**ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY**
Volume **18**, Number 3, 2015, 201–217

# Recognizer P Systems with Antimatter

Daniel DÍAZ-PERNIL[1], Artiom ALHAZOV[2], Rudolf FREUND[3], Miguel A. GUTIÉRREZ-NARANJO[4], Alberto LEPORATI[5]

[1]Research Group on Computational Topology and Applied Mathematics
Department of Applied Mathematics - University of Sevilla, 41012, Spain
E-mail: sbdani@us.es

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

[3]Faculty of Informatics, TU Wien
Favoritenstraße 9-11, 1040 Vienna, Austria
E-mail: rudi@emcc.at

[4]Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence, University of Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: magutier@us.es

[5]Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano-Bicocca, Italy
E-mail: alberto.leporati@unimib.it

**Abstract.** In this paper, we consider recognizer P systems with antimatter and the influence of the matter/antimatter annihilation rules having weak priority over all the other rules or not. We first provide a uniform family of P systems with active membranes which solves the strongly **NP**-complete problem SAT, the Satisfiability Problem, without polarizations and without dissolution, yet with division for elementary membranes and with matter/antimatter annihilation rules having weak priority over all the other rules. Then we show that without this weak priority of the matter/antimatter annihilation rules over all the other rules we only obtain the complexity class **P**.

## 1. Introduction and preliminary results

In [11], a solution of the Subset Sum problem in the polynomial complexity class of recognizer P systems with active membranes without polarizations, without dissolution and with division for elementary membranes endowed with antimatter and matter/antimatter annihilation rules, having weak priority over all the other rules, was provided. Then in [10] even the strongly **NP**-complete problem SAT was shown to be solvable by such recognizer P systems. On the other hand, in [9] it was proved that recognizer P systems of that kind, yet without matter/antimatter annihilation rules having weak priority over all the other rules characterize exactly the complexity class **P**. In this way, antimatter has been shown to be a frontier of tractability in membrane computing. In this paper, we combine the results from [9] and [10] to show in which way antimatter and matter/antimatter annihilation rules constitute a frontier of tractability depending on the role of the matter/antimatter annihilation rules, i.e., depending on them having the feature of weak priority or not.

The Subset Sum problem belongs to the class of so-called *weakly* **NP**-complete problems, since its intractability strongly depends on the fact that extremely large input numbers are allowed [12]. The reason for this *weakness* is based on the encoding scheme of the input, since every integer in the input denoting a weight $w_i$ should be encoded by a string of length $O(\log w_i)$.

On the other hand, *strongly* **NP**-complete problems are those which remain **NP**-complete even if the data are encoded in a unary way. The best-known one of these problems is the satisfiability problem (SAT for short). SAT was the first problem shown to be **NP**-complete, as proved by Stephen Cook at the University of Toronto in 1971, see [7], and it has been widely used in membrane computing to prove the ability of a P system model to solve **NP**-complete problems, for example, see [13, 15, 16, 18, 21, 22].

In this paper, following the proof given in [10] we provide a solution to the SAT problem in the polynomial complexity class of recognizer P systems with active membranes without polarizations and without dissolution, yet with division for elementary membranes as well as endowed with antimatter and matter/antimatter annihilation rules, those having weak priority over all the other rules. The details of the implementation may also provide new tools for a better understanding of the problem of searching new frontiers of tractability in membrane computing. Then, based on the arguments elaborated in [9], we show that the polynomial complexity class of recognizer P systems with active membranes without polarizations and without dissolution, but with division for elementary membranes as well as with antimatter and matter/antimatter annihilation rules, yet those not having weak priority over all the other rules, exactly characterize the complexity class **P**.

The paper is organized as follows. In Section 2 we discuss the results about P systems found in the literature on the power and the limitations of antimatter. A short overview on the relationship of model ingredients used in different solutions for solving computationally difficult problems by P systems with active membranes and the emerging computational power is given in Section 3. In Section 4, we recall the P systems model used in this paper, the main new ingredient being antimatter and

matter/antimatter annihilation rules as well as their semantics with respect to the matter/antimatter annihilation rules having weak priority over all the other rules or not. In Section 5, some basics on recognizer P systems are recalled, and in Section 6 our solution for the SAT problem is provided. Based on the arguments exhibited in [9], in Section 7 we show that for each problem in the complexity class **P** we can construct a uniform family of polynomial recognizer P systems with active membranes without polarizations and without dissolution, but with division for elementary membranes as well as with antimatter and matter/antimatter annihilation rules, yet those not having weak priority over all the other rules. The paper finishes with some conclusions and hints for future work.

## 2. Antimatter Overview

The concept of antimatter has been introduced in the framework of membrane computing as a control tool for the flow of spikes in spiking neural P systems, for example, see [24] and [19, 29, 30]. In this context, when one spike and one anti-spike appear in the same neuron, the annihilation occurs and both, spike and anti-spike, disappear. Antimatter and matter/antimatter annihilation rules later have been adapted to other models of membrane systems, and currently this is an active research area.

The concept of matter/antimatter annihilation rules in transitional P systems initially appeared in [2]. It turned out that combining annihilation rules, which are a specific form of cooperative erasing, with non-cooperative rules yields an elegant computationally complete model. Note that immediate annihilation precisely corresponds to *weak priority* of annihilation. It has been shown that this priority may be removed at the price of adding *one* catalyst. Then, it has also been shown that P systems with non-cooperative rules and matter/antimatter annihilation are computationally complete even in the deterministic case. A variant with annihilation generating energy was considered, too.

The work of [2] has been continued in [1]. In particular, the computational completeness results were generalized to computing vectors over $\mathbb{Z}$ instead of $\mathbb{N}$, as well as to computing languages, or even subsets of groups (as languages over symbols and anti-symbols). A number of universality results involving *small* computing devices was obtained in [3], in particular, a universal accepting P system with 53 rules, simulating a model called generalized counter automata introduced there for that purpose.

