i.

Checking phase

- $[\ c_1\ ]^0 \to [\ \ ]^1 o$;
- $[\ e_i \to e_{i+1} z\ ]^0$, $1 \le i < m$;
- $[\ c_1 \to \lambda\ ]^1$;
- $[\ c_i \to c_{i-1}\ ]^1$, $2 \le i \le m$;
- $[\ e_m \to e_{m+1}\ ]^0$;
- $[\ e_{m+1}\ ]^1 \to [\ \ ]^1 \mathtt{yes}$.

All clauses are satisfied if and only if all objects $c_1, \cdots, c_m$ are present in some membrane, and at the end all objects c_i, $1 \le i \le m$, have been sent out into the skin membrane. While checking the last clause, no object z (for resetting the polarization of the membrane as this is done in the preceding steps) is produced from e_m by applying the rule $[\ e_m \to e_{m+1}\ ]^0$, hence, e_{m+1} will be present in a membrane with polarization 1 thus allowing for the application of the rule

$$[\ e_{m+1}\ ]^1 \to [\ \ ]^1 \mathtt{yes}$$

indicating that the corresponding elementary membrane represented a solution of the given satisfiability problem. In total, this phase takes $2m$ steps.

Output phase

- $[\ \mathtt{yes}\ ]^0 \to [\ \ ]^1 \mathtt{yes}$;
- $[\ t_{n+2m+3}\ ]^0 \to [\ \ ]^0 \mathtt{no}$.

Every elementary membrane which after the first $n+1$ steps had represented a solution of the given satisfiability problem, after $n+1+1+2m$ steps has sent a copy of `yes` into the skin membrane, and in the next step one of these copies exits into the environment by using the rule

$$[\ \mathtt{yes}\ ]^0 \to [\ \ ]^1 \mathtt{yes}$$

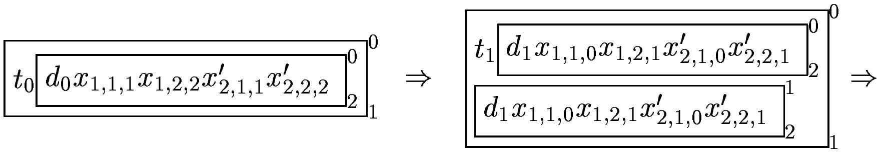
thus giving the positive result `yes` and changing the skin polarization to 1 in order to prevent further output. If, on the other hand, the given satisfiability problem has no solution, after $n+2m+3$ steps the polarization of the skin membrane will still be 0, hence, the rule

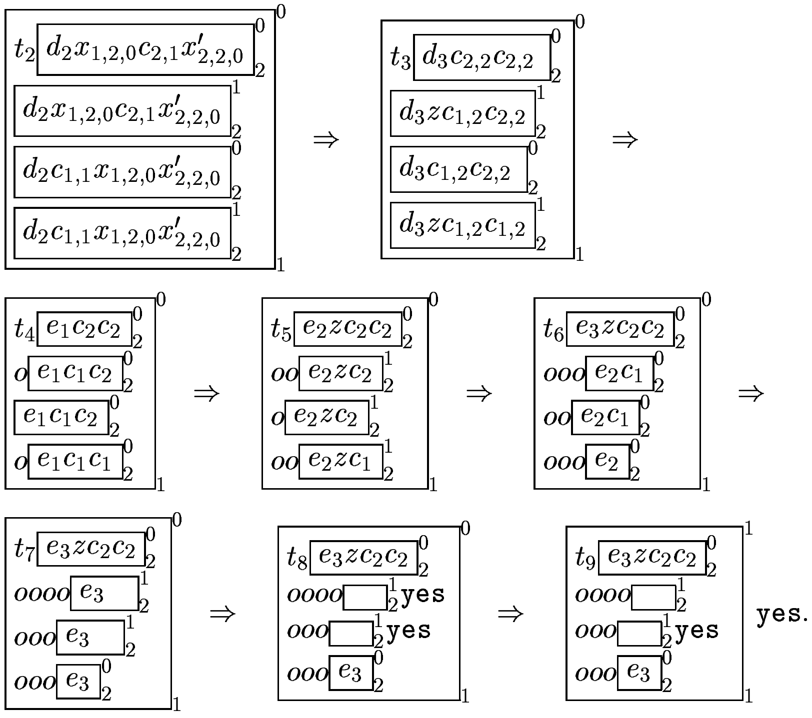
$$[\ t_{n+2m+3}\ ]^0 \to [\ \ ]^0 \mathtt{no}$$

sends out the correct answer `no`.

