Minimal cooperation in polarizationless P systems with active membranes

Luis Valencia-Cabrera, David Orellana-Martín,
Agustín Riscos-Núñez, Mario J. Pérez-Jiménez

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
Universidad de Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: {lvalencia, dorellana, ariscosn, marper}@us.es

Summary. P systems with active membranes is a well developed framework in the field of Membrane Computing. Using evolution, communication, dissolution and division rules, we know that some kinds of problems can be solved by those systems, but taking into account which ingredients are used. All these rules are inspired by the behavior of living cells, who "compute" with their proteins in order to obtain energy, create components, send information to other cells, kill themselves (in a process called *apoptosis*), and so on.

But there are other behaviors not captured in this framework. As *mitosis* is simulated by *division* rules (for elementary and non-elementary membranes), *meiosis*, that is, membrane fission inspiration is captured in *separation* rules. It differs from the first in the sense of duplication of the objects (that is, in *division* rules, we duplicate the objects not involved in the rule, meanwhile in *separation* rules we divide the content of the original membrane into the new membranes created).

Evolution rules simulate the transformation of components in membranes, but it is well known that elements interact with another ones in order to obtain new components. Cooperation in evolution rules is considered. More specifically, minimal cooperation (in the sense that only two objects can interact in order to create one or two objects).

Key words: Membrane Computing, Active membranes, Minimal cooperation, Mitosis, Computational Complexity, The **P** versus **NP** problem.

1 Introduction

Membrane Computing is a distributed parallel computing paradigm inspired by the way the living cells process chemical substances, energy and information. The processor units in the basic model are abstractions of biological membranes, selectively permeable barriers which give cells their outer boundaries (plasma membranes) and their inner compartments (organelles). They control the flow of information

between cells and the movement of substances into and out of cells, and they are also involved in the capture and release of energy. Biological membranes play an active part in the life of the cell. In fact, the passing of a chemical substance through a biological membrane is often implemented by an interaction between the membrane itself and the protein channels present in it. During this interaction, both the chemical substance and the membrane can be modified, at least locally.

Mitosis is a process by which two or more cells are produced/generated from one cell that could be considered as the "mother". Several cell division inspired mechanisms were introduced in Membrane Computing. Specifically, *P systems with active membranes* [9] incorporates the mitosis based mechanisms by means of *membrane division* rules. By applying this kind of rules, under the influence of the object triggering it, the membrane is divided into two membranes and that object is replaced in the two new ones by possibly new objects, while the remaining objects are *duplicated* in both newly created membranes. These models are universal (they are equivalent in power to deterministic Turing machines) and they have the ability to provide efficient solutions to computationally hard problems, by making use of an exponential workspace created in a polynomial time (often, in linear time). Moreover, **PSPACE**-complete problems can be efficiently solved by families of P systems with active membranes which use division for elementary and non-elementary membranes. This paper deals with P systems with active membranes where electrical charges are removed.

The paper is organized as follows. Next section briefly describes some preliminaries in order to make the work self-contained. In Section 3, syntax and semantics of polarizationless P systems with active membranes by using membrane division rules or membrane separation rules are introduced, and minimal cooperation in object evolution rules is considered. Definition of *Recognizer membrane systems* is recalled in Section 4, as a framework to provide efficient solutions to decision problems. The computational efficiency of polarizationless P systems with active membranes, division rules, minimal cooperation and without dissolution rules is established in Section 5 by providing a uniform polynomial-time solution to `SAT` problem. A formal verification of this result is presented in Section 6. Next section is dedicated to show the limits of the computational efficiency of the polarizationless P systems with active membranes, separation rules and minimal cooperation in object evolution rules. The paper ends with some open problems and concluding remarks.

2 Preliminaries

An *alphabet* Γ is a non-empty set and their elements are called *symbols*. A *string* u over Γ is an ordered finite sequence of symbols, that is, a mapping from a natural number $n \in \mathbb{N}$ onto Γ. The number n is called the *length* of the string u and it is denoted by $|u|$, that is, the length of a string is the number of occurrences of symbols that it contains. The empty string (with length 0) is denoted by λ. The

set of all strings over an alphabet Γ is denoted by Γ^*. A *language* over Γ is a subset of Γ^*.

A *multiset* over an alphabet Γ is an ordered pair (Γ, f) where f is a mapping from Γ onto the set of natural numbers $\mathbb{N}$. The *support* of a multiset $m = (\Gamma, f)$ is defined as $supp(m) = \{x \in \Gamma \mid f(x) > 0\}$. A multiset is finite (respectively, empty) if its support is a finite (respectively, empty) set. We denote by $\emptyset$ the empty multiset and we denote by $M_f(\Gamma)$ the set of all finite multisets over Γ.

Let $m_1 = (\Gamma, f_1)$, $m_2 = (\Gamma, f_2)$ be multisets over Γ, then the union of m_1 and m_2, denoted by $m_1 + m_2$, is the multiset (Γ, g), where $g(x) = f_1(x) + f_2(x)$ for each $x \in \Gamma$. We say that m_1 is contained in m_2 and we denote it by $m_1 \subseteq m_2$, if $f_1(x) \leq f_2(x)$ for each $x \in \Gamma$. The relative complement of m_2 in m_1, denoted by $m_1 \setminus m_2$, is the multiset (Γ, g), where $g(x) = f_1(x) - f_2(x)$ if $f_1(x) \geq f_2(x)$, and $g(x) = 0$ otherwise.

Let us recall that a *free tree* (*tree*, for short) is a connected, acyclic, undirected graph. A *rooted tree* is a tree in which one of the vertices (called *the root of the tree*) is distinguished from the others. In a rooted tree the concepts of ascendants and descendants are defined in a usual way. Given a node x (different from the root), if the last edge on the (unique) path from the root of the tree to the node x is $\{x, y\}$ (in this case, $x \neq y$), then y is **the** *parent* of node x and x is **a** *child* of node y. The root is the only node in the tree with no parent. A node with no children is called a *leaf* (see [3] for details).

3 Polarizationless P Systems with Active Membranes

Let us briefly recall some definitions of P systems models that will be used in the paper (see [12] for details).

A *basic transition* P system is a membrane system whose rules are of the following forms: evolution, communication, and dissolution. In these systems the size of the membrane structure does not increase, but an exponential workspace (in terms of number of objects) can be constructed in linear time, e.g. via evolution rules of the type $[\, a \to a^2 \,]_h$. Nevertheless, such capability is not enough to efficiently solve **NP**–complete problems, unless **P** = **NP** (see [6] for details).

Replication is one of the most important functions of a cell and, in ideal circumstances, a cell produces two identical copies by division. Bearing in mind that the reactions which take place in a cell are related to membranes, division rules for elementary and non-elementary membranes are considered in the so-called *P systems with active membranes*. Such variant was first introduced by Gh. Păun [10] and it has associated electrical charges with membranes but the rules are non-cooperative and there are not priorities. Nevertheless, the class of all problems solvable in polynomial time and in a uniform way by means of families of P systems with active membranes which use division for elementary and non-elementary membranes contains class **PSPACE** and it is contained in class

EXP [16]. Thus, in order to provide efficient solutions to computationally hard problems, this framework seems to be too powerful from the computational complexity point of view.

In this paper, electrical charges are removed from P systems with active membranes. Two different ways of producing an exponential number of membranes in linear time will be considered: division and separation rules (abstractions of mitosis and membrane fission processes, respectively).

3.1 Polarizationless P system with active membranes: Syntax

Definition 1. *A polarizationless P system with active membranes and membrane division of degree $q \geq 2$ is a tuple $\Pi = (\Gamma, H, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$, where*

- *Γ is a finite alphabet whose elements are called objects;*
- *H is a finite alphabet such that $H \cap \Gamma = \emptyset$ whose elements are called labels;*
- *μ is a labelled rooted tree (called membrane structure) consisting of q nodes injectively labeled by elements of H;*
- *$\mathcal{M}_1, \ldots, \mathcal{M}_q$ are finite multisets over Γ;*
- *$\mathcal{R}$ is a finite set of rules, of the following forms:*

  (a_0) $[\, a \to u \,]_h$ *for $h \in H$, $a \in \Gamma$, $u \in M_f(\Gamma)$* (object evolution rules).

  (b_0) $a\,[\ \ ]_h \to [\, b \,]_h$ *for $h \in H$, $a, b \in \Gamma$ and h is not the label of the root of μ* (send–in communication rules).

  (c_0) $[\, a \,]_h \to b\,[\ \ ]_h$ *for $h \in H$, $a, b \in \Gamma$* (send–out communication rules).

  (d_0) $[\, a \,]_h \to b$ *for $h \in H \setminus \{i_{out}\}$, $a, b \in \Gamma$ and h is not the label of the root of μ* (dissolution rules).

  (e_0) $[\, a \,]_h \to [\, b \,]_h\, [\, c \,]_h$ *for $h \in H \setminus \{i_{out}\}$, $a, b, c \in \Gamma$ and h is not the label of the root of μ* (division rules for elementary membranes).

  (f_0) $[\,[\ ]_{h_0}[\ ]_{h_1}\,]_h \to [\,[\ ]_{h_0}\,]_h\, [\,[\ ]_{h_1}\,]_h$, *where $h \in H \setminus \{i_{out}\}$ is not the label of the root of μ and $h_0, h_1 \in H$* (division rules for non–elementary membranes).
- *$i_{out} \in H \cup \{env\}$, where $env \notin H$ and in the case $i_{out} \in H$, i_{out} is the label of a leaf of μ.*

Definition 2. *A polarizationless P system with active membranes and membrane separation of degree $q \geq 2$ is a tuple $\Pi = (\Gamma, \Gamma_0, \Gamma_1, H, H_0, H_1, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{out})$, where*

- *Γ is a finite alphabet whose elements are called objects;*
- *H is a finite alphabet such that $H \cap \Gamma = \emptyset$ whose elements are called labels;*
- *$\{\Gamma_0, \Gamma_1\}$ is a partition of Γ and $\{H_0, H_1\}$ is a partition of H;*
- *μ is a labelled rooted tree (called membrane structure) consisting of q nodes injectively labeled by elements of H;*
- *$\mathcal{M}_1, \ldots, \mathcal{M}_q$ are finite multisets over Γ;*
- *$\mathcal{R}$ is a finite set of rules, of the following forms:*

  (a_0) $[\, a \to u \,]_h$ *for $h \in H$, $a \in \Gamma$, $u \in M_f(\Gamma)$* (object evolution rules).

(b_0) $a[\ ]_h \to [b]_h$ *for* $h \in H$, $a, b \in \Gamma$ *and* h *is not the label of the root of* μ (send–in communication rules).

(c_0) $[a]_h \to b[\ ]_h$ *for* $h \in H$, $a, b \in \Gamma$ (send–out communication rules).

(d_0) $[a]_h \to b$ *for* $h \in H \setminus \{i_{out}\}$, $a, b \in \Gamma$ *and* h *is not the label of the root of* μ (dissolution rules).

(e_0) $[a]_h \to [\Gamma_0]_h [\Gamma_1]_h$ *for* $h \in H \setminus \{i_{out}\}$, $a \in \Gamma$ *and* h *is not the label of the root of* μ (separation rules for elementary membranes).