Besides being studied for computational completeness and universality results involving small computing devices, matter/antimatter annihilation rules have been considered in the model of P systems with active membranes, for instance, see [11]. Under the basic settings, *i.e.*, with weak priority of the matter/antimatter annihilation rules over all the other rules, uniform families of recognizer P systems with active membranes solve Subset Sum, a well-known weakly **NP**-complete problem, and in [10] even a solution for SAT, the famous *strongly* **NP**-complete problem, has been described. Recently it has been shown in [9] that without the weak priority of the matter/antimatter annihilation rules over all the other rules, only the complexity class **P** is characterized within the framework of recognizer P systems.

## 3. Computation Theory Remarks

A computation is a sequence of configurations which starts from an initial configuration. A configuration describes the current status of the computing machine; this may include instances of objects, instances of membranes, and any other entity bearing information. A computation step consists of transformations of symbols by applying specific kinds of rules. Clearly, computations using rules without cooperation of symbols are quite limited in power; for example, it is known that $E0L$-behavior (i.e., the parallel use of non-cooperating rules as in Lindenmayer systems) with standard halting yields $PsREG$ (*i.e.*, semi-linear sets), and accepting P systems are considerably more degenerate.

In this sense, interaction of symbols is a fundamental part of membrane computing, or of theoretical computer science in general. Various ways of interaction of symbols have been studied in membrane computing. For the models with active membranes, the most commonly studied ways are various rules changing polarizations (or even sometimes labels) and membrane dissolution rules. One object may engage such a rule, which would affect the *context* (polarization or label) of other objects in the same membrane, thus affecting the behavior of the latter, *e.g.*, in case of dissolution, such objects find themselves in the parent membrane, which usually has a different label.

In the literature on P systems with active membranes, normally only the rules with at most one object on the left side were studied. Since recently, the model with matter/antimatter annihilation rules, *e.g.*, see [1] and [3], have attracted the attention of researchers. It provides a form of *direct* object-object interaction, albeit in a rather restricted way (*i.e.*, by erasing a pair of objects that are in a bijective relation). Although it is known that non-cooperative P systems with antimatter are universal, studying their efficiency turned out to be an interesting line of research. So how does matter/antimatter annihilation compare to other ways of organizing interaction of objects?

First, all known solutions of **NP**-complete (or more difficult) problems in membrane computing rely on the possibility of P systems to obtain *exponential space* in polynomial time; note that object replication alone does not count as building exponential space, since an exponential number can be written, e.g., in binary, in polynomial space. Such a possibility to obtain exponential space in polynomial time is provided by either of membrane division rules, membrane separation rules, see [4, 23, 25], membrane creation rules, see [20], (or string replication rules, but string-objects lie outside of the scope of the current paper). In tissue P systems, one may apply a similar approach to cells instead of membranes.

Note that in case of cell-like P systems, membrane creation alone (unlike the other types of rules mentioned above) makes it also possible to construct a hierarchy of membranes, let us refer to it as *structured workspace*, which is used to solve **PSPACE**-complete problems. The structured workspace can be alternatively created by elementary membrane division plus non-elementary membrane division (plus membrane dissolution if we have no polarizations).

Besides creating workspace, to solve **NP**-complete problems we need to be able to

effectively use that workspace by making objects interact. For instance, it is known that even with membrane division, without polarizations and without dissolution, only problems in **P** may be solved. However, already with two polarizations (the smallest non-degenerate value) P systems can solve **NP**-complete problems. What can be done without polarizations?

One solution is to use the power of switching the context by membrane dissolution. Coupled with non-elementary division, a suitable membrane structure can be constructed so that the needed interactions can be performed solving **NP**-complete or even **PSPACE**-complete problems [6]. It is not difficult to realize that elementary and non-elementary division rules can be replaced by membrane creation rules, or elementary division rules can be replaced by separation rules.

Finally, an alternative way of interaction of objects considered in this paper following [2] is matter/antimatter annihilation. What are the strengths and the weaknesses of these ingredients (the weaker is a combination of ingredients, the stronger is the result, while sometimes weaker ingredients do not let us do what stronger ones can do)?

Using matter/antimatter annihilation makes it possible to carry out multiple simultaneous interactions (for example, the checking phase in our solution for SAT is constant-time instead of linear with respect to the number of clauses), and it is a direct object-object interaction.

The power of dissolution and polarizations is the possibility of mass action (not critical for studying computational efficiency within **PSPACE** as all multiplicities are bounded with respect to the problem size) by changing context.

Using non-elementary division lets us build structured workspace (probably necessary for **PSPACE** if membrane creation is not used instead of membrane division, unless $\mathbf{P^{PP}}$=**PSPACE**, see [17]), and change non-local context (*e.g.*, the label of the parent membrane).

In the present paper we focus on using antimatter and matter/antimatter annihilation rules and on the significantly bigger power coming up when letting these rules having weak priority over all the other rules.

## 4. The P Systems Model

In this paper, we use the usual rules of evolution, communication, and division of elementary and non-elementary membranes which are common in P systems with active membranes. The main novelty in the model is the use of antimatter and matter/antimatter annihilation rules.

Inspired by physics, we consider the annihilation of two objects $a$ and $b$ from the alphabet $O$ in a membrane with label $h$, with the annihilation rule for $a$ and $b$ written as $[\ ab \to \lambda\ ]_h$. The *meaning* of the rule follows the idea of annihilation: If $a$ and $b$ occur simultaneously in the same membrane, then both are consumed (disappear) and nothing is produced (denoted by the empty string $\lambda$). The object $b$ is called the *antiparticle* of $a$ and it is usually written $\overline{a}$ instead of $b$.

With respect to the semantics, let us recall that this rule must be applied as many

times as possible in each membrane, according to the maximal parallelism. Following the intuition from physics, if $a$ and $\overline{a}$ occur simultaneously in the same membrane $h$ and the annihilation rule $[\ a\overline{a} \to \lambda\ ]_h$ is defined, then it has to be applied, regardless any other option. In this sense, any annihilation rule has (weak) priority over all rules of the other types, see [11] and [10]. Yet we may also relax this condition of weak priority, for instance, see [9].