We now illustrate the construction elaborated above by the following example:

$$\gamma = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$$





It is worth noticing that the rules are *global*: the same set of rules is valid for all membranes, i.e., in the rules, the labels of the membranes can be omitted. We also note that in the construction from [1] the membrane division rules do not depend on the polarization (and therefore can be omitted in the meaning of "applicable for any membrane"), and the contents of membranes after division is identical, but the polarizations are different. Finally, replacing the last rule $[\ t_{n+2m+3}\ ]^0 \to [\ \ ]^0\texttt{no}$ by

$$[\ t_{n+2m+3}\ ]^0 \to [\ \ ]^1\texttt{no}$$

will lead to an equivalent construction, where in addition every rule of type (c) changes the polarization (the superscript $\neg$ will be used to denote this variant).

Thus, all rules used are even of the following restricted forms (where the interpretation of the subscripts g, $g1$, and $g2$ is explained in the subsequent subsection; the superscript $\neg$ indicates that the polarization is changed):

(a_g) $[\ a \to v\ ]^i$,

(c_{g1}) $[\ a\ ]^i \to [\ \ ]^{\neg}b$,

(e_{g2}) $[\ a\ ] \to [\ b\ ]^0[\ b\ ]^1$

where $a, b \in O$, $v \in O^*$, $h \in H$, $i \in \{0,1\}$.

According to the explanations given above, we now have even proved a stronger result than that already shown in [1]. □

3.2 Using rules of a specific "normal form"

In this subsection we now consider the following forms (particular cases) of the types (a), (c), (e) of rules (where $a, b, c \in O$, $h \in H$, $i \in \{0,1\}$):

(a_{gb}) $[\ a \to bc\ ]^i$ (global split rule)

(a_{gu}) $[\ a \to b\ ]_h^i$ (rename only)

(c_{np1}) $[\ a\ ]_h \to [\ \ ]_h^{\neg} a$ (exit only, polarization switched)

(c_{gp1}) $[\ a\ ] \to [\ \ ]^{\neg} b$ (global exit rule, polarization switched)

(c_{gny}) $[\ \mathtt{yes}\ ]^0 \to [\ \ ]^1 \mathtt{yes}$ (a special rule for ejecting the result)

(e_{gp0}) $[\ a\ ] \to [\ b\ ][\ c\ ]$ (global polarizationless division rule)

(e_{gp2}) $[\ a\ ] \to [\ b\ ]^0[\ c\ ]^1$ (global polarization-independent division rule, producing membranes of different polarizations)

In the subscripts of the rules, we write g if the rule is global (does not depend on the label of the membrane), n if the rule is not-renaming (the object(s) in (each membrane of) the right-hand side is(are) the same as the object in the left-hand side), p if the rule does not depend on the polarization, 0 if the rule preserves it, 1 if the rule changes it, 2 if the rule produces two membranes with different polarizations, and b (u) if the number of the objects in (each membrane of) the right-hand side is two (one, respectively). Finally, y is used if the rule acts on object the `yes`.

The main idea of the possible restriction is the following: to try to make rules of types (c) and (e) independent of the polarization by remembering the needed value in a corresponding object, and then decoding it by generating copies of z if needed (using such an approach, the computation slows down by a constant factor). In the same time, other restriction are put on the general form of the rules, leading to the following theorem:

Theorem 2. *`SAT`(n, m) can be deterministically decided in linear time (linear with respect to nm, i.e., the algorithm has time complexity $O(nm)$) by a uniform family of P systems with active membranes with two polarizations and rules of the forms (a_{gb}), (c_{np1}), (c_{gny}), and (e_{gp0}).*

Proof. An instance β of the `SAT`(n, m) problem as described above is encoded as a multiset over

$$V(n, m) = \{x_{i,j,j,0}, x'_{i,j,j,0} \mid 1 \leq i \leq m, 1 \leq j \leq n\}.$$

