# Restricted Polarizationless P Systems with Active Membranes: Minimal Cooperation Only Outwards

Luis Valencia-Cabrera, David Orellana-Martín, Miguel Á. Martínez-del-Amor, 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, mdelamor, ariscosn, marper`}`@us.es`

**Summary.** Membrane computing is a computing paradigm providing a class of distributed parallel computing devices of a biochemical type whose process units represent biological membranes. In the cell-like basic model, a hierarchical membrane structure formally described by a rooted tree is considered. It is well known that families of such systems where the number of membranes can only decrease during a computation (for instance by dissolving membranes), can only solve in polynomial time problems in class **P**. *P systems with active membranes* is a variant where membranes play a central role in their dynamics. In the seminal version, membranes have an electrical polarization (positive, negative, or neutral) associated in any instant, and besides being dissolved, they can also replicate by using division rules. These systems are computationally universal, that is, equivalent in power to deterministic Turing machines, and computationally efficient, that is, able to solve computationally hard problems in polynomial time. If polarizations in membranes are removed and dissolution rules are forbidden, then only problems in class **P** can be solved in polynomial time by these systems (even in the case when division rules for non-elementary membranes are permitted). In that framework it has been shown that by considering minimal cooperation (left-hand side of such rules consists of at most two symbols) and minimal production (only one object is produced by the application of such rules) in object evolution rules, such systems provide efficient solutions to **NP**–complete problems. In this paper, minimal cooperation and minimal production in communication rules instead of object evolution rules is studied, and the computational efficiency of these systems is obtained in the case where division rules for non-elementary membranes are permitted.

**Key words:** Membrane Computing, polarizationless P systems with active membranes, cooperative rules, the **P** versus **NP** problem, `SAT` problem.

## 1 Introduction

Membrane Computing is an emergent branch of Natural Computing providing distributed parallel and non-deterministic computing models whose computational devices are called *membrane systems* having units processor called *compartments*. This computing paradigm is inspired by some basic biological features, by the structure and functioning of the living cells, as well as from the cooperation of cells in tissues, organs, and organisms. Celllike membrane systems use the biological membranes arranged hierarchically, inspired from the structure of the cell.

In Membrane Computing, some variants capture the fact that membranes are not at all passive from a biochemistry view, for instance, the passing of a chemical compound through a membrane is often done by a direct interaction with the membrane itself. Some variants of P systems where the central role in their dynamics is played by the membranes have been considered. In these models, the membranes not only directly mediate the evolution and the communication of objects, but they can replicate themselves by means of a division process. Inspired by these features, *P systems with active membranes* [6] were introduced, based on processing multisets by means of non-cooperative rewriting rules, that is, rules where its left-hand side has at most only one object. Specifically, objects evolve inside membranes which can communicate between each other, can dissolve, and moreover (inspired by cellular mitosis process) can replicate by means of division rules. It is assumed that each membrane has associated an electrical polarization in any instant, one of the three possible: positive, negative, or neutral.

P systems with active membranes are computationally complete, that is, any recursively enumerable set of vectors of natural numbers (in particular, each recursively enumerable set of natural numbers) can be generated by such a system [6]. Hence, they are equivalent in power to deterministic Turing machines.

What about the computational efficiency of P systems with active membranes? The key is certainly in the use of division rules, as we can deduce from the so-called *Milano theorem* [13]: *A deterministic P system with active membranes but without membrane division can be simulated by a deterministic Turing machine with a polynomial slowdown.*

However, P systems with active membranes which make use of division rules have the ability to provide efficient solutions to computationally hard problems, by making use of an exponential workspace created in a polynomial time. Specifically, **NP**-complete problems can be solved in polynomial time by families of P systems with active membranes, without dissolution rules and which use division rules only for elementary membranes [6]. Moreover, the class of decision problems which can be solved by families of P systems with active membranes with dissolution rules and which use division for elementary and non-elementary membranes is equal to **PSPACE** [8]. Consequently, the usual framework of P systems with active membranes and electrical polarizations for solving decision problems seems to be too powerful from the computational complexity point of view.

With respect to the computational efficiency, in the classical framework of P system with active membranes, dissolution rules play an “innocent” role as well as division for non-elementary membranes. However, if electrical charges are removed then these kind of rules come to play a relevant role. Specifically, P systems with active membranes and without electrical charges were initially studied in [1, 2] by replacing electrical charges by the ability to change the label of the membranes. In this paper, polarizationless P systems with active membranes where labels of membranes keep unchanged by the application of rules, are considered. In this new framework, if dissolution rules are forbidden then only problems in class **P** can be solved in an efficient way, even in the case that division for non-elementary membranes are permitted [5]. Is the class of polarizationless P systems with active membranes, with dissolution but using only division rules for elementary membranes computationally efficient? If $\mathbf{P} \neq \mathbf{NP}$, which is at all expected, then it is an open question, so-called *Păun's conjecture*.