A P system with active membranes without polarizations, without dissolution and with division of elementary and non-elementary membranes and with annihilation rules is a cell-like P system with rules of the following kinds (following [5], we use subscript 0 for the rule type to represent the restriction that such a rule does not depend on the polarization and is not allowed to change it; if all rules have this subscript, this is equivalent to saying that the P system is without polarizations):

($a_0$) $[\ a \to u\ ]_h$ for $h \in H$, $a \in O$, $u \in O^*$ *(object evolution rule).* An object $a \in O$ belonging to a membrane $h$ evolves to a multiset represented by the string $u \in O^*$.

($b_0$) $a[\ \ ]_h \to [\ b\ ]_h$ for $h \in H$, $a, b \in O$ *(send-in rule).* An object $a$ from the region immediately outside a membrane labeled by $h$ is taken into this membrane, possibly being transformed into another object $b$.

($c_0$) $[\ a\ ]_h \to b[\ \ ]_h$ for $h \in H$, $a, b \in O$ *(send-out rule).* An object $a$ is sent out from a membrane labeled by $h$ to the region immediately outside, possibly being transformed into another object $b$.

($e_0$) $[\ a\ ]_h \to [\ b\ ]_h[\ c\ ]_h$ for $h \in H$, $a, b, c \in O$ *(division rule for elementary membranes).* An elementary membrane can be divided into two membranes with the same label, possibly transforming one original object $a$ into a different one in each of the new membranes.

($f_0$) $[\ [\ \ ]_{h_1}[\ \ ]_{h_2}\ ]_{h_0} \to [\ [\ \ ]_{h_1}\ ]_{h_0}[\ [\ \ ]_{h_2}\ ]_{h_0}$, for $h_0, h_1, h_2 \in H$ *(division rules for non-elementary membranes).* If the membrane with label $h_0$ contains other membranes than those with labels $h_1, h_2$, then such membranes and their contents are duplicated and placed in both new copies of the membrane $h_0$; all membranes and objects placed inside membranes $h_1$, $h_2$, as well as the objects from membrane $h_0$ placed outside membranes $h_1$ and $h_2$, are reproduced in the new copies of membrane $h_0$.

($g_0$) $[\ a\overline{a} \to \lambda\ ]_h$ for $h \in H$, $a, \overline{a} \in O$ *(matter/antimatter annihilation rule).* The pair of objects $a, \overline{a} \in O$ belonging simultaneously to a membrane labeled by $h$ disappears.

Let us remark that dissolution rules - type ($d_0$) - are not considered in this model.

These rules are applied according to the following principles (with the special restrictions for annihilation rules specified above):

- All the rules are applied in parallel and in a maximal manner. In one step, one object of a membrane can be used by at most one rule (chosen in a non–deterministic way), and each membrane can be the subject of *at most one* rule of types $(b_0)$, $(c_0)$, $(e_0)$, and $(f_0)$.

- If at the same time a membrane labeled with $h$ is divided by a rule of type $(e_0)$ triggered by some object $a$ and there are other objects in this membrane to which rules of type $(a_0)$ or $(g_0)$ can be applied, then we suppose that first the rules of type $(g_0)$ and only then those of type $(a_0)$ are used, before finally the division is executed. This process in total takes only one step.

- The rules associated with membranes labeled by $h$ are used for all copies of membranes with label $h$.

## 5. Recognizer P Systems

Recognizer P systems are a well-known model of P systems which are basic for the study of complexity aspects in membrane computing. Next, we briefly recall some basic ideas related to them. For a detailed description, for example, see [26, 27]. In recognizer P systems all computations halt; there are two distinguished objects traditionally called *yes* and *no* (used to signal the result of the computation), and exactly one of these objects is sent out to the environment (only) in the last computation step.

Let us recall that a decision problem $X$ is a pair $(I_X, \theta_X)$ where $I_X$ is a language over a finite alphabet (the elements are called *instances*) and $\theta_X$ is a predicate (a total Boolean function) over $I_X$. Let $X = (I_X, \theta_X)$ be a decision problem. A *polynomial encoding* of $X$ is a pair $(cod, s)$ of polynomial time computable functions over $I_X$ such that for each instance $w \in I_X$, $s(w)$ is a natural number representing the *size* of the instance and $cod(w)$ is a multiset representing an encoding of the instance. Polynomial encodings are stable under polynomial time reductions.

A family of P systems $\mathbf{\Pi}$ is said to be *sound* with regard to $X$ if for each instance of the problem $w \in I_X$, if there exists an accepting computation of $\Pi(w)$, then $\theta_X(w) = 1$, and $\mathbf{\Pi}$ is *complete* with regard to $X$ if for each instance of the problem $w \in I_X$, provided that $\theta_X(w) = 1$, then every computation of $\Pi(w)$ is an accepting computation.

Let $\mathcal{R}$ be a class of recognizer P systems with input membrane. A decision problem $X = (I_X, \theta_X)$ is solvable in a uniform way and polynomial time by a family $\mathbf{\Pi} = \{\Pi(n)\}_{n \in \mathbf{N}}$ of P systems from $\mathcal{R}$ – we denote this by $X \in \mathbf{PMC}_{\mathcal{R}}$ – if the family $\mathbf{\Pi}$ is polynomially uniform by Turing machines, *i.e.*, there exists a polynomial encoding $(cod, s)$ from $I_X$ to $\mathbf{\Pi}$ such that the family $\mathbf{\Pi}$ is polynomially bounded with regard to $(X, cod, s)$; this means that there exists a polynomial function $p$ such that for each $u \in I_X$ every computation of $\Pi(s(u))$ with input $cod(u)$ is halting and, moreover, it performs at most $p(|u|)$ steps; the family $\mathbf{\Pi}$ is sound and complete with regard to $(X, cod, s)$. Moreover, we write $X \in \mathbf{PMC}^{\star}_{\mathcal{R}}$ if a decision problem $X = (I_X, \theta_X)$ is

solvable only in a semi-uniform way and polynomial time by a family $\mathbf{\Pi}$, *i.e.*, for each instance of the problem $w \in I_X$ we need a recognizer P system on its own.