The object $x_{i,j,j,0}$ represents the variable x_j appearing in the clause C_i without negation, and the object $x'_{i,j,j,0}$ represents the variable x_j appearing in the clause C_i with negation. Thus, the input multiset is

$$\begin{aligned} w &= \{x_{i,j,j,0} \mid x_j \in \{y_{i,k} \mid 1 \leq k \leq l_i\}, 1 \leq i \leq m, 1 \leq j \leq n\} \\ &\cup \{x'_{i,j,j,0} \mid \neg x_j \in \{y_{i,k} \mid 1 \leq k \leq l_i\}, 1 \leq i \leq m, 1 \leq j \leq n\}, \end{aligned}$$

which has to be put into membrane 2 in addition to the initial symbol $d_{0,0}$ in the P system $\Pi(n,m)$ defined below:

$$\begin{array}{rcl}
\Pi(n,m) & = & (O(n,m),\{0,1\},[_1\ [_2\ \ ]_2\ ]_1,t_0,d_{0,0},0,0,R,2), \\
O(n,m) & = & \{x_{i,j,k,l},x'_{i,j,k,l} \mid 1 \le i \le m, 0 \le k \le j \le n, 0 \le l \le 3\} \\
 & \cup & \{z,o,\mathtt{yes},\mathtt{no}\} \\
 & \cup & \{c_{i,j,l} \mid 0 \le i \le m, 0 \le k \le n, 0 \le l \le 3\} \\
 & \cup & \{c_{i,l} \mid 0 \le i \le m, 0 \le l \le 2\} \\
 & \cup & \{d_{i,l},d'_{i,j} \mid 0 \le i \le n, 0 \le l \le 3\} \\
 & \cup & \{e_{i,l} \mid 0 \le i \le m+1, 0 \le l \le 2\} \\
 & \cup & \{t_i \mid 0 \le i \le 2mn+4n+3m+4\};
\end{array}$$

Let us briefly describe the meaning of the objects: objects $x_{i,j,k,l}, x'_{i,j,k,l}$ encode the instance of the problem, objects $c_{i,j,l}, c_{i,l}$ represent clauses satisfied, objects $d_{i,l}, d'_{i,l}$ control the generation phase, objects $e_{i,l}$ control the checking phase, objects t_i produce the negative answer in case no positive answer is given. Object z is used to change the polarization of the membrane, object o is a "garbage" object, and finally `yes` and `no` are the possible results. The subscript l is used to switch between different states within cycles of the generation or the checking phase.

R contains the following rules (we also give explanations for the use of these rules):

Global control in skin membrane

- $[\ t_i \rightarrow t_{i+1}o\ ]^0,\ 0 \le i \le 2mn+4n+3m+3.$

The control variables $t_i$only occur in exactly one copy in the skin membrane. As we shall see at the end of the description of the whole algorithm, after $2mn+4n+3m+3$ derivation steps in the corresponding P system $\Pi(n,m)$ the answer `yes` appears outside the skin membrane if the given satisfiability problem has a solution, whereas in the case that no solution exists, one step later the answer `no`appears in the environment.

The main task of the algorithm is accomplished in the generation phase of the algorithm where for each possible truth assignment to the n variables one elementary membrane is generated which after $n+1$ steps will contain all the information needed to decide whether it represents a solution of the given problem or not:

Generation phase

- $[\ d_{i,0}\ ] \rightarrow [\ d_{i,1}\ ][\ d'_{i,1}\ ],\ 0 \le i \le n-1;$
- $[\ d_{i,1} \rightarrow d'_{i,2}z\ ]^0,$
  $[\ d'_{i,1} \rightarrow d_{i,2}o\ ]^0,\ 0 \le i \le n-1;$
- $[\ d_{i,2} \rightarrow d_{i,3}z\ ]^0,$
  $[\ d'_{i,2} \rightarrow d_{i,3}o\ ]^0,\ 0 \le i \le n-1;$

- $[\ d_{i,3} \to d_{i+1,0}o\ ]^{e}$, $e \in \{0,1\}$, $0 \le i \le n-1$;
- $[\ z\ ]_2 \to [\ \ ]_2^{\neg} z$.

There are n cycles, each taking four steps and duplicating every elementary membrane in order to examine all possible truth assignments to the variables $x_1, \cdots, x_n$. Symbols $d_{i,1}$ ($d'_{i,1}$) correspond to the value `true` (`false`) of x_i, respectively. In the case of the value `true`, the membrane polarization changes (using object z) to 1 two steps after the division, and then is restored.