In the seminal paper where P systems with active membranes were introduced, Gh. Păun says that "*working with non-cooperative rules is natural from a mathematical point of view but from a biochemical point of view this is not only non-necessary, but also non-realistic: most of the chemical reactions involve two or more than two chemical compounds* (*and also produce two or more than two compounds*)". In this context, a restricted cooperation has been considered in the classical framework of polarizationless P systems with active membranes. Specifically, minimal cooperation (the left-hand side and the right-hand side of any rules have, at most, two objects) in object evolution rules, has been previously studied from a computational complexity point of view. A polynomial-time solution to the `SAT` problem by means of families of polarizationless P systems with active membranes, with minimal cooperation in object evolution rules, has been provided [9]. Recently, this result has been improved by considering minimal cooperation in object evolution rules with and additional restriction: the right-hand side of any rules has only one object (called *minimal cooperation and minimal production*) [11]. A relevant fact in these results is the following: dissolution rules and division rules for non-elementary membranes are not necessary to reach the computational efficiency.

In this paper the role of minimal cooperation and minimal production in communication rules instead of object evolution rules, is studied from a complexity point of view. Specifically, by using families of membrane systems which use these syntactical ingredients, a polynomial-time solution to the `SAT` problem is provided but allowing division rules for non-elementary membranes.

The paper is structured as follows. First, some basic notions are recalled and the terminology and notation to be used in the paper is presented. Then, Section 3 introduces the model that will be investigated in this paper: polarizationless P systems with active membranes, with minimal cooperation and minimal production in their communication rules. Section 4 contains the main result of this paper, showing that these systems are capable of solving an **NP**-complete problem in an *efficient* way. Finally, the paper concludes with some final remarks and ideas for future work.

## 2 Preliminaries

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

A *multiset* over an alphabet $\Gamma$ is an ordered pair $(\Gamma, f)$ where $f$ is a mapping from $\Gamma$ 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. Let $m_1 = (\Gamma, f_1)$, $m_2 = (\Gamma, f_2)$ be multisets over $\Gamma$, then the union of $m_1$ and $m_2$, denoted by $m_1 + m_2$, is the multiset $(\Gamma, g)$, where $g(x) = f_1(x) + f_2(x)$ for each $x \in \Gamma$. We denote by $M_f(\Gamma)$ the set of all multisets over $\Gamma$.

### 2.1 Graphs and trees

Let us recall some notions related with graph theory (see [3] for details). An *undirected graph* is an ordered pair $(V, E)$ where $V$ is a set whose elements are called nodes or vertices and $E = \{\{x, y\} \mid x \in V, y \in V, x \neq y\}$ whose elements are called *edges*. A *path* of length $k \geq 1$ from a node $u$ to a node $v$ in a graph $(V, E)$ is a finite sequence $(x_0, x_1, \ldots, x_k)$ of nodes such that $x_0 = u$, $x_k = v$ and $\{x_i, x_{i+1}\} \in E$. If $k \geq 2$ and $x_0 = x_k$ then we say that the path is a *cycle* of the graph. A graph with no cycle is said to be *acyclic*. An undirected graph is *connected* if there exist paths between every pair of nodes.

A *rooted tree* is a a connected, acyclic, undirected graph in which one of the vertices (called *the root of the tree*) is distinguished from the others. 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*.

### 2.2 The Cantor pairing function

The Cantor pairing function encodes pairs of natural numbers by single natural numbers, and it is defined as follows: for each $n, p \in \mathbb{N}$

$$\langle n, p \rangle = \frac{(n+p)(n+p+1)}{2} + n$$

The Cantor pairing function is a primitive recursive function and bijective from $\mathbb{N} \times \mathbb{N}$ onto $\mathbb{N}$. Then, for each $t \in \mathbb{N}$ there exist unique natural numbers $n, p \in \mathbb{N}$ such that $t = \langle n, p \rangle$.

### 2.3 Decision problems and languages

A decision problem $X$ is an ordered pair $(I_X, \theta_X)$, where $I_X$ is a language over a finite alphabet $\Sigma_X$ and $\theta_X$ is a total Boolean function over $I_X$. The elements of $I_X$ are called *instances* of the problem $X$. Each decision problem $X$ has associated a language $L_X$ over the alphabet $\Sigma_X$ as follows: $L_X = \{u \in {\Sigma_X}^* \mid \theta_X(u) = 1\}$. Conversely, every language $L$ over an alphabet $\Sigma$ has associated a decision problem $X_L = (I_{X_L}, \theta_{X_L})$ as follows: $I_{X_L} = \Sigma^*$ and $\theta_{X_L}(u) = 1$ if and only if $u \in L$. Therefore, given a decision problem $X$ we have $X_{L_X} = X$, and given a language $L$ over an alphabet $\Sigma$ we have $L_{X_L} = L$. Then, solving a decision problem can be expressed equivalently as the task of recognizing the language associated with it.