(f_0) $[[\ ]_{h_0}[\ ]_{h_1}]_h \to [\Gamma_0[\ ]_{h_0}]_h\ [\Gamma_1[\ ]_{h_1}]_h$, *where* $h \in H \setminus \{i_{out}\}$ *is not the label of the root of* μ, $h_0 \in H_0$ *and* $h_1 \in H_1$ (separation rules for non–elementary membranes).

- $i_{out} \in H \cup \{env\}$, *where* $env \notin H$ *and in the case* $i_{out} \in H$, i_{out} *is the label of a leaf of* μ.

A polarizationless P system with active membranes of degree $q \geq 2$, can be viewed as a set of q membranes, labelled by elements of H, arranged in a hierarchical structure μ given by a rooted tree whose root is called the *skin membrane*, such that: (a) $\mathcal{M}_1, \ldots, \mathcal{M}_q$ represent the finite multisets of *objects* initially placed in the q membranes of the system; (b) $\mathcal{R}$ is a finite set of rules over Γ associated with the labels; and (c) $i_{out} \in H \cup \{env\}$ indicates the output region. We use the term *region* i to refer to membrane i in the case $i \in H$ and to refer to the "environment" of the system in the case $i = env$. The leaves of μ are called elementary membranes, otherwise, the membrane is said to be non-elementary.

3.2 Polarizationless P system with active membranes: Semantics

An *instantaneous description* or a *configuration* $\mathcal{C}_t$ at an instant t of a polarizationless P system with active membranes is described by the following elements: (a) the membrane structure at instant t, and (b) all multisets of objects over Γ associated with all the membranes present in the system at that moment.

An object evolution rule $[a \to u]_h$ for $h \in H$, $a \in \Gamma$, $u \in M_f(\Gamma)$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t, if there exists a membrane labelled by h in $\mathcal{C}_t$ which contains object a. When applying such a rule, object a is consumed and objects from multiset u are produced in that membrane.

A send-in communication rule $a[\ ]_h \to [b]_h$ for $h \in H$, $a, b \in \Gamma$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t, if there exists a membrane labelled by h in $\mathcal{C}_t$ such that h is not the label of the root of μ and its parent membrane contains object a. When applying such a rule, object a is consumed from the parent membrane and object b is produced in the corresponding membrane h.

A send-out communication rule $[a]_h \to b[\ ]_h$ for $h \in H$, $a, b \in \Gamma$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t, if there exists a membrane labelled by h in $\mathcal{C}_t$ such that it contains object a. When applying such a rule, object a is consumed from such membrane h and object b is produced in the parent of such membrane.

A dissolution rule $[a]_h \to b$ for $h \in H \setminus \{i_{out}\}$, $a, b \in \Gamma$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t, if there exists a membrane labelled by h in

$\mathcal{C}_t$, different from the skin membrane and the output region, such that it contains object a. When applying such a rule, object a is consumed, membrane h is dissolved and its objects are sent to the parent (or the first ancestor that has not been dissolved).

A division rule $[\,a\,]_h \rightarrow [\,b\,]_h[\,c\,]_h$ for $h \in H \setminus \{i_{out}\}, a, b, c \in \Gamma$, is *applicable* to a configuration $\mathcal{C}_t$ at an instant t, if there exists an elementary membrane labelled by h in $\mathcal{C}_t$, different from the skin membrane and the output region, such that it contains object a. When applying a division rule $[a]_h \rightarrow [\,b\,]_h\,[\,c\,]_h$ to a membrane labelled by h in a configuration $\mathcal{C}_t$, under the influence of object a, the membrane with label h is divided into two membranes with the same label; in the first copy, object a is replaced by object b, in the second one, object a is replaced by object c; all the other objects are replicated and copies of them are placed in the two new membranes.

A division rule $[\,[\,]_{h_0}[\,]_{h_1}\,]_h \rightarrow [\,[\,]_{h_0}\,]_h\ [\,[\,]_{h_1}\,]_h$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t, if there exists a membrane labelled by h in $\mathcal{C}_t$, different from the skin membrane and the output region, which contains a membrane labelled by h_0 and another membrane labelled by h_1. When applying such a division rule to a membrane labelled by h in a configuration $\mathcal{C}_t$, the membrane with label h is divided into two membranes with the same label; the first copy inherits membrane h_0 with its contents, and the second copy inherits membrane h_1 with its contents. Besides, if the membrane labelled by h contains more membranes other than those with the labels h_0, h_1, then such membranes are duplicated so that they become part of the contents of both new copies of the membrane h.

A separation rule $[\,a\,]_h \rightarrow [\ \Gamma_0\ ]_h\ [\ \Gamma_1\ ]_h$ for $h \in H$, $a \in \Gamma$, is applicable to a configuration $\mathcal{C}_t$ at an instant t, if there exists a membrane labelled by h in $\mathcal{C}_t$, different from the skin membrane and the output region, such that it contains object a. When applying such a rule, the membrane is separated into two membranes with the same label; at the same time, object a is consumed and the multiset of objects contained in membrane h gets distributed: the objects from Γ_0 are placed in the first membrane, those from Γ_1 are placed in the second membrane.

A separation rule $[\,[\,]_{h_0}[\,]_{h_1}\,]_h \rightarrow [\ \Gamma_0\ [\,]_{h_0}\,]_h\ [\ \Gamma_1\ [\,]_{h_1}\,]_h$, where h, h_0, h_1 are labels such that $h_0 \in H_0$ and $h_1 \in H_1$, is applicable to a configuration $\mathcal{C}_t$ at an instant t, if there exists a membrane labelled by h in $\mathcal{C}_t$, different from the skin membrane and the output region, such that it contains a membrane labelled by h_0 and another membrane labelled by h_1. When applying such a separation rule to a membrane labelled by h in a configuration $\mathcal{C}_t$, that membrane is separated into two membranes with the same label, in such a way that the contents (multiset of objects and inner membranes) are distributed as follows: The first membrane receives the multiset of objects from Γ_0, and all inner membranes whose label belongs to H_0; and the second membrane receives the multiset of objects from Γ_1, and all inner membranes whose label belongs to H_1.

In polarizationless P systems with active membranes, the rules are applied according to the following principles:

- The rules associated with membranes labelled with h are used for all copies of this membrane.
- At one transition step, one object can be used by only one rule (chosen in a non–deterministic way).
- At one transition step, *a membrane* can be the subject of *only one* rule of types (b_0)–(f_0), and then it is applied at most once.
- Object evolution rules can be simultaneously applied to a membrane with one rule of types (b_0)–(f_0). Object evolution rules are applied in a maximally parallel manner.
- 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 evolve by means of rules of type (a_0), then we suppose that first the evolution rules of type (a_0) are used, changing the objects, and then the division (or the separation) is produced. Of course, this process takes only one transition step.
- The skin membrane and the output membrane can never get divided, separated, nor dissolved.

3.3 Polarizationless P systems with active membranes and minimal cooperation in object evolution rules

Next, we incorporate cooperation in object evolution rules of polarizationless P systems with active membranes. In this paper, we use minimal cooperation in the following sense: the left-hand side of each object evolution rules has at most two objects, and the length of the right-hand side cannot be greater than the length of the left-hand side. Consequently, in contrast with the usual object evolution rules in P systems with active membranes, by applying these rules with minimal cooperation the number of objects of the system does not increase.

Definition 3. *A polarizationless P system with active membranes, division or separation rules and minimal cooperation in object evolution rules is a polarizationless P system with active membranes and division or separation rules such that the object evolution rules are of the following form:*

$$[\, a \to c \,]_h,\ [\, a\, b \to c \,]_h,\ [\, a\, b \to c\, d \,]_h$$

for $h \in H$ and $a, b, c, d \in \Gamma$.

The semantics of these variants are analogous to the semantics of polarizationless P systems with active membranes.

4 Recognizer membrane systems

In what follows, a *membrane system* denotes a P system of any of the different variants considered in the previous section.

Definition 4. *We say that a membrane system Π is a recognizer membrane system if the following holds:*

1. *The working alphabet Γ of Π has two distinguished objects* `yes` *and* `no`.
2. *Σ is an (input) alphabet strictly contained in Γ.*
3. *The initial multisets $\mathcal{M}_1, \ldots, \mathcal{M}_q$ of Π are finite multisets over $\Gamma \setminus \Sigma$.*
4. *There exists a distinguished membrane labelled by i_{in} called the input membrane.*
5. *The output region i_{out} is the environment.*
6. *All computations halt.*
7. *If $\mathcal{C}$ is a computation of Π, then either object* `yes` *or object* `no` *(but not both) must have been released into the environment, and only at the last step of the computation.*

For each finite multiset m over the input alphabet Σ, the *computation of the system Π with input m* starts from the configuration obtained by adding the input multiset m to the contents of the input membrane, in the initial configuration of Π. We denote it by $\Pi + m$. Therefore, we have an initial configuration associated with each input multiset m (over the input alphabet Σ) in this kind of systems.

We use the following notations:

- $\mathcal{DAM}^0(\gamma, \delta)$ where $\gamma \in \{-d, +d\}$ and $\delta \in \{-n, +n\}$, is the class of all recognizer polarizationless P systems with active membranes and division rules.
- $\mathcal{DAM}^0_{mc}(\gamma, \delta)$ where $\gamma \in \{-d, +d\}$ and $\delta \in \{-n, +n\}$, is the class of all recognizer polarizationless P systems with active membranes, minimal cooperation in object evolution rules and division rules.
- $\mathcal{SAM}^0(\gamma, \delta)$ where $\gamma \in \{-d, +d\}$ and $\delta \in \{-n, +n\}$, is the class of all recognizer polarizationless P systems with active membranes and separation rules.
- $\mathcal{SAM}^0_{mc}(\gamma, \delta)$ where $\gamma \in \{-d, +d\}$ and $\delta \in \{-n, +n\}$, is the class of all recognizer polarizationless P systems with active membranes, minimal cooperation in object evolution rules and separation rules.

The meaning of parameters γ and δ is the following:

- if $\gamma = +d$ then dissolution rules are permitted.
- if $\gamma = -d$ then dissolution rules are forbidden.
- if $\delta = +n$ then division rules for elementary and non–elementary membranes are permitted.
- if $\delta = -n$ then division rules only for elementary membranes are permitted.

Let us notice that standard notation in the literature referring to polarizationless P systems with active membranes ($\mathcal{AM}^0(\gamma, \delta)$) corresponds, within this new notation, to the class $\mathcal{DAM}^0(\gamma, \delta)$.

4.1 Polynomial complexity classes of recognizer membrane systems

Next, let us recall the concept of efficient solvability by means of a family of recognizer membrane systems (see [13] for more details).