- $[\ x_{i,j,k,l} \to x_{i,j,k,l+1}o\ ]^{0}$,
  $[\ x'_{i,j,k,l} \to x'_{i,j,k,l+1}o\ ]^{0}$, $1 \le i \le m$, $0 \le k \le j \le n$;
- $[\ x_{i,j,k,3} \to x_{i,j,k-1,0}o\ ]^{e}$,
  $[\ x'_{i,j,k,3} \to x'_{i,j,k-1,0}o\ ]^{e}$, $e \in \{0,1\}$, $1 \le i \le m$, $1 \le k \le j \le n$;
- $[\ x_{i,j,0,3} \to oo\ ]^{0}$,
  $[\ x_{i,j,0,3} \to c_{i,j,0}o\ ]^{1}$,
  $[\ x'_{i,j,0,3} \to c_{i,j,0}o\ ]^{0}$,
  $[\ x'_{i,j,0,3} \to oo\ ]^{1}$, $1 \le i \le m$, $1 \le j \le n$;
- $[\ c_{i,j,l} \to c_{i,j,l+1}o\ ]^{0}$, $0 \le l \le 2$,
  $[\ c_{i,j,3} \to c_{i,j+1,0}o\ ]^{e}$, $e \in \{0,1\}$, $1 \le i \le m$, $1 \le j < n$.

Now let us consider step $4i-1$ of the generation phase: Two steps after the application of the rule

$$[\ d_{i,0}\ ] \to [\ d_{i,1}\ ][\ d'_{i,1}\ ],$$

one of the resulting membranes carries polarization 0, corresponding to assigning the truth value `false` to x_i (and in this case the clauses where $\neg x_i$ appears are satisfied), and the other membrane carries polarization 1, corresponding to assigning the truth value `true` to x_i (and in this case those clauses where x_i appears without negation are satisfied). Most important for the correct answer to the decision problem is the application of the rules

$$[\ x_{i,j,0,3} \to oo\ ]^{0}, [\ x_{i,j,0,3} \to c_{i,j,0}o\ ]^{1}, [\ x'_{i,j,0} \to c_{i,j,0}o\ ]^{0}, [\ x'_{i,j,0,3} \to oo\ ]^{1},$$

which in the corresponding step of the derivation act according to the truth value assigned to x_i in the underlying elementary membrane, i.e., only those variables "survive" which correspond to the correct truth assignment at the moment the last index has reached the ground level 0.

After the end of this first phase of the algorithm, 2^n elementary membranes (each of them with label 2) have been produced, each of them containing $d_{n,0}$ and objects $c_{i,n,0}$ for all clauses C_i that are satisfied. This procedure described so far in total takes $4n$ steps.

Transition phase

- $[\ d_n \to e_{1,0}\ ]^0$,
  $[\ c_{i,n,0} \to c_{i,0}\ ]^0$, $1 \le i \le n$.

After this single step of the transition phase all the elementary membranes now have the polarization 0 and contain $e_{1,0}$ as well as $c_{i,0}$ for each satisfied clause C_i.

Checking phase

- $[\ c_{1,0} \to zz\ ]^0$,
- $[\ c_{i,0} \to c_{i,1}o\ ]^1$,
  $[\ c_{i,1} \to c_{i,2}o\ ]^0$,
  $[\ c_{i,2} \to c_{i,1}o\ ]^1$,
  $[\ c_{i,2} \to c_{i+1,0}o\ ]^0$, $1 < i \le m$;
- $[\ e_{i,0} \to e_{i,1}o\ ]^1$,
  $[\ e_{i,1} \to e_{i,2}o\ ]^0$,
  $[\ e_{i,2} \to e_{i,1}o\ ]^1$,
  $[\ e_{i,2} \to e_{i+1,0}o\ ]^0$, $1 \le i \le m$.

All clauses are satisfied if and only if all objects $c_1, \cdots, c_m$ are present in some membrane. There are n cycles, each removing objects $c_{1,0}$ and decrementing by one the first index of all objects $c_{i,0}$, $1 < i \le n$. Each cycle is activated by $c_{1,0}$ producing symbols z that change the polarization. Thus, if in the beginning of some cycle $c_{1,0}$ is not present, then the objects in the corresponding membrane do not evolve any more.

Otherwise, $k > 0$ copies of $c_{1,0}$ lead to $2k$ steps of changing the polarization between 0 and 1. Finally, the polarization becomes 0 and remains so, and then the objects $e_{i,2}$ and $c_{i,2}$ "notice" this and the next cycle begins. The whole cycle takes $2k + 3$ steps.