### 2.4 Recognizer membrane systems

Recognizer membrane systems were introduced in [7] and they provide a natural framework to solve decision problems. This kind of systems are characterized by the following features: (a) the working alphabet $\Gamma$ has two distinguished objects `yes` and `no`; (b) there exists an input membrane and an input alphabet $\Sigma$ strictly contained in $\Gamma$; (c) the initial contents of the membranes are multisets over $\Gamma \setminus \Sigma$; (d) all computations halt; and (e) for each computation, 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.

Given a recognizer membrane system, $\Pi$, for each multiset $m$ over the input alphabet $\Sigma$ we denote by $\Pi + m$ the membrane system $\Pi$ with input multiset $m$, that is in the initial configuration of that system, the multiset $m$ is added to the initial content of the input membrane. Thus, in a recognizer membrane system, $\Pi$, there exists an initial configuration associated with each multiset $m \in M_f(\Sigma)$.

## 3 Minimal cooperation and minimal production in communication rules

**Definition 1.** *A polarizationless P system with active membranes, with* <u>*simple object evolution rules*</u>*, without dissolution, with division rules for* <u>*elementary and non-elementary membranes*</u>*, and which makes use of* <u>*minimal cooperation*</u> *and* <u>*minimal production*</u> *in* <u>*send-out communication*</u> *rules, is a tuple*

$$\Pi = (\Gamma, \Sigma, H, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, \mathcal{R}, i_{in}, i_{out})$$

*where:*

- *$\Gamma$ is a finite alphabet whose elements are called objects and contains two distinguished objects* `yes` *and* `no`*.*
- *$\Sigma \subsetneq \Gamma$ is the input alphabet.*

- *$H$ is a finite alphabet such that $H \cap \Gamma = \emptyset$ whose elements are called labels.*
- *$q \geq 1$ is the degree of the system.*
- *$\mu$ is a labelled rooted tree (called membrane structure) consisting of $q$ nodes injectively labelled by elements of $H$ (the root of $\mu$ is labelled by $r_\mu$).*
- *$\mathcal{M}_1, \ldots, \mathcal{M}_q$ are multisets over $\Gamma \setminus \Sigma$.*
- *$\mathcal{R}$ is a finite set of rules, of the following forms:*
  - $(a_0)$ $[\, a \to b \,]_h$, *where* $h \in H$, $a, b \in \Gamma$, $u \in M_f(\Gamma)$ (simple object evolution rules).
  - $(b_0)$ $a\,[\ \ ]_h \to [\, b \,]_h$, *where* $h \in H \setminus \{r_\mu\}$, $a, b, c \in \Gamma$ (send–in communication rules).
  - $(c_0)$ $[\, a\, b \,]_h \to c\, [\ \ ]_h$, *where* $h \in H$, $a, b \in \Gamma$ (send–out communication rules with minimal cooperation and minimal production).
  - $(d_0)$ $[\, a \,]_h \to b$, *where* $h \in H \setminus \{i_{out}, r_\mu\}$, $a, b \in \Gamma$ (dissolution rules).
  - $(e_0)$ $[\, a \,]_h \to [\, b \,]_h\ [\, c \,]_h$, *where* $h \in H \setminus \{i_{out}, r_\mu\}$, $a, b, c \in \Gamma$ *and $h$ is the label of an elementary membrane $\mu$* (division rules for elementary membranes).
  - $(f_0)$ $[\, [\ ]_{h_1} [\ ]_{h_2} \,]_{h_0} \to [\, [\ ]_{h_1} \,]_{h_0}\ [\, [\ ]_{h_2} \,]_{h_0}$, *where* $h_0, h_1, h_2 \in H$ *and* $h_0 \neq r_\mu$ (division rules for non-elementary membranes).
- *$i_{in} \in H$, $i_{out} \in H \cup \{env\}$ (if $i_{out} \in H$ then $i_{out}$ is the label of a leaf of $\mu$).*

In a similar way is defined the concept of "*polarizationless P system with active membranes, with simple object evolution rules, without dissolution, with division rules for elementary and non-elementary membranes, and which makes use of minimal cooperation and minimal production in send-in communication rules* ". The only difference concerns rules of type $(b_0)$ and $(c_0)$. In this case are, respectively:

$(b'_0)$ $a\, b\, [\ \ ]_h \to [\, c \,]_h$ for $h \in H \setminus \{r_\mu\}$, $a, b \in \Gamma$ *(send–in communication rules with minimal cooperation and minimal production).*

$(c'_0)$ $[\, a \,]_h \to b\, [\ \ ]_h$ for $h \in H$, $a, b, c \in \Gamma$ *(send–out communication rules).*

The semantics of this kind of P systems follows the usual principles of P systems with active membranes [6].

We denote by $\mathcal{DAM}^0(+e_s, mcmp_{out}, -d, +n)$ (respectively, $\mathcal{DAM}^0(+e_s, mcmp_{in}, -d, +n)$) the class of all recognizer polarizationless P system with active membranes, with simple object evolution rules, without dissolution, with division rules for elementary and non-elementary membranes, which make use of minimal cooperation and minimal production in send-out (respectively, send-in) communication rules.