By following the standard notation, in [11] the class of polarizationless recognizer P systems with active membranes without dissolution and with division of elementary and non-elementary membranes, and with antimatter and matter/antimatter annihilation rules has been denoted by $\mathcal{AM}^0_{-d,+ne,+ant}$, without using a symbol in the name to specify the priority, as this was assumed to be part of the model definition. The class of P systems which uses the same model of P systems, but without priority for the application of the annihilation rules, as in [9], is denoted by $\mathcal{AM}^0_{-d,+ne,+ant\_NoPri}$. If division of non-elementary membranes is not used and only division of elementary membranes is used, we write $+e$ instead of $+ne$.

## 6. Solving SAT

By constructing a uniform family of deterministic recognizer P systems with active membranes, without polarizations, without non-elementary membrane division and without dissolution, yet with matter/antimatter annihilation rules, for solving SAT, in this section we show the following result:

**Theorem 1.** $\mathbf{NP} \subseteq \mathbf{PMC}_{\mathcal{AM}^0_{-d,+e,+ant}}$.

*Propositional Satisfiability* is the problem of determining, for a formula of the propositional calculus, if there is an assignment of truth values to its variables for which that formula evaluates to true. By SAT we mean the problem of propositional satisfiability for formulas in conjunctive normal form (CNF). In the following we describe a uniform family of P systems which solves it. As usual, we will address the resolution via a brute force algorithm, which consists of the following stages (some of the ideas for the design are taken from [8] and [28]):

- *Generation and Evaluation Stage:* All possible assignments associated with the formula are created and evaluated (in this paper we have subdivided this group into *Generation* and *Input processing* groups of rules, which take place in parallel).
- *Checking Stage:* In each membrane we check whether or not the formula evaluates to true for the assignment associated with it.
- *Output Stage:* The system sends out the correct answer to the environment.

Let us consider the pairing function $\langle\ ,\ \rangle$ defined by

$$\langle n, m\rangle = ((n+m)(n+m+1)/2) + n.$$

This function is polynomial-time computable (it is primitive recursive and bijective from $\mathbb{N}^2$ onto $\mathbb{N}$). For any given formula in CNF, $\varphi = C_1 \wedge \cdots \wedge C_m$, with $m$ clauses and $n$ variables $Var(\varphi) = \{x_1, \ldots, x_n\}$ we construct a P system $\Pi(\langle n, m\rangle)$ solving it,

where the multiset encoding the problem to be the input of $\Pi(\langle n, m\rangle)$ (for the sake of simplicity, in the following we will omit $m$ and $n$) is

$$cod(\varphi) = \{x_{i,j} : \ x_j \in C_i\} \cup \{y_{i,j} : \ \neg x_j \in C_i\}.$$

For solving SAT by a uniform family of deterministic recognizer P systems with active membranes, without polarizations, without non-elementary membrane division and without dissolution, yet with matter/antimatter annihilation rules, we now construct the members of this family as follows:

$$\begin{aligned}
\Pi &= (O, \Sigma, H = \{1,2\}, \mu = [\ [\ ]_2\ ]_1, w_1, w_2, R, i_{in} = 2), \text{ where}\\
\Sigma &= \{x_{i,j}, y_{i,j} \mid 1 \le i \le m,\ 1 \le j \le n\},\\
O &= \{d, t, f, F, \overline{F}, T, \overline{no}_{n+5}, \overline{F}_{n+5}, \overline{yes}_{n+6}, yes_{n+6}, no_{n+6}, yes, no\}\\
&\cup \{x_{i,j}, y_{i,j} \mid 1 \le i \le m,\ -1 \le j \le n\} \cup \{\overline{x}_{i,-1}, \overline{y}_{i,-1} \mid 1 \le i \le m\}\\
&\cup \{c_i, \overline{c}_i \mid 1 \le i \le m\} \cup \{e_j \mid 1 \le j \le n+3\}\\
&\cup \{yes_j, no_j, F_j \mid 0 \le j \le n+5\},\\
w_1 &= no_0\ yes_0\ F_0,\ w_2 = d^n\ e_1,
\end{aligned}$$

and the rules of the set $R$ are given below, presented in the groups Generation, Input Processing, Checking, and Output, together with explanations about how the rules in the groups work.

**Generation**

G1. $[\ d\ ]_2 \to [\ t\ ]_2[\ f\ ]_2$;

G2. $[\ t \to \overline{y}_{1,-1} \cdots \overline{y}_{m,-1}\ ]_2$;

G3. $[\ f \to \overline{x}_{1,-1} \cdots \overline{x}_{m,-1}\ ]_2$;

G4. $[\ \overline{x}_{i,-1} \to \lambda\ ]_2$, $1 \le i \le m$;

G5. $[\ \overline{y}_{i,-1} \to \lambda\ ]_2$, $1 \le i \le m$.

In each step $j$, $1 \le j \le n$, every elementary membrane is divided, one new membrane corresponding with assigning *true* to variable $j$ and the other one with assigning *false* to it. One step later, proper objects are produced to annihilate the input objects associated to variable $j$: in the *true* case, we introduce the antimatter object for the negated variable, i.e., it will annihilate the corresponding negated variable, and in the *false* case, we introduce the antimatter object for the variable itself, i.e., it will annihilate the corresponding variable. Remaining barred (antimatter) objects not having been annihilated with the input objects, are erased in the next step.

**Input Processing**

I1. $[\ x_{i,j} \to x_{i,j-1}\ ]_2$, $1 \le i \le m$, $0 \le j \le n$;

I2. $[\ y_{i,j} \to y_{i,j-1}\ ]_2$, $1 \le i \le m$, $0 \le j \le n$;

I3. $[\ x_{i,-1}\ \overline{x}_{i,-1} \to \lambda\ ]_2$, $1 \leq i \leq m$;

I4. $[\ y_{i,-1}\ \overline{y}_{i,-1} \to \lambda\ ]_2$, $1 \leq i \leq m$;

I5. $[\ x_{i,-1} \to c_i\ ]_2$, $1 \leq i \leq m$;

I6. $[\ y_{i,-1} \to c_i\ ]_2$, $1 \leq i \leq m$.

Input objects associated with variable $j$ decrement their second subscript during $j+1$ steps to $-1$. The variables not representing the desired truth value are eliminated by the corresponding antimatter object generated by the rules G2 and G3, whereas any of the input variables not annihilated then, shows that the associated clause $i$ is satisfied, which situation is represented by the introduction of the object $c_i$.