Definition 5. *Let $\mathcal{R}$ be a class of recognizer membrane systems. We say that a decision problem X is solvable in polynomial time by a family $\mathbf{\Pi} = \{\Pi(n) \mid n \in \mathbb{N}\}$ of systems from $\mathcal{R}$, in a uniform way, denoted by $X \in \mathbf{PMC}_{\mathcal{R}}$, if the following hold:*

- *the family $\mathbf{\Pi}$ is polynomially uniform by Turing machines, that is, there exists a deterministic Turing machine working in polynomial time which constructs the system $\Pi(n)$ from $n \in \mathbb{N}$;*
- *there exists a pair (cod, s) of polynomial-time computable functions over I_X such that:*
  - *for each instance $u \in I_X$, $s(u)$ is a natural number and $cod(u)$ is an input multiset of the system $\Pi(s(u))$;*
  - *for each $n \in \mathbb{N}$, $s^{-1}(n)$ is a finite set;*
  - *the family $\mathbf{\Pi}$ is polynomially bounded with regard to (X, cod, s), that is, there exists a polynomial function p, such that for each $u \in I_X$ every computation of $\Pi(s(u)) + cod(u)$ is halting and it performs at most $p(|u|)$ steps;*
  - *the family $\mathbf{\Pi}$ is sound with regard to (X, cod, s), that is, for each $u \in I_X$, if there exists an accepting computation of $\Pi(s(u)) + cod(u)$, then $\theta_X(u) = 1$;*
  - *the family $\mathbf{\Pi}$ is complete with regard to (X, cod, s), that is, for each $u \in I_X$, if $\theta_X(u) = 1$, then every computation of $\Pi(s(u)) + cod(u)$ is an accepting one.*

The polynomial complexity class $\mathbf{PMC}_{\mathcal{R}}$ is closed under polynomial-time reduction and under complement [14].

4.2 Known results on polarizationless P systems with active membranes

In previous works, membrane systems have been studied in terms of their computational efficiency and different borderlines between efficiency and non-efficiency have been obtained. Each of them provides attractive characterizations of the $\mathbf{P} \neq \mathbf{NP}$ conjecture.

In [5], by using the dependency graph technique and the tractability of the reachability problem, the following result has been proved.

Theorem 1. $\mathbf{P} = \mathbf{PMC}_{\mathcal{DAM}^0(-d,+n)}$

Thus, only problems in class $\mathbf{P}$ can be solved in polynomial time and in a uniform way by means of families of polarizationless P systems with active membranes making use of division rules for elementary and non-elementary membranes and not using dissolution rules.

In [2], a family of polarizationless P systems that make use of dissolution and division rules for elementary and non-elementary membranes solving the QSAT (*quantified satisfiability*) problem in polynomial time and in a uniform way was proposed.

Theorem 2. $\texttt{QSAT} \in \mathbf{PMC}_{\mathcal{DAM}^0(+d,+n)}$

Therefore, the following holds.

Corollary 1. PSPACE $\subseteq \mathbf{PMC}_{\mathcal{DAM}^0(+d,+n)}$

In [1], a family $\mathbf{\Pi}$ of P systems from $\mathcal{DAM}^0(+d,+n)$ solving SAT problem in polynomial time and in a *semi-uniform* way (each P system of the family is associated with only one instance of the problem) was proposed. Recall that SAT is one of the most well known **NP**-complete problems [4]. Next, based on the solution of QSAT problem provided in [2], a family of P systems from $\mathcal{DAM}^0(+d,+n)$ solving SAT problem in polynomial time and in a *uniform* way is presented.

Theorem 3. $\texttt{SAT} \in \mathbf{PMC}_{\mathcal{DAM}^0(+d,+n)}$

Proof. Let φ be a propositional formula in conjunctive normal form such that:

- $\varphi = C_1 \wedge \ldots \wedge C_p$
- $C_i = y_1 \vee \ldots \vee y_{l_i}$, for $1 \leq i \leq p, y_j \in \{x_k, \overline{x}_k \,|\, 1 \leq k \leq n\}$ being n the number of variables occurring in the formula.

We construct $\Pi = (\Gamma, \Sigma, H, \mu, \mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_{2n+p+3}, \mathcal{R}, i_{in}, i_{out})$ that will solve all instances of formulas with n variables and p clauses, provided that the appropriate input multiset $cod(\varphi) = \{v_{i,j} | x_i \in C_j\} \cup \{v'_{i,j} | \neg x_i \in C_j\}$ is supplied to the system (through the corresponding input membrane):

- $\Gamma = \Sigma \cup \{d_i | 1 \leq i \leq 7n+2p+2\} \cup \{f_i, t_i, a_i | 1 \leq i \leq n\} \cup \{c_i | 1 \leq i \leq p\} \cup \{u_{i,j}, u'_{i,j} | 1 \leq i \leq n, 1 \leq j \leq p\} \cup \{t', f', z, z', T, F, \texttt{yes}, \texttt{no}\}$
- $\Sigma = \{v_{i,j}, v'_{i,j} | 1 \leq i \leq n, 1 \leq j \leq p\}$
- $H = \{0, 1, \ldots, 2n+p+3\}$
- $[\,[\,[\,\ldots[\,[\,\,]_0\,]_1\,\,\ldots\,]_{2n+p+1}]_{2n+p+2}]_{2n+p+3}$
- $\mathcal{M}_0 = \mathcal{M}_{2n+p+2} = d_0, \mathcal{M}_i = \emptyset, i \notin \{0, 2n+p+2\}$
- $i_{in} = 0, i_{out} = env$

Rules are distributed as follow:

- **Generation Stage**

$$\left.\begin{array}{l}[d_{2i} \to a_{i+1}\ d_{2i+1}]_0 \\ [d_{2i+1} \to d_{2i+2}]_0\end{array}\right\} 1 \leq i < n$$
$$[a_i]_0 \to [t_i]_0 [f_i]_0, 1 \leq i \leq n$$
$$[[\,\,]_i[\,\,]_i]_{i+1} \to [[\,\,]_i]_{i+1} [[\,\,]_i]_{i+1}, 0 \leq i < 2n+p$$
$$[d_{2n+i} \to d_{2n+i+1}]_0, 0 \leq i \leq 2n+p$$
$$[d_{4n+p+1}]_0 \to T$$

$[d_i \to d_{i+1}]_{2n+p+2}, 0 \leq i \leq 7n + 2p + 1$

In $4n + p + 1$ steps, we expand the membrane structure in a tree-like fashion, preparing for the checking stage. First, we use $2n$ steps to generate 2^n copies of membrane 0, each one of them encoding a different truth assignment. Then, some more non-elementary divisions take place in the following $2n+p+1$ steps, in such a way that we get 2^n copies of a linear nested structure composed by membranes j, for $0 \leq j \leq 2n + p + 1$.
After $4n+p+1$ steps, all the contents of membranes labelled by 0 are released into their corresponding parent membranes (labelled by 1).

- **Assignments Stage**

$$\left.\begin{array}{l} [t_i \to t']_{2i-1} \\ [t']_{2i-1} \to z \\ [f_i]_{2i-1} \to f' \\ [f' \to z]_{2i} \\ [z]_{2i} \to z' \end{array}\right\} 1 \leq i \leq n$$

$$\left.\begin{array}{l} [v_{i,j} \to u_{i,j}]_{2i-1} \\ [v'_{i,j} \to u'_{i,j}]_{2i-1} \end{array}\right\} 1 \leq i \leq n, 1 \leq j \leq p$$

$$\left.\begin{array}{l} [u'_{i,j} \to \lambda]_{2i-1} \\ [u_{i,j} \to c_j]_{2i-1} \\ [u_{i,j} \to \lambda]_{2i} \\ [u'_{i,j} \to c_j]_{2i} \end{array}\right\} 1 \leq i \leq n, 1 \leq j \leq p$$

We have to see whether each truth assignment makes true φ or not. The formula φ has been satisfied if and only if objects c_i, with all $i \in \{1, \ldots, n\}$ have been created. After $3n$ steps, all membranes labelled by j with $0 \leq j \leq 2n$ have been dissolved, and their contents are gathered into membranes labelled by $2n + 1$.

- **Checking Stage**

$[c_i]_{2n+i} \to z', 1 \leq i \leq p$
$[T]_{2n+p+1} \to T$

That means, if the truth assignment satisfies all clauses of the formula φ, then we have that φ is satisfied, so we can proceed to the output stage.

- **Output Stage**

$[d_{7n+2p+2}]_{2n+p+2} \to F$
$[T]_{2n+p+2} \to T$
$[T \to T']_{2n+p+3}$
$[T']_{2n+p+3} \to \texttt{yes}[\ \ ]_{2n+p+3}$
$[F]_{2n+p+3} \to \texttt{no}[\ \ ]_{2n+p+3}$

After $7n+2p+4$ steps, we obtain an object `yes` or an object `no`, but not both, in the environment, and that is the solution for the `SAT` instance that is being analyzed.

□

Let us notice that from Theorem 1 and Corollary 1 we have:

- $\mathbf{P} = \mathbf{PMC}_{\mathcal{DAM}^0(-d,+n)}$.
- $\mathbf{PSPACE} \subseteq \mathbf{PMC}_{\mathcal{DAM}^0(+d,+n)}$.

Therefore, in the framework of polarizationless P systems with active membranes making use of division rules for elementary and non-elementary membranes, dissolution rules provide a frontier of the efficiency, that is, in that framework passing from forbidden to allowed dissolution rules amounts to passing from non–efficiency to efficiency, assuming that $\mathbf{P} \neq \mathbf{PSPACE}$.

At the beginning of 2005, Gh. Păun proposed a problem (problem **F** from [11]) which can be formally formulated as follows:

"Is the complexity class $\mathbf{PMC}_{\mathcal{DAM}^0(+d,-n)}$ *equal to* $\mathbf{P}$*?"*

The so-called *Păun conjecture* is $\mathbf{PMC}_{\mathcal{DAM}^0(+d,-n)} = \mathbf{P}$, and until now it has not been proved. Nevertheless, in [5] a partial affirmative answer was given when such membrane systems make no use of dissolution rules ($\mathbf{PMC}_{\mathcal{DAM}^0(-d,+n)} = \mathbf{P}$), and assuming that $\mathbf{P} \neq \mathbf{NP}$, a partial negative answer was given when division rules both for elementary and non-elementary membranes are permitted in such membrane systems ($\mathbf{NP} \cup \mathbf{co}-\mathbf{NP} \subseteq \mathbf{PMC}_{\mathcal{DAM}^0(+d,+n)}$).

5 On efficiency of membrane systems from $\mathcal{DAM}^0_{mc}(-d,-n)$

Dissolution rules play a relevant role in the efficiency of polarizationless P systems which make use of division rules both for elementary and non-elementary membranes. In this section, we show that the syntactical ingredient of minimal cooperation in polarizationless P systems with active membranes (without dissolution and allowing only division for elementary membranes) is enough to solve computationally hard problems in an efficient way. That is, in the previous framework efficiency is reached by trading minimal cooperation for dissolution.

Next, a polynomial time solution to `SAT` problem, by a family $\mathbf{\Pi} = \{\Pi(t) \mid t \in \mathbb{N}\}$ of recognizer P systems from $\mathcal{DAM}^0_{mc}(-d,-n)$ is provided. Each system $\Pi(t)$ will process all Boolean formulas φ in conjunctive normal form with n variables and p clauses, where $t = \langle n,p\rangle$, provided that the appropriate input multiset $cod(\varphi)$ is supplied to the system (through the corresponding input membrane).