## 4 Solving SAT in $\mathcal{DAM}^0(+e_s, mcmp_{out}, -d, +n)$

In this section, a polynomial-time solution to SAT problem, is explicitly given in the framework of recognizer polarizationless P systems with active membranes with simple object evolution rules, without dissolution and with division rules for elementary and non-elementary membranes which make use of minimal cooperation and minimal production in send-in communication rules. For that, a family $\mathbf{\Pi} = \{\Pi(t) \mid t \in \mathbb{N}\}$ of recognizer P systems from $\mathcal{DAM}^0(+e_s, mcmp_{out}, -d, +n)$ will be presented.

### 4.1 Description of a solution to SAT problem in $\mathcal{DAM}^0(+e_s, mcmp_{out}, -d, +n)$

For each $n, p \in \mathbb{N}$, we consider the recognizer P system

$$\Pi(\langle n, p\rangle) = (\Gamma, \Sigma, H, \mu, \mathcal{M}_0, \mathcal{M}_1, \mathcal{M}_2, \mathcal{R}, i_{in}, i_{out})$$

from $\mathcal{DAM}^0(+e_s, mcmp_{out}, -d, +n)$, defined as follows:

**(1)** Working alphabet:

$$\begin{aligned}\Gamma = \Sigma \cup \{\texttt{yes}, \texttt{no}, \#\} \cup \{a_{i,k} \mid 1 \le i \le n \wedge 1 \le k \le 2i-1\} \cup \\ \{\alpha_k \mid 0 \le k \le 4np+n+2p\} \cup \{\beta_k \mid 0 \le k \le 4np+n+2p+1\} \cup \\ \{\gamma_k \mid 0 \le k \le 4np+n\} \cup \\ \{t_{i,k}, f_{i,k} \mid 1 \le i \le n \wedge 2i-1 \le k \le 2n+2p-1\} \cup \\ \{T_i, F_i \mid 1 \le i \le n\} \cup \{c_j \mid 1 \le j \le p\} \cup \\ \{c_{j,k} \mid 1 \le j \le p \wedge 0 \le k \le np-1\} \cup \{d_j \mid 1 \le j \le p\} \cup \\ \{x_{i,j,k}, \overline{x}_{i,j,k}, x^*_{i,j,k} \mid 1 \le i \le n \wedge 1 \le j \le p \wedge \\ 1 \le k \le n+2np+n(j-1)+(i-1)\}\end{aligned}$$

where the input alphabet is $\Sigma = \{x_{i,j,0}, \overline{x}_{i,j,0}, x^*_{i,j,0} \mid 1 \le i \le n \wedge 1 \le j \le p\}$;

**(2)** $H = \{0, 1, 2\}$;

**(3)** Membrane structure: $\mu = [\,[\,[\;\;]_2\,]_1\,]_0$, that is, $\mu = (V, E)$ where $V = \{0, 1, 2\}$ and $E = \{(0,1)(1,2)\}$

**(4)** Initial multisets: $\mathcal{M}_0 = \{\alpha_0, \beta_0\}$, $\mathcal{M}_1 = \emptyset$, $\mathcal{M}_2 = \{\gamma_0\} \cup \{a_{i,1}, T_i^p, F_i^p \mid 1 \le i \le n\}$.

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

**5.1** Counters for synchronize the answer of the system.

$$\begin{aligned}&[\,\alpha_k \longrightarrow \alpha_{k+1}\,]_0, \text{ for } 0 \le k \le 4np+n+2p-1\\ &[\,\beta_k \longrightarrow \beta_{k+1}\,]_0, \text{ for } 0 \le k \le 4np+n+2p\\ &[\,\gamma_k \longrightarrow \gamma_{k+1}\,]_2, \text{ for } 0 \le k \le 4np+n-1\end{aligned}$$

**5.2** Rules to generate $2^n$ membranes labelled by 1 and $2^n$ membranes labelled by 2 (these encoding all possible truth assignment of $n$ variables of the input formula).

$$[\,a_{i,2i-1}\,]_2 \longrightarrow [\,t_{i,i}\,]_2\,[\,f_{i,i}\,]_2, \text{ for } 1 \le i \le n$$

$$[\,a_{i,j} \longrightarrow a_{i,j+1}\,]_2, \text{ for } 2 \le i \le n, 1 \le j \le 2i-2$$

$$[\,[\;\;]_2\,[\;\;]_2\,]_1 \longrightarrow [\,[\;\;]_2\,]_1\;\;[\,[\;\;]_2\,]_1$$