**Checking**

C1. $[\ e_j \to e_{j+1}\ ]_2$, $1 \leq j \leq n+1$;

C2. $[\ e_{n+2} \to \overline{c}_1 \cdots \overline{c}_m e_{n+3}\ ]_2$;

C3. $[\ c_i\ \overline{c}_i \to \lambda\ ]_2$, $1 \leq i \leq m$;

C4. $[\ \overline{c}_i \to F\ ]_2$, $1 \leq i \leq m$;

C5. $[\ e_{n+3} \to \overline{F}\ ]_2$;

C6. $[\ F\ \overline{F} \to \lambda\ ]_2$, $1 \leq i \leq m$;

C7. $[\ \overline{F}\ ]_2 \to [\ \ ]_2 T$.

It takes $n+2$ steps to produce objects $c_i$ for every satisfied clause, possibly multiple times. Starting from object $e_1$, we have obtained the object $e_{n+2}$ until then; from this object $e_{n+2}$, at step $n+2$ one anti-object is produced for each clause. If any of these clause anti-objects $\overline{c}_i$ is not annihilated, then it is transformed into $F$, showing that the chosen variable assignment did not satisfy the corresponding clause. It remains to notice that object $T$ is sent to the skin (at step $n+4$) if and only if an object $\overline{F}$ did not get annihilated, i.e., no clause failed to be satisfied.

**Output**

O1. $[\ yes_j \to yes_{j+1}\ ]_1$, $0 \leq j \leq n+5$;

O2. $[\ no_j \to no_{j+1}\ ]_1$, $0 \leq j \leq n+5$;

O3. $[\ F_j \to F_{j+1}\ ]_1$, $0 \leq j \leq n+4$;

O4. $[\ T \to \overline{no}_{n+5} \overline{F}_{n+5}\ ]_1$;

O5. $[\ no_{n+5}\ \overline{no}_{n+5} \to \lambda\ ]_1$;

O6. $[\ no_{n+6}\ ]_1 \to [\ \ ]_1 no$;

O7. $[\ F_{n+5}\ \overline{F}_{n+5} \to \lambda\ ]_1$;

O8. $[\ F_{n+5} \to \overline{yes}_{n+6}\ ]_1$;

O9. $[\ yes_{n+6}\ \overline{yes}_{n+6} \to \lambda\ ]_1$;

O10. $[\ yes_{n+6}\ ]_1 \to [\ \ ]_1 yes$.

If no object $T$ has been sent to the skin, then the initial *no*-object can count up to $n+6$ and then send out the negative answer *no*, while the initial object $F$ counts up to $n+5$, generates the antimatter object for the *yes*-object at stage $n+6$ and annihilates with the corresponding object *yes* at stage $n+6$. On the other hand, if (at least one) object $T$ arrives in the skin, then the object *no* is annihilated at stage $n+5$ before it would be sent out in the next step, and the object $F$ is annihilated before it could annihilate with the object *yes*, so that the positive answer *yes* can be sent out in step $n+6$.

Finally, we notice that the solution is uniform, deterministic, and uses only rules of types $(a_0)$, $(c_0)$, $(e_0)$ as well as matter/antimatter annihilation rules. The result is produced in $n+6$ steps.

## 7. Characterizing P by Removing the Priority for the Annihilation Rules

In this section, rules of type $(g_0)$ (matter/antimatter annihilation rules) have no priority over all the other types of rules. If at the same time a membrane labelled with $h$ is divided by a rule of type $(e_0)$ or $(f_0)$ and there are objects in this membrane which are chosen to be annihilated by means of rules of type $(g_0)$, then we assume that first the annihilation is performed and then the division is produced. Of course, this process takes only one step. On the other hand, we now even allow non-elementary membrane division, and still we can show that we cannot go beyond the deterministic polynomial class **P**.

The main result elaborated in this section is the following claim; its proof follows the one already exhibited in [9].

**Theorem 2.** $\mathbf{PMC}_{\mathcal{AM}^0_{-d,+ne,+ant\_NoPri}} = \mathbf{P}$.

*Proof.* It is well known, for example, see [14], that

$$\mathbf{PMC}_{\mathcal{AM}^0_{-d,+ne}} = \mathbf{PMC}^{\star}_{\mathcal{AM}^0_{-d,+ne}} = \mathbf{P}.$$

On the other hand, the following inclusion obviously holds:

$$\mathbf{PMC}_{\mathcal{AM}^0_{-d,+ne}} \subseteq \mathbf{PMC}_{\mathcal{AM}^0_{-d,+ne,+ant\_NoPri}},$$

therefore

$$\mathbf{P} \subseteq \mathbf{PMC}_{\mathcal{AM}^0_{-d,+ne,+ant\_NoPri}}.$$

Thus it only remains to prove that also the converse inclusion holds, *i.e.*, that

$$\mathbf{PMC}_{\mathcal{AM}^0_{-d,+ne,+ant\_NoPri}} \subseteq \mathbf{P}.$$

In order to prove this inclusion, since $\mathbf{PMC}^{\star}_{\mathcal{AM}^0_{-d,+ne}} = \mathbf{P}$, it suffices to prove that

$$\mathbf{PMC}_{\mathcal{AM}^0_{-d,+ne,+ant\_NoPri}} \subseteq \mathbf{PMC}^{\star}_{\mathcal{AM}^0_{-d,+ne}}.$$

Hence, let $X \in \mathbf{PMC}_{\mathcal{AM}^0_{-d,+ne,+ant\_NoPri}}$ be a decision problem. By definition, there exist a polynomial encoding $(cod, s)$ and a family of P systems $\{\Pi(i)\}_{i\in\mathbb{N}}$ in $\mathcal{AM}^0_{-d,+ne,+ant\_NoPri}$ such that for each instance $u$ of the problem $X$:

- all computations of $\Pi(s(u))$ on input $cod(u)$ halt;
- in all computations, the system sends out either one copy of the object *yes* or one copy of the object *no* (but not both), and only in the last step of computation.