Let us recall that the polynomial–time computable function (the *pair function*) $\langle n,p\rangle = ((n+p)(n+p+1)/2)+n$ is a primitive recursive and bijective function

from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$. Then, for each $n, p \in \mathbb{N}$, we consider the recognizer P system of degree 2 from $\mathcal{DAM}^0_{mc}(-d, -n)$

$$\Pi(\langle n, p\rangle) = (\Gamma, \Sigma, H, \mu, \mathcal{M}_1, \mathcal{M}_2, \mathcal{R}, i_{in}, i_{out})$$

defined as follows:

(1) Working alphabet:

$\Gamma = \Sigma \cup \{\texttt{yes}, \texttt{no}, \alpha, \beta', \beta'', \gamma, \gamma', \gamma'', \#\} \cup \{a_{i,k} \mid 1 \leq i \leq n, 1 \leq k \leq i\} \cup$
$\{\beta_k \mid 0 \leq k \leq n + 2p\} \cup \{t_{i,k}, f_{i,k} \mid 1 \leq i \leq n-1, i \leq k \leq n-1\} \cup$
$\{T_{i,j}, F_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq p\} \cup \{c_{j,k} \mid 1 \leq j \leq p-1, j \leq k \leq p-1\} \cup$
$\{c_j \mid 1 \leq j \leq p\} \cup \{d_j \mid 2 \leq j \leq p\}$

where the input alphabet is $\Sigma = \{x_{i,j}, \overline{x}_{i,j}, x^*_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq p\}$.

(2) $H = \{1, 2\}$.

(3) Membrane structure: $\mu = [\ [\ \ ]_2]_1$, that is, $\mu = (V, E)$ where $V = \{1, 2\}$ and $E = \{\{1, 2\}\}$.

(4) Initial multisets: $\mathcal{M}_1 = \{\alpha, \beta_0\}$ and $\mathcal{M}_2 = \{a_{1,1}, \cdots, a_{n,1}\}$.

(5) The set $\mathcal{R}$ of rules consists of the following rules:

1.1 Rules to produce an affirmative answer.

$[\ \alpha\ \gamma\ \longrightarrow\ \gamma'\ ]_1$

$[\ \gamma'\ \longrightarrow\ \gamma''\ ]_1$

$[\ \gamma''\ ]_1\ \longrightarrow\ \texttt{yes}\ [\ \ ]_1$

1.2 Rules to produce a negative answer.

$[\ \beta_k\ \longrightarrow\ \beta_{k+1}\ ]_1$, for $0 \leq k \leq n + 2p - 1$

$[\ \beta_{n+2p}\ \longrightarrow\ \beta'\ ]_1$

$[\ \alpha\ \beta'\ \longrightarrow\ \beta''\ ]_1$

$[\ \beta''\ ]_1\ \longrightarrow\ \texttt{no}\ [\ \ ]_1$

2.1 Rules to generate truth assignments.

$[\ a_{i,i}\ ]_2 \longrightarrow [\ t_{i,i}\ ]_2\ [\ f_{i,i}\ ]_2$, for $1 \leq i \leq n-1$

$[a_{k,i} \longrightarrow\ a_{k,i+1}\ ]_2$, for $2 \leq k \leq n, 1 \leq i \leq k-1$

$[\ a_{n,n}\ ]_2 \longrightarrow [\ T_{n,1}\ ]_2\ [\ F_{n,1}\ ]_2$

2.2 Rules of synchronization.

$$\left.\begin{array}{l}[t_{i,k} \longrightarrow\ t_{i,k+1}\ ]_2 \\ [f_{i,k} \longrightarrow\ f_{i,k+1}\ ]_2\end{array}\right\} 1 \leq i \leq n-2, i \leq k \leq n-2$$

$$\left.\begin{array}{l}[t_{i,n-1} \longrightarrow\ T_{i,1}\ ]_2 \\ [f_{i,n-1} \longrightarrow\ F_{i,1}\ ]_2\end{array}\right\} 1 \leq i \leq n-1$$

2.3 Rules to check clauses.

$$\left.\begin{array}{l}
[T_{i,j}\ x_{i,j} \longrightarrow T_{i,j+1}\ c_{j,j}\ ]_2 \\
[T_{i,j}\ \overline{x}_{i,j} \longrightarrow T_{i,j+1}\ ]_2 \\
[T_{i,j}\ x^*_{i,j} \longrightarrow T_{i,j+1}\ ]_2 \\
[F_{i,j}\ x_{i,j} \longrightarrow F_{i,j+1}\ ]_2 \\
[F_{i,j}\ \overline{x}_{i,j} \longrightarrow F_{i,j+1}\ c_{j,j}\ ]_2 \\
[F_{i,j}\ x^*_{i,j} \longrightarrow F_{i,j+1}\ ]_2
\end{array}\right\} 1 \leq i \leq n, 1 \leq j \leq p-1$$

$$\left.\begin{array}{l}
[T_{i,p}\ x_{i,p} \longrightarrow c_p\ ]_2 \\
[T_{i,p}\ \overline{x}_{i,p} \longrightarrow \#\ ]_2 \\
[T_{i,p}\ x^*_{i,p} \longrightarrow \#\ ]_2 \\
[F_{i,p}\ x_{i,p} \longrightarrow \#\ ]_2 \\
[F_{i,p}\ \overline{x}_{i,p} \longrightarrow c_p\ ]_2 \\
[F_{i,p}\ x^*_{i,p} \longrightarrow \#\ ]_2
\end{array}\right\} 1 \leq i \leq n$$

$$[c_{i,j} \longrightarrow c_{i,j+1}\ ]_2\ 1 \leq i \leq j \leq p-2$$

$$[c_{i,p-1} \longrightarrow c_i\ ]_2\ 1 \leq i \leq p-1$$

2.4 Rules to detect if a truth assignment makes true the input formula.

$$[c_1\ c_2 \longrightarrow d_2\ ]_2$$

$$[d_j\ c_{j+1} \longrightarrow d_{j+1}\ ]_2\,, \text{ for } 1 \leq j \leq p-1$$

$$[\ d_p\ ]_2 \longrightarrow \gamma\ [\ \ ]_2$$

(6) The input membrane is membrane labelled by 2 ($i_{in} = 2$) and the output region is the environment ($i_{out} = env$).

Let us notice that for each $t \in \mathbb{N}$, the system $\Pi(t)$ is deterministic.

6 A formal verification

Let $\varphi = C_1 \wedge \cdots \wedge C_p$ an instance of the SAT problem consisting of p clauses $C_j = l_{j,1} \vee \cdots \vee l_{j,r_j}$, $1 \leq j \leq p$, where $Var(\varphi) = \{x_1, \cdots, x_n\}$, and $l_{j,k} \in \{x_i, \neg x_i \mid 1 \leq i \leq n\}$, $1 \leq j \leq p, 1 \leq k \leq r_j$. Let us assume that the number of variables, n, and the number of clauses, p, of φ, are greater or equal to 2.

We consider the polynomial encoding (cod, s) from SAT in $\mathbf{\Pi}$ defined as follows: for each $\varphi \in I_{\mathtt{SAT}}$ with n variables and p clauses, $s(\varphi) = \langle n, p \rangle$ and

$$cod(\varphi) = \{x_{i,j} \mid\ x_i \in C_j\} \ \cup\ \{\overline{x}_{i,j} \mid \neg x_i \in C_j\} \ \cup\ \{x^*_{i,j} \mid x_i \notin C_j, \neg x_i \notin C_j\}$$

For instance, the formula $\varphi = (x_1 + x_2 + \overline{x}_3)(\overline{x}_2 + x_4)(\overline{x}_2 + x_3 + \overline{x}_4)$ is encoded as follows:

$$cod(\varphi) = \begin{pmatrix} x_{1,1} & x_{2,1} & \overline{x}_{3,1} & x^*_{4,1} \\ x^*_{1,2} & \overline{x}_{2,2} & x^*_{3,2} & x_{4,2} \\ x^*_{1,3} & \overline{x}_{2,3} & x_{3,3} & \overline{x}_{4,3} \end{pmatrix}$$

That is, j-th row ($1 \leq j \leq p$) represents the j-th clause C_j of φ. We denote $(cod(\varphi))_j^p$ the code of the clauses $C_j, \ldots, C_p$, that is, the expression containing from j-th row to p-th row. For instance,

$$cod(\varphi)_2^p = \begin{pmatrix} x_{1,2}^* \ \overline{x}_{2,2} \ x_{3,2}^* \ x_{4,2} \\ x_{1,3}^* \ \overline{x}_{2,3} \ x_{3,3} \ \overline{x}_{4,3} \end{pmatrix}$$

The Boolean formula φ will be processed by the system $\Pi(s(\varphi)) + cod(\varphi)$. Next, we informally describe how that system works.

The solution proposed follows a brute force algorithm in the framework of recognizer P systems with active membranes, minimal cooperation in object evolution rules and division rules only for elementary membranes, and it consists of the following stages:

- *Generation stage*: using division rules, all truth assignments for the variables $\{x_1, \ldots, x_n\}$ associated with φ are produced. Specifically, 2^n membranes labelled by 2 are generated, each of them encoding a truth assignment. This stage spends n computation steps exactly, being n the number of variables of φ.
- *First Checking stage*: checking whether or not each clause of the input formula φ is satisfied by the truth assignments generated in the previous stage, encoded by each membrane labelled by 2. This stage takes exactly p steps, being p the number of clauses of φ.
- *Second Checking stage*: checking whether or not all clauses of the input formula φ are satisfied by some truth assignment encoded by a membrane labelled by 2. This stage takes exactly $p - 1$ steps, being p the number of clauses of φ.
- *Output stage*: the system sends to the environment the right answer according to the results of the previous stage. This stage takes exactly 4 steps.

6.1 Generation stage

At this stage, all truth assignments for the variables associated with the Boolean formula $\varphi(x_1, \ldots, x_n)$ are going to be generated, by applying division rules from **2.1** in membranes labelled by 2. In such manner that in the i–th step ($1 \leq i \leq n-1$) of this stage, division rule associated with object $a_{i,i}$ is triggered, producing objects $t_{i,1}, f_{i,1}$ in the new created membranes labelled by 2. In the last step of this stage the objects produced are $T_{n,1}$ and $F_{n,1}$, respectively.

Proposition 1. *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_q)$ be a computation of the system $\Pi(s(\varphi))$ with input multiset $cod(\varphi)$.*

(a) For each i $(1 \leq i \leq n-1)$ at configuration $\mathcal{C}_i$ we have the following:

- $\mathcal{C}_i(1) = \{\alpha\,,\, \beta_i\}$.
- *There are 2^i membranes labelled by 2 such that each of them contains*
  - ⋆ *the input multiset $cod(\varphi)$;*
  - ⋆ *objects $a_{i+1,i+1}, \ldots, a_{n,i+1}$; and*

⋆ *a different subset* $\{r_{1,i}, \ldots, r_{i,i}\}$, *being* $r \in \{t\,,\, f\}$.

(b) $\mathcal{C}_n(1) = \{\alpha\,,\, \beta_n\}$, *and in* $\mathcal{C}_n(2)$ *there are* 2^n *membranes labelled by* 2 *such that each of them contains the input multiset* $cod(\varphi)$, *as well as a different subset* $\{R_{1,1}, \ldots, R_{n,1}\}$, *being* $R \in \{T, F\}$.

Proof. (a) is going to be demonstrated by induction on i.