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

**5.3** Rules to produce exactly $p$ copies of each truth assignment encoded by membranes labelled by 2.

$$\left.\begin{array}{l} [t_{i,2jn}\ F_i]_2 \longrightarrow t_{i,2jn+1}\ [\ ]_2 \\ [f_{i,2jn}\ T_i]_2 \longrightarrow f_{i,2jn+1}\ [\ ]_2 \\ t_{i,(2j+1)n}\ [\ \ ]_2 \longrightarrow [\ t_{i,(2j+1)n+1}\ ]_2 \\ f_{i,(2j+1)n}\ [\ \ ]_2 \longrightarrow [\ f_{i,(2j+1)n+1}\ ]_2 \end{array}\right\},\ \text{for } 1 \le i \le n,\ 1 \le j \le p-1$$

$$\left.\begin{array}{l} [t_{i,2np}\ F_i]_2 \longrightarrow \#\ [\ ]_2 \\ [f_{i,2np}\ T_i]_2 \longrightarrow \#\ [\ ]_2 \end{array}\right\},\ \text{for } 1 \le i \le n$$

$$\left.\begin{array}{l} [\ t_{i,(2j+1)n+k} \longrightarrow t_{i,(2j+1)n+k+1}\ ]_2 \\ [\ f_{i,(2j+1)n+k} \longrightarrow f_{i,(2j+1)n+k+1}\ ]_2 \\ [\ t_{i,2jn+k} \longrightarrow t_{i,2jn+k+1}\ ]_1 \\ [\ f_{i,2jn+k} \longrightarrow f_{i,2jn+k+1}\ ]_1 \end{array}\right\},\ \text{for } \begin{array}{l} 1 \le i \le n, \\ 1 \le j \le p-1, \\ 1 \le k \le n-1 \end{array}$$

**5.4** Rules to prepare the input formula for check clauses:

$$\left.\begin{array}{l} [\ x_{i,j,k} \longrightarrow x_{i,j,k+1}\ ]_2 \\ [\ \overline{x}_{i,j,k} \longrightarrow \overline{x}_{i,j,k+1}\ ]_2 \\ [\ x^*_{i,j,k} \longrightarrow x^*_{i,j,k+1}\ ]_2 \end{array}\right\},\ \text{for } \begin{array}{l} 1 \le i \le n, \\ 1 \le j \le p, \\ 0 \le k \le 2np+n+n(j-1)+(i-1)-1 \end{array}$$

**5.5** Rules implementing the first checking stage.

$$\left.\begin{array}{l} [T_i\ x_{i,j,2np+n+n(j-1)+(i-1)}]_2 \longrightarrow c_{j,0}[\ \ ]_2 \\ [T_i\ \overline{x}_{i,j,2np+n+n(j-1)+(i-1)}]_2 \longrightarrow \#[\ \ ]_2 \\ [T_i\ x^*_{i,j,2np+n+n(j-1)+(i-1)}]_2 \longrightarrow \#[\ \ ]_2 \\ [F_i\ x_{i,j,2np+n+n(j-1)+(i-1)}]_2 \longrightarrow \#[\ \ ]_2 \\ [F_i\ \overline{x}_{i,j,2np+n+n(j-1)+(i-1)}]_2 \longrightarrow c_{j,0}[\ \ ]_2 \\ [F_i\ x^*_{i,j,2np+n+n(j-1)+(i-1)}]_2 \longrightarrow \#[\ \ ]_2 \end{array}\right\},\ \text{for } \begin{array}{l} 1 \le i \le n, \\ 1 \le j \le p \end{array}$$

**5.6** Rules implementing the second checking stage.

$$[\ c_{j,k} \longrightarrow c_{j,k+1}\ ]_1,\ \text{for } 1 \le j \le p,\ 0 \le k \le np-2$$
$$c_{j,np-1}[\ \ ]_2 \longrightarrow [c_j]_2,\ \text{for } 1 \le j \le p$$
$$[\ \gamma_{4np+n}\ c_1]_2 \longrightarrow d_1[\ \ ]_2$$
$$\left.\begin{array}{l} [\ d_j\ c_{j+1}\ ]_2 \longrightarrow d_{j+1}[\ \ ]_2 \\ d_j\ [\ \ ]_2 \longrightarrow [\ d_j\ ]_2 \end{array}\right\},\ \text{for } 1 \le j \le p-1$$

**5.7** Rules to provide the correct answer of the system.

$$[\ d_p\ ]_1 \longrightarrow d_p[\ \ ]_1$$
$$[\ \alpha_{4np+n+2p}\ d_p\ ]_0 \longrightarrow \texttt{yes}\ [\ \ ]_0$$
$$[\ \alpha_{4np+n+2p}\ \beta_{4np+n+2p+1}\ ]_0 \longrightarrow \texttt{no}\ [\ \ ]_0$$

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

## 5 A formal verification

Let $\varphi = C_1 \wedge \ldots \wedge C_p$ an instance of SAT problem consisting of $p$ clauses $C_j = l_{j,1} \vee \ldots \vee l_{j,r_j}$, $1 \le j \le p$, where $Var(\varphi) = \{x_1, \ldots, x_n\}$, and $l_{j,k} \in$