Let us first provide an informal idea of the proof. Given an instance $u \in I_X$, we know that all computations of $\Pi(s(u))$ on input $cod(u)$ halt, and that they all answer *yes* or all answer *no*. Let $\mathcal{C} = \{C_0, \ldots, C_n\}$ be one of these halting computations, and let us assume that the answer is *yes* (the other case is analogous). Then there exists an object $a_1$ and a rule $r_1 \equiv [\, a_1 \,]_{skin} \to yes\, [\,]_{skin}$ which has been applied in the last step of the computation. There are two possibilities: either object $a_1$ is in the skin membrane since the beginning of the computation, or there exists a rule $r_2$ which must have produced it inside or moved it into the skin membrane. Note that there may be several rules which produce object $a_1$ in the skin membrane; in such a case, just pick one of them as $r_2$. Rule $r_2$ is triggered by the occurrence of an object $a_2$ in a membrane with label $h_2$. Obviously, $r_2$ cannot be an annihilation rule, since no object is produced by such rules, then rule $r_2$ must belong to types $(a_0)$ to $(e_0)$. Going back with the reasoning, either $a_2$ appears in the membrane with label $h_2$ since the beginning of the computation, or it is produced or moved there by the application of a rule $r_3$, and so on.

Finally we have a chain

$$(yes, env) \xleftarrow{r_1} (a_1, skin) \xleftarrow{r_2} (a_2, h_2) \xleftarrow{r_3} \cdots \xleftarrow{r_k} (a_k, h_k)$$

where $k \leq n$ and $a_k$ appears in a membrane with label $h_k$ in the initial configuration (possibly as part of the input multiset). The key idea here is two-folded. On the one hand, annihilation rules do not produce any object; the objects that trigger an annihilation rule disappear and nothing is produced. On the other hand, for any halting configuration there *must* exist a finite sequence of rules $(r_k, r_{k-1}, \ldots, r_2, r_1)$ where $r_k$ is triggered by an object from the initial configuration, $r_1$ produces *yes* and each $r_i$ produces an object that triggers $r_{i-1}$. Therefore, none of rules $r_1, \ldots, r_k$ is an annihilation rule.

To formally prove the result we have to check that the amount of resources for finding the sequence of rules is polynomially bounded. With this aim, we will start by considering the dependency graph associated with $\Pi(s(u))$, but considering only

evolution, communication and division rules, *i.e.*, only rules which can produce new occurrences of objects (see [14] for the details about polynomial resources).

Now, if $R$ is the set of rules associated with $\Pi(s(u))$, we consider the corresponding directed graph $G = (V, E)$ defined as follows, where the function $f : H \to H$ returns the label of the parent membrane:

$V = VL \cup VR$,

$$\begin{aligned} VL = \quad \{(a,h) \in \Gamma \times H : \quad & \exists u \in \Gamma^* \ ([a \to u]_h \in R) \vee \\ & \exists b \in \Gamma \ ([a]_h \to [\ ]_h b \in R) \vee \\ & \exists b \in \Gamma \ \exists h' \in H \ (h = f(h') \wedge a[\ ]_{h'} \to [b]_{h'} \in R) \vee \\ & \exists b, c \in \Gamma \ ([a]_h \to [b]_h [c]_h \in R)\}, \end{aligned}$$

$$\begin{aligned} VR = \{(b,h) \in \Gamma \times H : \quad & \exists a \in \Gamma \ \exists u \in \Gamma^* \ ([a \to u]_h \in R \wedge b \in u) \vee \\ & \exists a \in \Gamma \ \exists h' \in H \ (h = f(h') \wedge [a]_{h'} \to [\ ]_{h'} b \in R) \vee \\ & \exists a \in \Gamma \ (a[\ ]_h \to [b]_h \in R) \vee \\ & \exists a, c \in \Gamma \ ([a]_h \to [b]_h [c]_h \in R)\}, \end{aligned}$$

$$\begin{aligned} E = \{((a,h),(b,h')) : \quad & \exists u \in \Gamma^* \ ([a \to u]_h \in R \wedge b \in u \wedge h = h') \vee \\ & ([a]_h \to [\ ]_h b \in R \ \wedge \ h' = f(h)) \vee \\ & (a[\ ]_{h'} \to [b]_{h'} \in R \ \wedge h \ = f(h')) \vee \\ & \exists c \in \Gamma \ ([a]_h \to [b]_h [c]_h \in R \wedge h = h')\}. \end{aligned}$$

Such a dependency graph can be constructed by a Turing machine working in polynomial time with respect to the instance size (see [14]). Finally, let us consider the set

$$\begin{aligned} \Delta_\Pi = \{(a,h) \in \Gamma \times H : \quad & \text{there exists a path (within the dependency graph)} \\ & \text{from } (a,h) \text{ to } (yes, env)\}\,. \end{aligned}$$

It has also been proved that there exists a Turing machine that constructs $\Delta_\Pi$ in polynomial time; the proof uses the *Reachability Problem* in order to prove the polynomially bounded construction (again we refer to [14]).

From this construction we directly obtain that the set of rules used in the chain

$$(yes, env) \xleftarrow{r_1} (a_1, skin) \xleftarrow{r_2} (a_2, h_2) \xleftarrow{r_3} \cdots \xleftarrow{r_k} (a_k, h_k)$$

described above can be found in polynomial time.

Finally, for the instance $u \in I_X$, let us consider the P system $\Pi(u')$ with only one membrane with label $s$ and only one object $(a_k, h_k)$ in the initial configuration. The set of rules is

- $[(a_i, h_i) \to (a_{i-1}, h_{i-1})]_s$ for each $i \in \{3, \ldots, k-1\}$

– $[(a_2, h_2) \to (a_1, skin)]_s$

– $[(a_1, skin)]_s \to yes\,[\,]_s$

Moreover, we observe that a similar construction can be carried out for the answer *no*.

The system $\Pi(u')$ can be built in polynomial time by a deterministic Turing machine. A direct inspection of the rules shows that $\Pi(u') \in \mathcal{AM}^0_{-d,+ne}$ (in fact, we are not even using membrane division rules at all). The behavior of the system is deterministic, and it computes the correct answer for the instance $u \in I_X$, sending out the object *yes* to the environment in the last step of computation.