- The base case $i = 1$ is trivial because at the initial configuration $\mathcal{C}_0$ we have: $\mathcal{C}_0(1) = \{\alpha\,,\, \beta_0\}$ and there exists a single membrane labelled by 2 containing $cod(\varphi)$ and the set $\{a_{1,1}, \ldots, a_{n,1}\}$. Then, configuration $\mathcal{C}_0$ yields configuration $\mathcal{C}_1$ by applying the rules:
  $[\ a_{1,1}\ ]_2 \ \rightarrow\ [\ t_{1,1}\ ]_2\ [\ f_{1,1}\ ]_2$
  $[\ a_{i,1}\ \rightarrow\ a_{i,2}\ ]_2$, for $2 \leq i \leq n$
  $[\ \beta_0\ \rightarrow\ \beta_1\ ]_1$
  Thus, $\mathcal{C}_1(1) = \{\alpha\,,\, \beta_1\}$ and in $\mathcal{C}_1$ there exist two membranes labelled by 2 such that their contents is $cod(\varphi)$ and the set $\{a_{2,2}, \ldots, a_{n,2}\}$. Also, one of those membranes contains object $t_{1,1}$ and the other one object $f_{1,1}$. Hence, the result holds for $i = 1$.
- Supposing that, by induction, result is true for i $(1 \leq i < n-1)$; that is,
  - $\mathcal{C}_i(1) = \{\alpha\,,\, \beta_i\}$.
  - There are 2^i membranes labelled by 2 such that each of them contains
    ⋆ the input multiset $cod(\varphi)$;
    ⋆ objects $a_{i+1,i+1}, \ldots, a_{n,i+1}$; and
    ⋆ a different subset $\{r_{1,i}, \ldots, r_{i,i}\}$, being $r \in \{t\,,\, f\}$.

  Then, configuration $\mathcal{C}_i$ yields configuration $\mathcal{C}_{i+1}$ by applying the rules:
  $[\ t_{k,i}\ \rightarrow\ t_{k,i+1}\ ]_2$, for $1 \leq k \leq i$
  $[\ a_{i+1,i+1}\ ]_2\ \rightarrow\ [\ t_{i+1,i+1}\ ]_2\ [\ f_{i+1,i+1}\ ]_2$
  $[\ a_{k,i+1}\ \rightarrow\ a_{k,i+2}\ ]_2$, for $i+2 \leq k \leq n$
  $[\ \beta_i\ \rightarrow\ \beta_{i+1}\ ]_1$
  Therefore, the following holds:
  - $\mathcal{C}_{i+1}(1) = \{\alpha\,,\, \beta_{i+1}\}$.
  - There are 2^{i+1} membranes labelled by 2 such that each of them contains
    ⋆ the input multiset $cod(\varphi)$;
    ⋆ objects $a_{i+2,i+2}, \ldots, a_{n,i+2}$; and
    ⋆ a different subset $\{r_{1,i+1}, \ldots, r_{i+1,i+1}\}$, being $r \in \{t\,,\, f\}$.

  Hence, the result holds for $i + 1$.

In order to prove (b) it is enough to notice that, on the one hand, from (a) configuration $\mathcal{C}_{n-1}$ holds:

- $\mathcal{C}_{n-1}(1) = \{\alpha\,,\, \beta_{n-1}\}$.
- There are 2^{n-1} membranes labelled by 2 such that each of them contains
  ⋆ the input multiset $cod(\varphi)$;
  ⋆ object $a_{n,n}$; and
  ⋆ a different subset $\{r_{1,n-1}, \ldots, r_{n-1,n-1}\}$, being $r \in \{t\,,\, f\}$.

On the other hand, configuration $\mathcal{C}_{n-1}$ yields configuration $\mathcal{C}_n$ by applying the rules:

$[\ t_{k,n-1}\ \rightarrow\ T_{k,1}\ ]_2$, for $1 \leq k \leq n-1$
$[\ a_{n,n}\ ]_2\ \rightarrow\ [\ T_{n,1}\ ]_2\ [\ F_{n,1}\ ]_2$
$[\ \beta_{n-1}\ \rightarrow\ \beta_n\ ]_1$

Then, we have $\mathcal{C}_n(1) = \{\alpha\,,\,\beta_n\}$, and in $\mathcal{C}_n(2)$ there are 2^n membranes labelled by 2 such that each of them contains the input multiset $cod(\varphi)$, as well as a different subset $\{R_{1,1},\ldots,R_{n,1}\}$, being $R \in \{T,F\}$.

□

6.2 First Checking stage

At this stage, we try to determine the clauses satisfied for the truth assignment encoded by each membrane labelled by 2. For that, rules from **2.3** will be applied in such manner that in the j-th step ($1 \leq j \leq p$) of this stage, clause j is checked and an object c_j is produced in the case that clause is satisfied.

Proposition 2. *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_q)$ be a computation of the system $\Pi(s(\varphi))$ with input multiset $cod(\varphi)$.*

(a) For each i $(1 \leq i \leq p-1)$ at configuration $\mathcal{C}_{n+i}$ we have the following:
- *$\mathcal{C}_{n+i}(1) = \{\alpha\,,\,\beta_{n+i}\}$.*
- *There are 2^n membranes labelled by 2 such that each of them contains*
  - ⋆ *the input multiset $cod(\varphi)^p_{i+1}$ corresponding to the clauses $c_{i+1},\ldots,c_p$;*
  - ⋆ *a different subset $\{R_{1,i+1},\ldots,R_{n,i+1}\}$, being $R \in \{T\,,\,F\}$ encoding a truth assignment for the variables $\{x_1,\ldots,x_n\}$; and*
  - ⋆ *objects $c_{j,i}$ $(1 \leq j \leq i)$ such that clause C_j is satisfied by the truth assignment encoded by such a membrane.*

(b) $\mathcal{C}_{n+p}(1) = \{\alpha\,,\,\beta_{n+p}\}$, and in $\mathcal{C}_{n+p}(2)$ there are 2^n membranes labelled by 2 such that each of them contains objects c_j such that clause C_j is satisfied by the truth assignment encoded by such a membrane. Besides, the multiplicity of object c_j represents the number of values of the truth assignment making true C_j.

Proof. (a) is going to be demonstrated by induction on i.

- In order to prove the base case $i = 1$ let us notice that from the previous proposition we deduce that configuration $\mathcal{C}_n$ verifies: $\mathcal{C}_n(1) = \{\alpha\,,\,\beta_n\}$ and in $\mathcal{C}_n(2)$ there are 2^n membranes labelled by 2 such that each of them contains the input multiset $cod(\varphi)$, as well as a different subset $\{R_{1,1},\ldots,R_{n,1}\}$, being $R \in \{T,F\}$. Besides, configuration $\mathcal{C}_n$ yields configuration $\mathcal{C}_{n+1}$ by applying rule $[\ \beta_n\ \rightarrow\ \beta_{n+1}\ ]_1$ and rules:

$$\left.\begin{array}{l}
[T_{i,1}\ x_{i,1} \rightarrow\ T_{i,2}\ c_{1,1}\ ]_2 \\
[T_{i,1}\ \overline{x}_{i,1} \rightarrow\ T_{i,2}\ ]_2 \\
[T_{i,1}\ x^*_{i,1} \rightarrow\ T_{i,2}\ ]_2 \\
[F_{i,1}\ x_{i,1} \rightarrow\ F_{i,2}\ ]_2 \\
[F_{i,1}\ \overline{x}_{i,1} \rightarrow\ F_{i,2}\ c_{1,1}\ ]_2 \\
[F_{i,1}\ x^*_{i,1} \rightarrow\ F_{i,2}\ ]_2
\end{array}\right\} 1 \leq i \leq n, 1 \leq j \leq p-1$$

Thus, the following holds for configuration $\mathcal{C}_{n+1}$:

- $\mathcal{C}_{n+1}(1) = \{\alpha\,,\, \beta_{n+1}\}$.
- There are 2^n membranes labelled by 2 such that each of them contains
  - ⋆ the input multiset $cod(\varphi)^p_2$ corresponding to the clauses $c_2, \ldots, c_p$;
  - ⋆ a different subset $\{R_{1,2}, \ldots, R_{n,2}\}$, being $R \in \{T\,,\, F\}$ encoding a truth assignment for the variables $\{x_1, \ldots, x_n\}$; and
  - ⋆ objects $c_{1,1}$ such that clause C_1 is satisfied by the truth assignment encoded by such a membrane.

Hence, the result holds for $i = 1$.

- Let us assume that by induction hypothesis, the result holds for i $(1 \leq i < p-1)$; that is,
  - $\mathcal{C}_{n+i}(1) = \{\alpha\,,\, \beta_{n+i}\}$.
  - There are 2^n membranes labelled by 2 such that each of them contains
    - ⋆ the input multiset $cod(\varphi)^p_{i+1}$ corresponding to the clauses $c_{i+1}, \ldots, c_p$;
    - ⋆ a different subset $\{R_{1,i+1}, \ldots, R_{n,i+1}\}$, being $R \in \{T\,,\, F\}$ encoding a truth assignment for the variables $\{x_1, \ldots, x_n\}$; and
    - ⋆ objects $c_{j,i}$ $(1 \leq j \leq i)$ such that clause C_j is satisfied by the truth assignment encoded by such a membrane.

  Besides, configuration $\mathcal{C}_{n+i}$ yields configuration $\mathcal{C}_{n+(i+1)}$ by applying rule $[\ \beta_{n+i}\ \rightarrow\ \beta_{n+(i+1)}\ ]_1$ and rules:

$$\begin{array}{l}
[T_{i,i+1}\ x_{i,i+1} \rightarrow\ T_{i,i+2}\ c_{i+1,i+1}\ ]_2 \\
[T_{i,i+1}\ \overline{x}_{i,i+1} \rightarrow\ T_{i,i+2}\ ]_2 \\
[T_{i,i+1}\ x^*_{i,i+1} \rightarrow\ T_{i,i+2}\ ]_2 \\
[F_{i,i+1}\ x_{i,i+1} \rightarrow\ F_{i,2}\ ]_2 \\
[F_{i,i+1}\ \overline{x}_{i,i+1} \rightarrow\ F_{i,i+2}\ c_{i+1,i+1}\ ]_2 \\
[F_{i,i+1}\ x^*_{i,i+1} \rightarrow\ F_{i,i+2}\ ]_2 \\
[c_{j,i}\ \longrightarrow\ c_{j,i+1}\ ]_2\ :\ 1 \leq j \leq i
\end{array}$$

Thus, the following holds for configuration $\mathcal{C}_{n+(i+1)}$:

- $\mathcal{C}_{n+(i+1)}(1) = \{\alpha\,,\, \beta_{n+(i+1)}\}$.
- There are 2^n membranes labelled by 2 such that each of them contains
  - ⋆ the input multiset $cod(\varphi)^p_{i+2}$ corresponding to the clauses $c_{i+2}, \ldots, c_p$;
  - ⋆ a different subset $\{R_{1,i+2}, \ldots, R_{n,i+2}\}$, being $R \in \{T\,,\, F\}$ encoding a truth assignment for the variables $\{x_1, \ldots, x_n\}$; and
  - ⋆ objects $c_{j,i+1}$ $(1 \leq j \leq i+1)$ such that clause C_j is satisfied by the truth assignment encoded by such a membrane.

Hence, the result holds for $i + 1$.