If all clauses are satisfied, then the membrane will only contain object $e_{m+1,0}$. In total, this phase takes at most $3m + 2mn$ steps.

Output phase

- $[\ e_{m+1,0} \to \mathtt{yes}o\ ]^0$,
  $[\ \mathtt{yes}\ ]^0 \to [\ \ ]^1\mathtt{yes}$;
- $[\ t_{2mn+4n+3m+4} \to \mathtt{no}o\ ]^0$,
  $[\ \mathtt{no}\ ] \to [\ \ ]^{\neg}\mathtt{no}$.

Every elementary membrane which after the first $4n$ steps had represented a solution of the given satisfiability problem, after at most $(4n)+1+(3m+2mn)+2 = 2mn + 4n + 3m + 4$ steps has sent a copy of `yes` into the skin membrane, and, when the first copy of `yes` arrives in the skin, one copy of these copies exits into the environment, thus giving the positive result `yes` and changing the skin polarization to 1 in order to prevent further output. If, on the other hand, the given satisfiability problem has no solution, after $2mn + 4n + 3m + 4$ steps the polarization of the skin membrane still will be 0, hence, the object `no` is produced and sent out as the correct answer.

Due to the explanations given above one can easily verify that in any case the given algorithm will correctly decide a given satisfiability problem in n variables and m clauses in at most $2mn+4n+3m+6$ steps, i.e., the algorithm has time complexity $O(nm)$. This observation completes the proof. □

3.3 Remarks and other variants

Some definitions of decisional P systems require that the result is ejected into the environment only in the last step of the computation. Our construction can be easily adjusted to fulfill this property by remembering, in the objects $e_{i,j}$, also the number l of times the number of steps the membrane had polarization 1 during the checking phase, and then "keeping them busy" for $2(mn - l)$ steps. Then, all elementary membranes with positive answers will stop evolving at the same time by sending `yes` into the skin, and those with negative answers will stop evolving earlier.

In the construction given above, the time for giving a positive answer is actually bounded by $2K+4n+3m+4$, where K is the number of occurrences of the variables in β. Thus, if the size of the problem is given as (n, m, K), then (adjusting the counter in the skin) the time can be made at most $2K + 4n + 3m + 6$.

On the other hand, we do not believe that the rule sending object `yes` in the skin can be made independent of the polarization; otherwise, multiple answers are given and the halting time is no longer polynomial. This can easily be avoided for the price of using membrane dissolution (rules of type (d_{gp})) and one more membrane: a copy of the "witness" of the positive result dissolves the middle membrane, releasing a unique object `yes` into the skin, otherwise object `no` is ejected to the skin, as it was done in the proof of Theorem 9 in [3].

Finally, we mention alternative variants of restrictions:

Using the generation phase similar to that from the proof of Theorem 1 and making relevant adjustments to the global control, one can quite easily replace the rules of type (e_{gp0}) by rules of type (e_{gp2}).

By replacing the rule $[\ z\ ]_2 \to [\ \ ]_2^{\neg} z$ by $[\ z\ ] \to [\ \ ]^{\neg} o$, one can remove type (c_{np1}) for the price of introducing type (c_{gp1}).

Corollary 3. *For* $\mathbf{t} \in \{n, g\}$ *and* $\mathbf{k} \in \{0, 2\}$, $\mathtt{SAT}(n, m)$ *can be decided in linear time (linear with respect to* nm*) by a uniform family of P systems with active membranes with two polarizations and rules of the forms* (a_{gb}), $(c_{\mathbf{t}p1})$, *and* $(e_{gp\mathbf{k}})$.

4 Conclusions

In Theorem 2 we have given an algorithm for deciding the NP-complete decision problem $\mathtt{SAT}(n, m)$ by a uniform family of P system with active membranes in linear time (linear with respect to nm) with only two polarizations and rules of types (a), (c), and (e), of specific restrictive types. Various other restrictions are summarized in Corollary 3, and the discussion is given in Subsection 3.3.

The question remains whether further or other restrictions, respectively, of the general form of these rules are possible. For instance, can the problem be solved using only rules of types (a), (c_{p0}), (e) (the rules of type (c) do not depend on the polarization and preserve it)? What about using only types (a_p), (c), (e) (the rules of type (a) do not depend on the polarization)?

Another interesting question is to study systems with rules of types (a_u), (b), (c), (d), (e); such systems can only increase the number of objects via membrane division. What is their generative power? Are they efficient?