Notice that *any* multiplicity of object $a$ in a membrane $h$ is represented by *one* node $(a, h)$ in the dependency graph. Since the labels cannot be renamed and membranes cannot be dissolved, we can safely assume that each membrane label uniquely determines its parent's label and so on until the skin (e.g., by starting with all membranes having different labels). Recall, however, that the result of computation comes from moving/renaming *one* $(a, h)$ into $(yes, env)$ or $(no, env)$. Hence, there is no need to distinguish different instances of membranes with the same label. It follows that rules of type $(f_0)$ do not lead to any transitions in the dependency graph, hence, the result also holds with allowing for non-elementary membrane division rules.

In sum, we conclude that $X \in \mathbf{PMC}^{\star}_{\mathcal{AM}^0_{-d,+ne}} = \mathbf{P}$. □

**Remark 1.** Let us finally explain the idea how to even get a *uniform* family of recognizer P systems from the family constructed in the preceding proof by making some preprocessing: For any input of length $n$, we include all possible input symbols in the dependency graph. If there is a path from some symbol to *yes* and from another symbol to *no*, then by the definition of confluence, an input containing both of these symbols simultaneously cannot be a valid input. So, once we get an input of length $n$, we first check if it has symbols deriving *yes* and symbols deriving *no*. This certainly is possible within polynomial time.

## 8. Conclusions

In this paper we have considered polarizationless recognizer P systems with antimatter and annihilation rules, without dissolution, and with division of elementary (and non-elementary) membranes. We have proved that by removing priority in polarizationless recognizer P systems with antimatter and matter/antimatter annihilation rules, without dissolution, and with division of elementary (and non-elementary) membranes, we obtain a new characterization of the standard complexity class **P**. On the other hand, we have proved that polarizationless recognizer P systems without dissolution and with division of elementary membranes as well as with antimatter and matter/antimatter annihilation rules, those having weak priority over all the other types of rules, can solve the strongly **NP**-complete problem SAT. In total we have shown that this weak priority plays an important role in the computational power

of these recognizer P systems, which indeed is the most interesting aspect of our results. We thus have proved that the semantics of a model can be a useful tool for studying problems of tractability. To the best of our knowledge, this is the first time that two models of P systems syntactically identical were shown to correspond to two (presumably) different complexity classes simply because they use different semantics.

This opens a new research area in the study of tractability in membrane computing. Not only new ingredients or new models must be studied in order to find new frontiers: classical results can also be revisited in order to explore the consequences of considering alternative semantics.

Let us finally remark the important role of the definition for recognizer P systems we have used in this paper, as this definition is quite restrictive, since only one object *yes* or *no* is sent to the environment in any computation. In the literature one can find other definitions of recognizer P systems and therefore other definitions of what it means *to solve* a problem in the framework of membrane computing. Hence, the study of the complexity classes in membrane computing deserves further investigations under these specific definitions.

**Acknowledgements.** M.A. Gutiérrez-Naranjo acknowledges the support of the project TIN2012-37434 of the Ministerio de Economía y Competitividad of Spain. A. Leporati acknowledges the support of Fondo d'Ateneo 2014, "Complessità Computazionale nei Sistemi a Membrane", of the Università degli Studi di Milano-Bicocca.

# References

[1] ALHAZOV A., AMAN B., FREUND R., *P systems with anti-matter*, in: M. Gheorghe, G. Rozenberg, A. Salomaa, P. Sosík, C. Zandron (Eds.): *Membrane Computing - 15th International Conference, CMC 2014, Prague, 2014, Revised Selected Papers*. Lecture Notes in Computer Science **8961**, Springer, 2014, pp. 66–85.

[2] ALHAZOV A., AMAN B., FREUND R., PĂUN GH., *Matter and anti-matter in membrane systems*, in: L.F. Macías-Ramos, M.A. Martínez-del-Amor, Gh. Păun, A. Riscos-Núñez, L. Valencia-Cabrera (Eds.): *Twelfth Brainstorming Week on Membrane Computing*. RGNC report 1/2014, University of Seville, Fénix Editora, Sevilla, 2014, pp. 1–26.

[3] ALHAZOV A., AMAN B., FREUND R., PĂUN GH., *Matter and anti-matter in membrane systems*, in: H. Jürgensen, J. Karhumäki, A. Okhotin (Eds.): *Descriptional Complexity of Formal Systems 2014*. Lecture Notes in Computer Science **8614**, Springer, 2014, pp. 65–76.

[4] ALHAZOV A., ISHDORJ TS.-O., *Membrane operations in P systems with active membranes*, in: Gh. Păun, A. Riscos-Núñez, A. Romero-Jimenez, F. Sancho-Caparrini (Eds.): *Second Brainstorming Week on Membrane Computing*. RGNC report 01/2004, University of Seville, Sevilla, 2004, pp. 37–44.

[5] ALHAZOV A., PAN L., PĂUN GH., *Trading polarizations for labels in P systems with active membranes*, Acta Informatica **41** (2-3), pp. 111–144 (2004).

[6] ALHAZOV A., PÉREZ-JIMÉNEZ M.J., *Uniform solution of* QSAT *using polarizationless active membranes*, in: J. Durand-Lose, M. Margenstern (Eds.): *5th International*

*Conference, MCU 2007, Orléans, France, September 10-13, 2007. Proceedings*. Lecture Notes in Computer Science **4664**, Springer, 2007, pp. 122–133.

[7] COOK S.A., *The complexity of theorem-proving procedures*, in: *Proceedings of the Third Annual ACM Symposium on Theory of Computing, STOC '71*. ACM, New York, NY, 1971, pp. 151–158.

[8] CORDÓN-FRANCO A., GUTIÉRREZ-NARANJO M.A., PÉREZ-JIMÉNEZ M.J., SANCHO-CAPARRINI F., *A Prolog simulator for deterministic P systems with active membranes*, New Generation Computing **22** (4), pp. 349–363 (2004).

[9] DÍAZ-PERNIL D., FREUND R., GUTIÉRREZ-NARANJO M.A., LEPORATI A., *On the semantics of annihilation rules in membrane computing*, in: J. Sempere et al. (Eds.): *Proceedings of the 16th International Conference on Membrane Computing, CMC 2015*, 2015, pp. 91–101.