In order to prove (b) it is enough to notice that, on the one hand, from (a) configuration $\mathcal{C}_{n+p-1}$ verifies the following:

- $\mathcal{C}_{n+p-1}(1) = \{\alpha\,,\,\beta_{n+p-1}\}$.
- There are 2^n membranes labelled by 2 such that each of them contains
  - ⋆ the input multiset $cod(\varphi)_p^p$ corresponding to the clause c_p;
  - ⋆ a different subset $\{R_{1,p},\ldots,R_{n,p}\}$, being $R \in \{T\,,\,F\}$ encoding a truth assignment for the variables $\{x_1,\ldots,x_n\}$; and
  - ⋆ objects $c_{j,p-1}$ $(1 \leq j \leq p-1)$ such that clause C_j is satisfied by the truth assignment encoded by such a membrane.

On the other hand, configuration $\mathcal{C}_{n+p-1}$ yields configuration $\mathcal{C}_{n+p}$ by applying rule $[\ \beta_n\ \rightarrow\ \beta_{n+1}\ ]_1$ and rules:

$$\left.\begin{array}{l}[T_{i,p}\ x_{i,p} \rightarrow\ c_p\ ]_2\\ [T_{i,p}\ \overline{x}_{i,p} \rightarrow\ \#\ ]_2\\ [T_{i,p}\ x^*_{i,p} \rightarrow\ \#\ ]_2\\ [F_{i,p}\ x_{i,p} \rightarrow\ \#\ ]_2\\ [F_{i,p}\ \overline{x}_{i,p} \rightarrow\ c_p\ ]_2\\ [F_{i,p}\ x^*_{i,p} \rightarrow\ \#\ ]_2\end{array}\right\} 1 \leq i \leq n$$

$$[c_{j,p-1}\ \longrightarrow\ c_j\ ]_2\ :\ 1 \leq j \leq p-1$$

Thus, configuration $\mathcal{C}_{n+p}$ holds: $\mathcal{C}_{n+p}(1) = \{\alpha\,,\,\beta_{n+p}\}$ and in $\mathcal{C}_{n+p}(2)$ there are 2^n membranes labelled by 2 such that each of them contains objects c_j such that clause C_j is satisfied by the truth assignment encoded by such a membrane. Besides, the multiplicity of object c_j represents the number of values of the truth assignment making true C_j.

□

6.3 Second Checking stage

At this stage, we try to determine if some truth assignment encoded by a membrane labelled by 2 satisfied all clauses of the input formula. For that, rules from **2.4** will be applied in such manner that object d_j $(2 \leq j \leq p)$ is produced in the case clauses $c_1,\ldots,c_j$ all satisfied. Then, the input formula is satisfied by the truth assignment encoded by a membrane labelled by 2 if and only if object d_p appears in that membrane. This stage spends $p-1$ computation steps.

Proposition 3. *Let $\mathcal{C} = (\mathcal{C}_0,\mathcal{C}_1,\ldots,\mathcal{C}_q)$ be a computation of the system $\Pi(s(\varphi))$ with input multiset $cod(\varphi)$.*

(a) For each i $(1 \leq i \leq p-1)$ at configuration $\mathcal{C}_{n+p+i}$ we have the following:

- $\mathcal{C}_{n+p+i}(1) = \{\alpha\,,\,\beta_{n+p+i}\}$.
- *There are 2^n membranes labelled by 2 such that each of them contains objects d_{i+1} if and only if the truth assignment encoded in that membrane, makes true clauses $C_1,\ldots,C_{i+1}$.*

(b) φ is satisfiable if and only if at configuration $\mathcal{C}_{n+2p-1}$ there exists some membrane labelled by 2 which contains some object d_p.

Proof. (a) is going to be demonstrated by induction on i.

- In order to prove the base case $i = 1$, let us notice that from the previous proposition we deduce that configuration $\mathcal{C}_{n+p}$ verifies: $\mathcal{C}_{n+p}(1) = \{\alpha\,,\, \beta_{n+p}\}$ and in $\mathcal{C}_{n+p}(2)$ there are 2^n membranes labelled by 2 such that each of them contains objects c_j such that clause C_j is satisfied by the truth assignment encoded by such a membrane. Besides, the multiplicity of object c_j represents the number of values of the truth assignment making true C_j.
  Configuration $\mathcal{C}_{n+p}$ yields configuration $\mathcal{C}_{n+p+1}$ by applying the rules:
  $[\ c_1\ c_2\ \rightarrow\ d_2\ ]_2$
  $[\ \beta_{n+p}\ \rightarrow\ \beta_{n+p+1}\ ]_1$
  Thus, in configuration $\mathcal{C}_{n+p+1}$ the following holds:
  - $\mathcal{C}_{n+p+1}(1) = \{\alpha\,,\, \beta_{n+p+1}\}$.
  - There are 2^n membranes labelled by 2 such that each of them contains objects d_2 if and only if the truth assignment encoded in that membrane, makes true clauses C_1 and C_2.

  Hence, the result holds for $i = 1$.
- Supposing that, by induction, result is true for i $(1 \leq i < n-1)$; that is,
  - $\mathcal{C}_{n+p+i}(1) = \{\alpha\,,\, \beta_{n+p+i}\}$.
  - There are 2^n membranes labelled by 2 such that each of them contains objects d_{i+1} if and only if the truth assignment encoded in that membrane, makes true clauses $C_1, \ldots, C_{i+1}$.

  Then, configuration $\mathcal{C}_{n+p+i}$ yields configuration $\mathcal{C}_{n+p+(i+1)}$ by applying the rules:
  $[\ c_{i+1}\ c_{i+2}\ \rightarrow\ d_2\ ]_2$
  $[\ \beta_{n+p+i}\ \rightarrow\ \beta_{n+p+(i+1)}\ ]_1$
  Thus, in configuration $\mathcal{C}_{n+p+(i+1)}$ the following holds:
  - $\mathcal{C}_{n+p+(i+1)}(1) = \{\alpha\,,\, \beta_{n+p+(i+1)}\}$.
  - There are 2^n membranes labelled by 2 such that each of them contains objects d_{i+2} if and only if the truth assignment encoded in that membrane, makes true clauses $C_1, \ldots, C_{i+2}$.

  Hence, the result holds for $i + 1$.

In order to proof (b), let us note that formula φ is satisfiable if and only if there exists a truth assignment σ making true φ, that is, making true clauses $C_1, \ldots, C_p$. From (a) we deduce that φ is satisfiable if and only at configuration $\mathcal{C}_{n+2p-1}$ there exists some membrane labelled by 2 which contains some object d_p.

□

6.4 Output stage

The output phase starts at the $(n + 2p)$-th step, and takes exactly four steps.

- *Affirmative answer*: if the input formula φ of SAT problem is satisfiable then at least one of the truth assignments from a membrane with label 2 has satisfied all clauses. Thus, a copy of object d_p will appear in that membrane at configuration $\mathcal{C}_{n+2p-1}$. Then, by applying the last rule from **2.4** and rule $[\,\beta_{n+2p-1} \longrightarrow \beta_{n+2p}\,]_1$, objects γ and β_{n+2p} are produced in the skin membrane. At the next step, by applying rules $[\,\alpha\,\gamma \longrightarrow \gamma'\,]_1$ and $[\,\beta_{n+2p} \longrightarrow \beta'\,]_1$, objects γ' and β' are produced in the skin membrane. At the next step, by applying rule $[\,\gamma' \longrightarrow \gamma''\,]_1$, object γ'' is produced in the skin membrane (let us notice that object β' cannot interact with α). Finally, at the step $n + 2p + 3$ by applying rule $[\,\gamma''\,]_1 \longrightarrow \texttt{yes}[\ \ ]_1$, object yes is sent out to the environment and the computation halts.
- *Negative answer*: if the input formula φ of SAT problem is not satisfiable then none of the truth assignments encoded by a membrane with label 2 makes the formula φ true. Thus, object d_p does not appear in any membrane with label 2. Thus, at step $n + 2p$, only rule $[\,\beta_{n+2p-1} \longrightarrow \beta_{n+2p}\,]_1$ is applicable to $\mathcal{C}_{n+2p-1}$. Then, $\mathcal{C}_{n+2p}(1) = \{\alpha\,,\,\beta_{n+2p}\}$. At the next step, by applying rule $[\,\beta_{n+2p} \longrightarrow \beta'\,]_1$ we have $\mathcal{C}_{n+2p+1}(1) = \{\alpha\,,\,\beta'\}$. Then rule $[\,\alpha\,\beta' \longrightarrow \beta''\,]_1$ produces an object β'' in the skin membrane. Finally, at step $n + 2p + 3$ by applying rule $[\,\beta''\,]_1 \longrightarrow \texttt{no}[\ \ ]_1$ releases an object no at the environment. Then, the computation halts and the answer of the computation is no.

6.5 Result

Theorem 4. $\texttt{SAT} \in \mathbf{PMC}_{\mathcal{DAM}^0_{mc}(-d,-n)}$.

Proof: The family of P systems previously constructed verifies the following:

(a) Every system of the family $\mathbf{\Pi}$ is a recognizer P system from $\mathcal{DAM}^0_{mc}(-d,-n)$.
(b) The family $\mathbf{\Pi}$ is polynomially uniform by Turing machines because for each $n, p \in \mathbb{N}$, the rules of $\Pi(\langle n,p\rangle)$ of the family are recursively defined from $n, p \in \mathbb{N}$, and the amount of resources needed to build an element of the family is of a polynomial order in n and p, as shown below:
   - Size of the alphabet: $5np + \frac{3n^2-5n+p^2-3p+6}{2} + n + 4p + 9 \in \Theta((max\{n,p\})^2)$.
   - Initial number of cells: $2 \in \Theta(1)$.
   - Initial number of objects in cells: $n + 2 \in \Theta(n)$.
   - Number of rules: $6np + \frac{3n^2+p^2+3n+5p}{2} + 6 \in \Theta((max\{n,p\})^2)$.
   - Maximal number of objects involved in any rule: $4 \in \Theta(1)$.
(c) The pair (cod, s) of polynomial–time computable functions defined fulfill the following: for each input formula φ of SAT problem, $s(\varphi)$ is a natural number, $cod(\varphi)$ is an input multiset of the system $\Pi(s(\varphi))$, and for each $n \in \mathbb{N}$, $s^{-1}(n)$ is a finite set.
(d) The family $\mathbf{\Pi}$ is polynomially bounded: indeed for each input formula φ of SAT problem, the deterministic P system $\Pi(s(\varphi)) + cod(\varphi)$ takes exactly, in $n + 2p + 3$ steps, being n the number of variables of φ and p the number of clauses.

(e) The family $\mathbf{\Pi}$ is sound with regard to (X, cod, s): indeed for each input formula φ, if the computation of $\Pi(s(\varphi)) + cod(\varphi)$ is an accepting computation, then φ is satisfiable (see Section 6).
(f) The family $\mathbf{\Pi}$ is complete with regard to (X, cod, s): indeed, for each input formula φ such that it is satisfiable, the accepting computation of $\Pi(s(\varphi)) + cod(\varphi)$ is an accepting computation (see Section 6).

Therefore, the family $\mathbf{\Pi}$ of P systems previously constructed solves SAT problem in polynomial time and in a uniform way, according to Definition 5.