Acknowledgements

This paper was initiated during the Brainstorming Week on Membrane Computing taking place in Sevilla during the first week of February, 2004 (see [1] for more details).

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

References

[1] A. Alhazov, R. Freund, Gh. Păun: P systems with active membranes and two polarizations. In: Gh. Păun, A. Riscos-Núñez, A. Romero-Jiménez, F. Sancho-Caparrini (eds.): *Second Brainstorming Week on Membrane Computing*, TR 01/2004, University of Seville, Seville (2004) 20–36

[2] A. Alhazov, L. Pan: Polarizationless P systems with active membranes. *Grammars* **7**, 1 (2004)

[3] A. Alhazov, L. Pan, Gh. Păun: Trading polarizations for labels in P systems with active membranes. *Submitted* (2003)

[4] A. Alhazov, C. Martín-Vide, L. Pan: Linear time solutions of NP-complete graph problems by P systems with restricted elementary active membranes. *Submitted* (2003)

[5] A. Alhazov, C. Martín-Vide, L. Pan: Solving a PSPACE-complete problem by P systems with restricted active membranes. *Submitted* (2003)

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

[7] R. Freund, C. Martín-Vide, Gh. Păun: From regulated rewriting to computing with membranes: collapsing hierarchies, *Theoretical Computer Science* **312** (2004) 143–188

[8] R. Freund, Gh. Păun: Deterministic P systems. *Submitted* (2004)

[9] M. A. Gutierrez-Narajano, M.J. Pérez-Jiménez, A. Riscos-Núñez: Solving numerical NP-complete problems using P systems with active membranes: the partition problem and beyond. *EMCC Workshop,* Vienna (2003)

[10] S. N. Krishna, R. Rama: A variant of P systems with active membranes: solving NP-complete problems. *Romanian J. of Information Science and Technology* **2**, 4 (1999) 357–367

[11] M. Madhu, K. Krithivasan: Improved results about the universality of P systems. *Bulletin of the EATCS* **76** (February 2002) 162–168

[12] A. Obtułowicz: Deterministic P systems for solving SAT problem. *Romanian J. of Information Science and Technology* **4**, 1-2 (2001) 195–202

[13] A. Obtułowicz: On P systems with active membranes solving integer factorizing problem in a polynomial time. In: C.S. Calude, Gh. Păun, G. Rozenberg, A. Salomaa (eds.): Multiset Processing. Mathematical, Computer Science, and Molecular Computing Points of View. *Lecture Notes in Computer Science* **2235**, Springer-Verlag (2001) 267–286

[14] A. Păun: On P systems with global rules. In: N. Jonoska, N.C. Seeman (eds.): P*roc. 7th Intern. Meeting on DNA Based Computers.* Tampa (2001) 43–52

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

[16] Gh. Paun: Computing with Membranes - A Variant: P systems with polarized membranes. *Intern. J. of Foundations of Computer Science* **11**, 1 (2000) 167–182 and CDMTCS TR **098**, Univ. of Auckland (1999).

[17] M. Pérez-Jiménez, A. Romero Jiménez, F. Sancho Caparrini: *Teoria de la complejidad en modelos de computacion celular con membranas.* Kronos Editorial, Seville (2002)

[18] M. J. Pérez Jiménez, A. Romero Jiménez, F. Sancho Caparrini: Solving VALIDITY problem by active membranes with input. In: M. Cavaliere, C. Martín-Vide, Gh. Păun (eds.): *Rovira i Virgili Univ. Tech. Rep.* **26/03**, Brainstorming Week on Membrane Computing, Tarragona (2003) 279–290

[19] A. Salomaa, G. Rozenberg (eds.): Handbook of Formal Languages. Springer-Verlag, Berlin (1997)

[20] P. Sosík: Solving a PSPACE-complete problem by P systems with active membranes. In: M. Cavaliere, C. Martín-Vide, Gh. Păun (eds.): *Rovira i Virgili Univ. Tech. Rep.* **26/03**, Brainstorming Week on Membrane Computing, Tarragona (2003) 305–312

[21] Cl. Zandron, Cl. Ferretti, G. Mauri: Solving NP-complete problems using P systems with active membranes. In: I. Antoniou, C.S. Calude, M.J. Dinneen (eds.): *Unconventional Models of Computation.* Springer-Verlag, London (2000) 289–301