$\{x_i, \neg x_i | 1 \leq i \leq n\}$, $1 \leq j \leq p$, $1 \leq k \leq r_j$. Let us asume that the number of variables, $n$, and the number of clauses, $p$, of $\varphi$, 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

$$cod(\varphi) = \{x_{i,j,0} | x_i \in C_j\} \cup \{\overline{x}_{i,j,0} | \neg x_i \in C_j\} \cup \{x^*_{i,j,0} | x_i \notin C_j, \neg x_i \notin C_j\}$$

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:

$$cod(\varphi) = \begin{pmatrix} x_{1,1,0} & x_{2,1,0} & \overline{x}_{3,1,0} & x^*_{4,1,0} \\ x^*_{1,2,0} & \overline{x}_{2,2,0} & x^*_{3,2,0} & x_{4,2,0} \\ x^*_{1,3,0} & \overline{x}_{2,3,0} & x_{3,3,0} & \overline{x}_{4,3,0} \end{pmatrix}$$

That is, $j$-th row $(1 \leq j \leq p)$ represents the $j$-th clause $C_j$ of $\varphi$. We denote $(cod(\varphi))^p_j$ 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,

$$cod(\varphi)^p_2 = \begin{pmatrix} x^*_{1,2,0} & \overline{x}_{2,2,0} & x^*_{3,2,0} & x_{4,2,0} \\ x^*_{1,3,0} & \overline{x}_{2,3,0} & x_{3,3,0} & \overline{x}_{4,3,0} \end{pmatrix}$$

We denote $(cod_k(\varphi))^p_j)$ the code $cod(\varphi)^p_j$ when the third index of the variables equal 3. For instance: row to $p$-th row. For instance,

$$cod_3(\varphi)^p_2 = \begin{pmatrix} x^*_{1,2,3} & \overline{x}_{2,2,3} & x^*_{3,2,3} & x_{4,2,3} \\ x^*_{1,3,3} & \overline{x}_{2,3,3} & x_{3,3,3} & \overline{x}_{4,3,3} \end{pmatrix}$$

We denote $(cod'_k(\varphi))^p_j)$ the code $cod(\varphi)^p_j$ when the third index of the variables equal 3. For instance: row to $p$-th row. For instance,

$$cod'_3(\varphi)^p_2 = \begin{pmatrix} {x^*}'_{1,2,3} & \overline{x}'_{2,2,3} & {x^*}'_{3,2,3} & x'_{4,2,3} \\ {x^*}'_{1,3,3} & \overline{x}'_{2,3,3} & x'_{3,3,3} & \overline{x}'_{4,3,3} \end{pmatrix}$$

We denote $(cod^*(\varphi))^p_j)$ the code $cod(\varphi)^p_j$ when the third index does not exist. For instance: row to $p$-th row. For instance,

$$cod^*(\varphi)^p_2 = \begin{pmatrix} {x^*}_{1,2} & \overline{x}_{2,2} & x^*_{3,2} & x_{4,2} \\ {x^*}_{1,3} & \overline{x}_{2,3} & x_{3,3} & \overline{x}_{4,3} \end{pmatrix}$$

The Boolean formula $\varphi$ 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 separation rules, beside other rules that make a "simulation" of division rules, we get all truth assignments for the variables $\{x_1, \ldots, x_n\}$ associated with $\varphi$ are produced. Specifically, $2^n$ membranes labelled by 1 and $2^n$ labelled by 2 are generated. Each of the former ones encodes a truth assignment. This stage takes exactly $n+2np$ steps, being $n$ the number of variables of $\varphi$.

- *First Checking stage*: checking whether or not each clause of the input formula $\varphi$ is satisfied by the truth assignments generated in the previous stage, encoded by each membrane labelled by 2. This stage takes exactly $np$ steps, being $n$ the number of the variables and $p$ the number of clauses of $\varphi$.
- *Second Checking stage*: checking whether or not all clauses of the input formula $\varphi$ are satisfied by some truth assignment encoded by a membrane labelled by 2. This stage takes exactly $np + 2p$ steps, being $n$ the number of variables and $p$ the number of clauses of $\varphi$.
- *Output stage*: the system sends to the environment the right answer according to the results of the previous stage. This stage takes 2 steps if the answer is `yes` and 3 steps if the answer is `no`.

### 5.1 Generation stage

Through this stage, all the different truth assignments for the variables associated with the Boolean formula $\varphi$ will be generated within membranes labelled by 2, by the applications of rules from **5.2** and **5.3**. In the first $2n$ steps, $2^n$ membranes labelled by 1 and $2^n$ membranes labelled by 2, alternating between the division of membranes labelled by 2 (in odd steps) and the division of membranes labelled by 1 (in even steps).