□

Corollary 2. $\mathbf{NP} \cup \mathbf{co} - \mathbf{NP} \subseteq \mathbf{PMC}_{\mathcal{DAM}^0_{mc}(-d,-n)}$

Proof: It suffices to notice that SAT problem is a **NP**-complete problem, SAT$\in \mathbf{PMC}_{\mathcal{DAM}^0_{mc}(-d,-n)}$, and the complexity class $\mathbf{PMC}_{\mathcal{DAM}^0_{mc}(-d,-n)}$ is closed under polynomial-time reduction and under complement.

□

7 Limits on efficient computations in $\mathcal{SAM}^0_{mc}(+d, +n)$

In this section we study the computational efficiency of polarizationless P systems with active membranes, dissolution rules and minimal cooperation when separation rules (for elementary and non-elementary membranes) are considered as a mechanism to generate an exponential workspace in linear time. Specifically, we will show that these kind of P systems can only solve problems in class **P** in an efficient way. The proof is inspired on a similar result, obtained in the framework of cell-like P systems with symport/antiport rules and cell separation [7].

Let $\Pi = (\Gamma, \Gamma_0, \Gamma_1, \Sigma, H, H_0, H_1, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{in}, i_{out})$ be a recognizer P system from $\mathcal{SAM}^0_{mc}(+d, +n)$. In what follows we use the concepts of notation from [15].

- We denote by $p(i)$ (resp., $ch(i)$) the label of the parent (resp., a child) of the membrane labelled by i, the parent of the skin membrane is the environment (we write $p(1) = 0$). We denote by $\mathcal{R}_E$ (resp., $\mathcal{R}_C$, $\mathcal{R}_D$ and $\mathcal{R}_S$) the set of evolution rules (resp., communication, dissolution and separation rules) of Π. We will fix total orders in $\mathcal{R}_E$, $\mathcal{R}_C$, $\mathcal{R}_D$ and $\mathcal{R}_S$.
- Let $\mathcal{C}$ be a computation of Π, and $\mathcal{C}_t$ an arbitrary configuration of $\mathcal{C}$. With respect to the number of objects of the system, let us notice that by applying a single rule, this number remains unchanged or decreases by one. Thus, the total number of objects in $\mathcal{C}_t$ is, at most, M, being $M = |\mathcal{M}_0 + ... + \mathcal{M}_q|$.
  With respect to the number of membranes of the system, by applying a separation rule for elementary membranes, an object is removed from the system, no new objects are produced and a new membrane is created. Thus, at most M membranes can be produced by means of this process. Also, by applying a separation rule for non-elementary membranes, the number of objects remains

unchanged but a new membrane is created (when such a rule is applied to a non-elementary membrane, it cannot be applied to that membrane anymore). In this way, no more than $q-2$ new membranes can be generated. Consequently, $q+M+(q-2)=M+2q-2$ is an upper bound of the total number of membranes at $\mathcal{C}_t$.

- In order to identify the membranes created by the application of a separation rule, we modify the labels of the new membranes in the following recursive manner:
  - The label of a membrane will be a pair (i,σ) where $0 \leq i \leq q$ and $\sigma \in \{0,1\}^*$. At the initial configuration, the labels of the membranes are $(1,\lambda),\ldots,(q,\lambda)$. The label of the environment is denoted by $(0,\lambda)$.
  - If a separation rule is applied to a membrane labelled by (i,σ), then the new created membranes will be labelled by $(i,\sigma 0)$ and $(i,\sigma 1)$, respectively. Membrane $(i,\sigma 0)$ will only contain the objects of membrane (i,σ) which belong to Γ_0, and membrane $(i,\sigma 1)$ will only contain the objects of membrane (i,σ) which belong to Γ_1. Only elementary membranes can be separated, so if a membrane i is non-elementary then we denote it by the label (i,λ).
  - If an object evolution rule or a communication rule is applied to a membrane labelled by (i,σ), then after the application of the rule, the membrane keeps its label.
- Let us notice that the number of labels we need to identify all membranes appearing along any computation of a P system from $\mathcal{SAM}^0_{mc}(+d,+n)$ is of the order $O(M+q)$.
- A configuration $\mathcal{C}_t$ of a P system from $\mathcal{SAM}^0_{mc}(+d,+n)$ is described by the current membrane structure and the multisets of labelled objects of the type

$$\{(a,i,\sigma) \,:\, a \in \Gamma, 0 \leq i \leq q, \sigma \in \{0,1\}^*\}$$

  The expression $(a,i,\sigma) \in \mathcal{C}_t$ means that object a belongs to membrane labelled by (i,σ).
- Let $r=[ab \to c]_h \in \mathcal{R}$ be an object evolution rule of Π. We denote by $n \cdot LHS(r,(i,\sigma))$, $n \in \mathbf{N}$, the multiset of labelled objects $(a,i,\sigma)^n(b,i,\sigma)^n$ We denote by $n \cdot RHS(r,(i,\sigma))$ the multiset of labelled objects $(c,i,\sigma)^n$ produced by applying n times rule r over membrane (i,σ). Similarly these concepts are defined for object evolution rules of the forms $[\,ab \to cd\,]_h$ and $[\,a \to c\,]_h$.
- Let $r=[a]_h \to b[\ \ ]_h \in \mathcal{R}$ be a send-out communication rule of Π. We denote by $LHS(r,(i,\sigma))$ the labelled object (a,i,σ). We denote by $RHS(r,(i,\sigma))$ the labelled object $(b,p(i),\tau)$ produced by applying rule r over membrane (i,σ), where $(p(i),\tau)$ is the parent of membrane (i,σ).
- Let $r=a[\ \ ]_h \to [b]_h \in \mathcal{R}$ be a send-in communication rule of Π. We denote by $LHS(r,(i,\sigma))$ the labelled object $(a,p(i),\tau)$, where $(p(i),\tau)$ is the parent of membrane (i,σ). We denote by $RHS(r,(i,\sigma))$ the labelled object (b,i,σ) produced by applying rule r over membrane (i,σ).
- Let $\mathcal{C}_t$ is a configuration of Π, we denote by $\mathcal{C}_t + \{(x,i,\sigma)/\sigma'\}$ the multiset obtained by replacing in $\mathcal{C}_t$ every occurrence of (x,i,σ) by (x,i,σ'). Besides,

$\mathcal{C}_t + m$ (resp., $\mathcal{C}_t \setminus m$) is used to denote that a multiset m of labelled objects is added (resp., removed) to the configuration.

Next, we provide a deterministic algorithm $\mathcal{A}$ working in polynomial time that receives as input a recognizer P system Π from $\mathcal{SAM}^0_{mc}(+d,+n)$ together with an input multiset m of Π. Then algorithm $\mathcal{A}$ reproduces the behaviour of a single computation of such system.

The pseudocode of the algorithm $\mathcal{A}$ is described as follows:

Input: A P system Π from $\mathcal{SAM}^0_{mc}(+d,+n)$ and an input multiset m of Π
  Initialization stage: the initial configuration $\mathcal{C}_0$ of $\Pi + m$
  $t \leftarrow 0$
  while $\mathcal{C}_t$ is a non-halting configuration **do**
    Selection stage: Input $\mathcal{C}_t$, Output $(\mathcal{C}'_t, A)$
    Execution stage: Input $(\mathcal{C}'_t, A)$, Output $\mathcal{C}_{t+1}$
    $t \leftarrow t + 1$
  end while
Output: *Yes* if $\mathcal{C}_t$ is an accepting configuration, *No* otherwise

The selection stage and the execution stage implement a transition step of a recognizer P system Π. Specifically, the selection stage receives as input a configuration $\mathcal{C}_t$ of Π at an instant t. The output of this stage is a pair $(\mathcal{C}'_t, A)$, where A encodes a multiset of rules selected to be applied to $\mathcal{C}_t$, and $\mathcal{C}'_t$ is the configuration obtained from $\mathcal{C}_t$ once the labelled objects corresponding to the application of rules from A have been consumed. The execution stage receives as input the output $(\mathcal{C}'_t, A)$ of the selection stage, and the output is the next configuration $\mathcal{C}_{t+1}$ of $\mathcal{C}_t$. Specifically, at this stage, configuration $\mathcal{C}'_t$ yields configuration $\mathcal{C}_{t+1}$ by adding the labelled objects produced by the application of rules from A.

Next, selection stage and execution stage are described in detail.

Selection stage.

Input: A configuration $\mathcal{C}_t$ of Π at instant t
  $\mathcal{C}'_t \leftarrow \mathcal{C}_t$; $A \leftarrow \emptyset$; $B \leftarrow \emptyset$
  for each membrane (i,σ) of $\mathcal{C}'_t$ according to the lexicographical order **do**
    for each $r \in \mathcal{R}_E$ according to the order chosen **do**
      $n_r \leftarrow$ maximum number of times that r is applicable to (i,σ)
      if $n_r > 0$ **then**
        $\mathcal{C}'_t \leftarrow \mathcal{C}'_t \setminus n_r \cdot LHS(r,(i,\sigma))$
        $A \leftarrow A \cup \{(r, n_r, (i,\sigma))\}$
      end if
    end for
    for each $r \in \mathcal{R}_C$ according to the order chosen **do**
      if $(i,\sigma) \notin B$ and r is applicable to (i,σ) in $\mathcal{C}'_t$ **then**
        $\mathcal{C}'_t \leftarrow \mathcal{C}'_t \setminus LHS(r,(i,\sigma))$
        $A \leftarrow A \cup \{(r, 1, (i,\sigma))\}$
        $B \leftarrow B \cup \{(i,\sigma)\}$
      end if
    end for

```
    for each r ≡ [ a ]_i → b ∈ R_D according to the order chosen do
      if (i,σ) ∉ B and r is applicable to (i,σ) in C'_t then
        C'_t ← C'_t \ {(a,(i,σ))}
        A ← A ∪ {(r,1,(i,σ))}
        B ← B ∪ {(i,σ)}
      end if
    end for
    for r ∈ R_S according to the order chosen do
      if (i,σ) ∉ B and r is applicable to (i,σ) in C'_t then
        C'_t ← C'_t \ LHS(r,(i,σ))
        A ← A ∪ {(r,1,(i,σ))}
        B ← B ∪ {(i,σ)}
      end if
    end for
  end for
```

This algorithm is deterministic and works in polynomial time. Indeed, the cost in time is polynomial in the size of Π because the number of cycles of the external main **for** loop is of order $O(M+q)$, and the number of cycles of the three internal main **for** loops are of order $O(|R|)$. Besides, the cost of each internal loops is of the order $O(M+q)$.

Let us notice that the number of tuples in set A is of the order $O(M)$ because each object in the system can be involved in, at most, one rule and at any configuration $\mathcal{C}_t$ the total number of objects is upper bounded by M. In set A an order is considered in a natural way (a product order concerning the rules, natural numbers and labels).

In order to complete the simulation of a computation step of the system Π, the execution stage takes care of the effects of applying the rules selected in the previous stage: updating the objects according to the RHS of the rules.