[10] DÍAZ-PERNIL D., ALHAZOV A., FREUND R., GUTIÉRREZ-NARANJO M.A., *Solving* SAT *with Antimatter in Membrane Computing*, in: Michael J. Dinneen (Ed.): *Proceedings of the Workshop on Membrane Computing 2015 (WMC2015)*, CDMTCS Research Report Series CDMTCS-487 (Satellite workshop of UCNC2015), University of Auckland, Auckland, New Zealand, pp. 48–58.

[11] DÍAZ-PERNIL D., PEÑA-CANTILLANA F., ALHAZOV A., FREUND R., GUTIÉRREZ-NARANJO M.A., *Antimatter as a frontier of tractability in membrane computing*, Fundamenta Informaticae **134** (1-2), pp. 83–96 (2014).

[12] GAREY M.R., JOHNSON D.S., *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, 1979.

[13] GAZDAG ZS., KOLONITS G., *A new approach for solving* SAT *by P systems with active membranes*, in: E. Csuhaj-Varjú, M. Gheorghe, G. Rozenberg, A. Salomaa, Gy. Vaszil (Eds.): *International Conference on Membrane Computing. 13th International Conference, CMC 2012, Budapest, Hungary, August 28-31, 2012, Revised Selected Papers*. Lecture Notes in Computer Science **7762**, Springer, 2012, pp. 195–207.

[14] GUTIÉRREZ-NARANJO M.A., PÉREZ-JIMÉNEZ M.J., RISCOS-NÚÑEZ A., ROMERO-CAMPERO F.J., *On the power of dissolution in P systems with active membranes*, In: R. Freund, Gh. Păun, G. Rozenberg, A. Salomaa (Eds.): *Workshop on Membrane Computing. 6th International Workshop, WMC 2005, Vienna, Austria, July 18-21, 2005, Revised Selected and Invited Papers*. Lecture Notes in Computer Science **3850**, Springer, 2005, pp. 224–240.

[15] GUTIÉRREZ-NARANJO M.A., PÉREZ-JIMÉNEZ M.J., ROMERO-CAMPERO F.J., *A uniform solution to* SAT *using membrane creation*, Theoretical Computer Science **371** (1-2), pp. 54–61 (2007).

[16] ISHDORJ TS.-O., LEPORATI A., *Uniform solutions to* SAT *and 3-*SAT *by spiking neural P systems with pre-computed resources*, Natural Computing **7** (4), pp. 519–534 (2008).

[17] LEPORATI A., MANZONI L., MAURI G., PORRECA A.E., ZANDRON C., *Simulating elementary active membranes – with an application to the P conjecture*, in: M. Gheorghe, G. Rozenberg, A. Salomaa, P. Sosík, C. Zandron (Eds.): *Conference on Membrane Computing*. Lecture Notes in Computer Science **8961**, Springer, 2014, pp. 284–299.

[18] LEPORATI A., MAURI G., ZANDRON C., PĂUN GH., PÉREZ-JIMÉNEZ M.J., *Uniform solutions to* SAT *and subset sum by spiking neural P systems*, Natural Computing **8** (4), pp. 681–702 (2009).

[19] METTA V.P., KRITHIVASAN K., GARG D., *Computability of spiking neural P systems with anti-spikes*, New Mathematics and Natural Computation **8** (3), pp. 283–295 (2012).

[20] MUTYAM M., KRITHIVASAN K., *P systems with membrane creation: universality and efficiency*, in: M. Margenstern, Yu. Rogozhin (Eds.): Proceedings of *Machines, Computations, and Universality, MCU 2001, Chişinău, 2001.* Lecture Notes in Computer Science **2055**, Springer, 2001, pp. 276–287.

[21] OBTUŁOWICZ A., *Deterministic P systems for solving* SAT *problem*, Romanian Journal of Information Science and Technology **4** (1-2), pp. 195–201 (2001).

[22] PAN L., ALHAZOV A., *Solving HPP and SAT by P systems with active membranes and separation rules*, Acta Informatica **43** (2), pp. 131–145 (2006).

[23] PAN L., ISHDORJ TS.-O., *P systems with active membranes and separation rules*, Journal of Universal Computer Science **10** (5), pp. 630–649 (2004).

[24] PAN L., PĂUN GH., *Spiking neural P systems with anti-spikes*, International Journal of Computers, Communications & Control **IV** (3), pp. 273–282 (2009).

[25] PAN L., PÉREZ-JIMÉNEZ M.J., *Computational complexity of tissue-like P systems*, Journal of Complexity **26** (3), pp. 296–315 (2010).

[26] PÉREZ-JIMÉNEZ M.J., *An approach to computational complexity in membrane computing*, in: G. Mauri, Gh. Păun, M.J. Pérez-Jiménez, G. Rozenberg, A. Salomaa (Eds.): *Workshop on Membrane Computing. 5th International Workshop, WMC 2004, Milan, Italy, June 14-16, 2004, Revised Selected and Invited Papers.* Lecture Notes in Computer Science **3365**, Springer, 2004, pp. 85–109.

[27] PÉREZ-JIMÉNEZ M.J., RISCOS-NÚÑEZ A., ROMERO-JIMÉNEZ Á., WOODS D., *Complexity – membrane division, membrane creation*, in: Gh. Păun, G. Rozenberg, A. Salomaa (Eds.): *The Oxford Handbook of Membrane Computing*, Oxford University Press, 2010, pp. 302–336.

[28] PÉREZ-JIMÉNEZ M.J., ROMERO-JIMÉNEZ Á., SANCHO-CAPARRINI F., *Complexity classes in models of cellular computing with membranes*, Natural Computing **2** (3), pp. 265–285 (2003).

[29] SONG T., JIANG Y., SHI XI., ZENG XI., *Small universal spiking neural P systems with anti-spikes*, Journal of Computational and Theoretical Nanoscience **10** (4), pp. 999–1006 (2013).

[30] TAN G., SONG T., CHEN ZH., ZENG XI., *Spiking neural P systems with anti-spikes and without annihilating priority working in a 'flip-flop' way*, International Journal of Computing Science and Mathematics **4** (2), pp. 152–162 (2013).