**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 $cod(\varphi)$.*

$(a_0)$ *For each $2k$ $(0 \leq k \leq n-1)$ at configuration $\mathcal{C}_{2k}$ we have the following:*
- $\mathcal{C}_{2k}(0) = \{\alpha_{2k}, \beta_{2k}\}$
- *There are $2^k$ empty membranes labelled by 1.*
- *There are $2^k$ membranes labelled by 2 such that each of them contains*
  - ⋆ *the input multiset $cod_{2k}(\varphi)$;*
  - ⋆ *an object $\gamma_{2k}$; and*
  - ⋆ *$p$ copies of every $T_i$ and $F_i$, $1 \leq i \leq n$.*
  - ⋆ *objects $a_{i,2k+1}$, $k+1 \leq i \leq n$; and*
  - ⋆ *a different subset $\{r_{1,j}, \ldots, r_{k,j}\}$, $k+1 \leq j \leq 2k$, being $r \in \{t, f\}$.*

$(a_1)$ *For each $2k+1$ $(0 \leq j \leq n-1)$ at configuration $\mathcal{C}_{2k+1}$ we have the following:*
- $\mathcal{C}_{2k+1}(0) = \{\alpha_{2k+1}, \beta_{2k+1}\}$
- *There are $2^k$ empty membranes labelled by 1.*
- *There are $2^{k+1}$ membranes labelled by 2 such that each of them contains*
  - ⋆ *the input multiset $cod_{2k+1}(\varphi)$;*
  - ⋆ *an object $\gamma_{2k+1}$; and*
  - ⋆ *$p$ copies of every $T_i$ and $F_i$, $1 \leq i \leq n$.*
  - ⋆ *objects $a_{i,2(k+1)}$, $k+1 \leq i \leq n$; and*
  - ⋆ *a different subset $\{r_{1,j}, \ldots, r_{k+1,j}\}$, $k+1 \leq j \leq 2k+1$, being $r \in \{t, f\}$.*

$(b)$ *$\mathcal{C}_{2n}(0) = \{\alpha_{2n}, \beta_{2n}\}$, and in $\mathcal{C}_{2n}$ there are $2^n$ empty membranes labelled by 1; and $2^n$ membranes labelled by 2, such that each of them contains the input multiset $cod_{2n}(\varphi)$, $p$ copies of every $T_i$ and $F_i$ $(1 \leq i \leq n)$, an object $\gamma_{2n}$ and a different subset of objects $r_{i,2n+1-i}$, $1 \leq i \leq n$.*

*Proof.* $(a)$ is going to be demonstrated by induction on $k$

- The base case $k = 0$ is trivial because:

  $(a_0)$ at the initial configuration $\mathcal{C}_0$ we have: $\mathcal{C}_0(0) = \{\alpha_0, \beta_0\}$ and there exists a single empty membrane labelled by 1 containing ; and a single membrane labelled by 2 containing the input multiset $cod(\varphi)$, an object $\gamma_0$, $p$ copies of $T_i$ and $F_i$, being $1 \le i \le n$, the objects $a_{1,1}, \ldots, a_{n,1}$. Then, configuration $\mathcal{C}_0$ yields configuration $\mathcal{C}_1$ by applying the rules:

  $[\ a_{1,1}\ ]_2 \to [\ t_{1,1}\ ]_2\ [\ f_{1,1}\ ]_2$
  $[\ a_{i,1} \to a_{i,2}\ ]_2$ , for $k+1 \le i \le n$
  $[\ \alpha_0 \to \alpha_1\ ]_0$
  $[\ \beta_0 \to \beta_1\ ]_0$
  $[\ \gamma_0 \to \gamma_1\ ]_2$
  $\left.\begin{array}{l} [\ x_{i,j,0} \to x_{i,j,1}\ ]_2 \\ [\ \overline{x}_{i,j,0} \to \overline{x}_{i,j,1}\ ]_2 \\ [\ x^*_{i,j,0} \to x^*_{i,j,1}\ ]_2 \end{array}\right\}$ for $1 \le i \le n, 1 \le j \le p$

  $(a_1)$ at $\mathcal{C}_1$ we have $\mathcal{C}_1(0) = \{\alpha_1, \beta_1\}$ and there exists a single empty membrane labelled by 1; and two membranes labelled by 2 containing the input multiset $cod_1(\varphi)$, an object $\gamma_1$, $p$ copies of $T_i$ and $F_i$, being $1 \le i \le n$, the objects $a_{2,2}, \ldots, a_{n,2}$ and one with the object $t_{1,1}$ and the other one with the object $f_{1,1}$. Then, the configuration $\mathcal{C}_1$ yields configuration $\mathcal{C}_2$ by applying the rules:

  $[\ t_{1,1} \to t_{1,2}\ ]_2$
  $[\ f_{1,1} \to f_{1,2}\ ]_2$
  $[\ [\quad ]_2\ [\quad ]_2 ]_1 \to [\ [\quad ]_2 ]_1\ [\ [\quad ]_2 ]_1$
  $[a_{i,2} \to a_{i,3}\ ]_2$ , for $2 \le i \le n$
  $[\ \alpha_1 \to \alpha_2\ ]_0$
  $[\ \beta_1 \to \beta_2\ ]_0$
  $[\ \gamma_1 \to \gamma_2\ ]_2$
  $\left.\begin{array}{l} [\ x_{i,j,1} \to x_{i,j,2}\ ]_2 \\ [\ \overline{x}_{i,j,1} \to \overline{x}_{i,j,2}\ ]_2 \\ [\ x^*_{i,j,1} \to x^*_{i,j,2}\ ]_2 \end{array}\right\}$ for $1 \le i \le n, 1 \le j \le p$