Execution stage.

```
Input: The output C'_t and A of the selection stage
    for each (r,n_r,(i,σ)) ∈ A according to the order chosen do
      if r ∈ R_E then
        C'_t ← C'_t + n_r · RHS(r,(i,σ))
      if r ∈ R_C then
        C'_t ← C'_t + RHS(r,(i,σ))
      if r ∈ R_D then
        C'_t ← C'_t + RHS(r,(p(i),σ))
        C'_t ← C'_t + {(x,(p(i),σ)) | x is in membrane (i,σ) in C'_t}
        Update the parent function by removing the membrane (i,σ)
      else if r ∈ R_S then
        C'_t ← C'_t + {(λ,i,σ)/σ0}
        C'_t ← C'_t + {(λ,i,σ1)}
        for each (x,i,σ) ∈ C'_t according to the lexicographical order do
          if x ∈ Γ_0 then
            C'_t ← C'_t + {(x,i,σ)/σ0}
```

```
            else
                C'_t ← C'_t + {(x, i, σ)/σ1}
            end if
        end for
        for each (j, τ) ∈ C'_t do
            if p(j, τ) = (i, σ) and j ∈ H_0 then p(j, τ) = p(i, σ0)
            else if p(j, τ) = (i, σ) and j ∈ H_1 then p(j, τ) = p(i, σ1)
            end if
        end for
    end if
end for
C_{t+1} ← C'_t
```

This algorithm is deterministic and works in polynomial time. Indeed, on the one hand, the number of cycles of the main **for** loop is of order $O(M)$. On the other hand, each cycle of the main **for** loop takes $O(|R|)$ steps plus the number of steps spend by the two secondary **for** loops: the first takes $O(M(M+q))$ steps and the second takes $O(M+q)$ steps.

Theorem 5. $\mathbf{P} = \mathbf{PMC}_{\mathcal{SAM}^0_{mc}(+d,+n)}$.

Proof: It suffices to prove that $\mathbf{PMC}_{\mathcal{SAM}^0_{mc}(+d,+n)} \subseteq \mathbf{P}$. For that, let $X = (I_X, \theta_X)$ be a decision problem in $\mathbf{PMC}_{\mathcal{SAM}^0_{mc}(+d,+n)}$. Let $\{\Pi(n) \mid n \in \mathbb{N}\}$ be a family of P systems from $\mathcal{SAM}^0_{mc}(+d,+n)$ solving X, according to Definition 5. Let (cod, s) be a polynomial encoding associated with that solution. Let us recall that instance $u \in I_X$ of the problem X is processed by the system $\Pi(s(u))+cod(u)$.

Let us consider the following deterministic algorithm $\mathcal{A}'$:

Input: `an instance` u `of the decision problem` X
`Construct the system` $\Pi(s(u)) + cod(u)$
`Run algorithm` $\mathcal{A}$ `with input the system` $\Pi(s(u)) + cod(u)$
Output: *Yes* `if` $\Pi(s(u))+cod(u)$ `has an accepting computation,` *No* `otherwise`

Given an instance u of the decision problem $X = (I_X, \theta_X)$, the following assertions are equivalent:

1. $\theta_X(u) = 1$, that is, the answer of problem X to instance u is affirmative.
2. Every computation of $\Pi(s(u)) + cod(u)$ is an accepting computation.
3. The output of the algorithm with input u is *Yes*.

Therefore, algorithm $\mathcal{A}'$ provides a solution of the decision problem X. Bearing in mind that $\mathcal{A}'$ works in polynomial time, we finally deduce that $X \in \mathbf{P}$.

□

8 Conclusions

The classical definition of polarizationless P systems with active membranes makes use of non-cooperative rules and their object evolution rules are of the form $[\, a \rightarrow u\,]_h$, where a is an object and u is a finite multiset of objects. In that context, the capability of these membrane systems to create an exponential workspace in linear time is implemented by means of division rules (for both elementary and non-elementary membranes). It is well known [5] that only tractable problems can be solved in an efficient way by families of such kind of P systems which do not make use of dissolution rules, that is, $\mathbf{P} = \mathbf{PMC}_{\mathcal{DAM}^0(-d,+n)}$ (in the notation from [5], $\mathbf{P} = \mathbf{PMC}_{\mathcal{AM}^0(-d,+n)}$).

In this paper, two new variants are considered. First, by using separation rules inspired on the membrane fission mechanism, instead of division rules in order to create an exponential workspace in linear time. Second, minimal cooperation in object evolution rules is incorporated in polarizationless P systems with active membranes making use of division or separation rules. Object evolution rules with minimal cooperation are of the forms $[a \rightarrow c]_h$, $[ab \rightarrow c]_h$ or $[ab \rightarrow cd]_h$.

The computational efficiency of these models is studied and two main results have been obtained. On the one hand, a polynomial-time and uniform solution to SAT problem by a family of polarizationless P systems with active membranes, minimal cooperation in object evolution rules, without dissolution rules and using only division for elementary membranes, is provided. On the other hand, the limits on efficient computations of polarizationless P systems with active membranes, minimal cooperation in object evolution rules, and using separation rules for elementary membranes and non-elementary membranes, has been established, in the sense that only problems in class $\mathbf{P}$ can be solved by families of such kind of membrane systems in an efficient way.

Consequently, in the framework of polarizationless P systems with active membranes and without dissolution rules, two frontiers of the efficiency have been presented.

- If these membrane systems make use of division rules then passing from non-cooperation to minimal cooperation in object evolution rules amounts passing from non-efficiency to efficiency, that is, $\mathbf{P} = \mathbf{PMC}_{\mathcal{DAM}^0(-d,+n)}$ and $\mathtt{SAT} \in \mathbf{PMC}_{\mathcal{DAM}^0_{mc}(-d,-n)}$
- If these membrane systems make use of minimal cooperation in object evolution rules then passing from separation rules to division rules amounts passing from non-efficiency to efficiency, that is, that is, $\mathbf{P} = \mathbf{PMC}_{\mathcal{SAM}^0_{mc}(+d,+n)}$ and $\mathtt{SAT} \in \mathbf{PMC}_{\mathcal{DAM}^0_{mc}(-d,-n)}$.

It is worth pointing out some remarks regarding to the Păun's conjecture, $\mathbf{P} = \mathbf{PMC}_{\mathcal{DAM}^0(+d,-n)}$. In [5] a key role of the –apparently "innocent"– operation of dissolution rules has been highlighted in the context of computational efficiency of polarizationless P systems with active membranes, assuming that $\mathbf{P} \neq \mathbf{NP}$. Therefore, bearing in mind that $\mathtt{SAT} \in \mathbf{PMC}_{\mathcal{DAM}^0_{mc}(-d,-n)}$, the role

of dissolution rules is now not relevant because in the sense that computationally hard problems can be solved in an efficient way without using these kind of rules. On the other hand, assuming that $\mathbf{P} \neq \mathbf{NP}$, a new partial negative answer to the Păun's conjecture has been obtained

As future work, we propose several research lines related to the computational efficiency of new variants of polarizationless P systems with active membranes.

(a) Membrane systems with membrane separation which make use of classical object evolution rules.
(b) Membrane systems that incorporate minimal cooperation in object evolution rules, removing the restriction about the length of the right-hand side of the rules.
(c) Membrane systems that incorporate an environment with an active role in polarizationless P systems with active membranes through a distinguished alphabet $\mathcal{E}$ similarly to the considered in cell-like P systems with symport/antiport rules (see [7, 8] for details). Then two kind of semantics can be considered: the classical semantics of active membranes or a semantics based on maximal parallelism of the rules except for division or separation rules. Is relevant the role of the environment from a computational complexity point of view?

References

1. A. Alhazov, L. Pan, G. Păun. Trading polarizations for labels in P systems with active membranes, *Acta Informatica*, **41** (2004), 111–144.
2. A. Alhazov, M.J. Pérez–Jiménez. Uniform solution of QSAT using polarizationless active membranes. *Lecture Notes in Computer Science*, **4664** (2007), 122-133.
3. T.H. Cormen, C.E. Leiserson, R.L. Rivest. *An Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, 1994.
4. M.R. Garey, D.S. Johnson. *Computers and Intractability. A guide to the theory of NP-completeness*. W.H. Freeman and Company, New York, 1979.
5. M.A. Gutiérrez–Naranjo, M.J. Pérez–Jiménez, A. Riscos–Núñez, F.J. Romero–Campero. On the power of dissolution in P systems with active membranes. *Lecture Notes in Computer Science*, **3850** (2006), 224–240.
6. M.A. Gutiérrez–Naranjo, M.J. Pérez–Jiménez, A. Riscos–Núñez, F.J. Romero–Campero, A. Romero–Jiménez. Characterizing tractability by cell–like membrane systems. In K.G. Subramanian, K. Rangarajan, M. Mukund (eds.) *Formal models, languages and applications*, World Scientific, Singapore, 2006, pp. 137–154.
7. L.F. Macías-Ramos, M.J. Pérez-Jiménez, A. Riscos-Núñez, L. Valencia-Cabrera. Membrane fission versus cell division: When membrane proliferation is not enough. *Theoretical Computer Science*, **608** (2015), 57–65.
8. L.F. Macías-Ramos, B. Song, L. Valencia-Cabrera, L. Pan, M.J. Pérez-Jiménez. Membrane fission: A computational complexity perspective. *Complexity*, online version 2015 (doi: 10.1002/cplx.21691).
9. Gh. Păun. Attacking **NP**-complete problems. In *Unconventional Models of Computation, UMC'2K* (I. Antoniou, C. Calude, M. J. Dinneen, eds.), Springer-Verlag, 2000, 94-115.

10. Gh. Păun. P systems with active membranes: Attacking **NP**–complete problems. *Journal of Automata, Languages and Combinatorics*, **6**, 1 (2001), 75–90.
11. Gh. Păun. Further twenty six open problems in membrane computing. In M.A. Gutiérrez, A. Riscos, F.J. Romero, D. Sburlan (eds.) *Proceedings of the Third Brainstorming Week on Membrane Computing*, Report RGNC 01/04, Fénix Editora, Sevilla, 2005, pp. 249–262.
12. Gh. Păun, G. Rozenberg, A. Salomaa (eds.). *The Oxford Handbook of Membrane Computing*, Oxford University Press, Oxford, 2010.
13. M.J. Pérez-Jiménez. An approach to computational complexity in Membrane Computing. In G. Mauri, Gh. Păun, M.J. Pérez-Jiménez, G. Rozenberg, A. Salomaa (eds.). *Membrane Computing, International Workshop, WMC5, Milano, Italy, 2004, Selected Papers, Lecture Notes in Computer Science*, **3365** (2005), 85-109.
14. M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini. A polynomial complexity class in P systems using membrane division. *Journal of Automata, Languages and Combinatorics*, **11**, 4 (2006), 423-434. A preliminary version in E. Csuhaj-Varjú, C. Kintala, D. Wotschke, Gy. Vaszil (eds.) *Proceedings of the Fifth International Workshop on Descriptional Complexity of Formal Systems*, DCFS 2003, Budapest, Hungary, July 12-14, 2003, pp. 284-294.
15. M.J. Pérez-Jiménez. A Computational Complexity Theory in Membrane Computing. (invited talk). Membrane Computing, 10th International Workshop, WMC 2009, Curtea de Arges, Romania, August 24-27, 2009, Revised Selected and Invited Papers. *Lecture Notes in Computer Science*, **5957** (2010), 125-148.
16. A.E. Porreca, G. Mauri, C. Zandron. Complexity classes for membrane systems. *Informatique théorique et applications*, **40**, 2 (2006), 141–162.