  Thus, $\mathcal{C}_2(0) = \{\alpha_2, \beta_2\}$, and there exist two empty membranes labelled by 1; and two membranes labelled by 2 containing the input multiset $cod_2(\varphi)$, an object $\gamma_2$, $p$ copies of $T_i$ and $F_i$, being $1 \le i \le n$, the objects $a_{2,3}, \ldots, a_{n,3}$ and one with the object $t_{1,2}$ and the other one with the object $f_{1,2}$. Hence, the result holds for $k = 1$.

- Supposing, by induction, result is true for $k$ $(0 \le k \le n-1)$
  - $\mathcal{C}_{2k}(0) = \{\alpha_{2k}, \beta_{2k}\}$
  - In $\mathcal{C}_{2k}$ there are $2^k$ empty membranes labelled by 1.
  - In $\mathcal{C}_{2k}$ there are $2^k$ membranes labelled by 2 such that each of them contains
    - ⋆ the input multiset $cod_{2k}(\varphi)$;
    - ⋆ an object $\gamma_{2k}$;
    - ⋆ $p$ copies of $T_i$ and $F_i$, $1 \le i \le n$.
    - ⋆ objects $a_{i,2k+1}$, $k+1 \le i \le n$; and

  - $\star$ a different subset $\{r_{1,j}, \dots, r_{k,j}\}$, $k+1 \le j \le 2k$, being $r \in \{t, f\}$.

  Then, configuration $\mathcal{C}_{2k}$ yields configuration $\mathcal{C}_{2k+1}$ by applying the rules:

$$
\begin{array}{l}
[\ a_{k,2k+1}\ ]_2 \to [\ t_{k,k}\ ]_2\ [\ f_{k,k}\ ]_2 \\
[\ a_{i,2k+1} \to a_{i,2k+2}\ ]_2 \text{ , for } k+1 \le i \le n \\
\left.\begin{array}{l} [\ t_{i,j} \to t_{i,j+1}\ ]_2 \\ [\ f_{i,j} \to f_{i,j+1}\ ]_2 \end{array}\right\} \text{for } 1 \le i \le k-1, k+1 \le j \le 2k \\
[\ \alpha_{2k} \to \alpha_{2k+1}\ ]_0 \\
[\ \beta_{2k} \to \beta_{2k+1}\ ]_0 \\
[\ \gamma_{2k} \to \gamma_{2k+1}\ ]_2 \\
\left.\begin{array}{l} [\ x_{i,j,2k} \to x_{i,j,2k+1}\ ]_2 \\ [\ \overline{x}_{i,j,1} \to \overline{x}_{i,j,2k+1}\ ]_2 \\ [\ x^*_{i,j,1} \to x^*_{i,j,2k+1}\ ]_2 \end{array}\right\} \text{for } 1 \le i \le n, 1 \le j \le p
\end{array}
$$

  Therefore, the following holds

- $\mathcal{C}_{2k+1}(0) = \{\alpha_{2k+1}, \beta_{2k+1}\}$
- In $\mathcal{C}_{2k+1}$ there are $2^k$ empty membranes labelled by 1.
- In $\mathcal{C}_{2k+1}$ there are $2^{k+1}$ membranes labelled by 2 such that each of them contains
  - $\star$ the input multiset $cod_{2k+1}(\varphi)$;
  - $\star$ an object $\gamma_{2k+1}$;
  - $\star$ $p$ copies of $T_i$ and $F_i$, $1 \le i \le n$.
  - $\star$ objects $a_{i,2(k+1)}$, $k+1 \le i \le n$; and
  - $\star$ a different subset $\{r_{1,j}, \dots, r_{k+1,j}\}$, $k+1 \le j \le 2k+1$, being $r \in \{t, f\}$.

  Then, configuration $\mathcal{C}_{2k+1}$ yields configuration $\mathcal{C}_{2(k+1)}$ by applying the rules:

$$
\begin{array}{l}
\left.\begin{array}{l} [\ t_{i,j} \to t_{i,j+1}\ ]_2 \\ [\ f_{i,j} \to f_{i,j+1}\ ]_2 \end{array}\right\} \text{for } 1 \le i \le k+1, k+1 \le j \le 2k+1 \\
[\ [\quad ]_2\ [\quad ]_2 ]_1 \to [\ [\quad ]_2 ]_1\ [\ [\quad ]_2 ]_1 \\
[\ a_{i,2(k+1)} \to a_{i,2(k+1)+1}\ ]_2 \text{ , for } k+1 \le i \le n \\
[\ \alpha_{2k+1} \to \alpha_{2(k+1)}\ ]_0 \\
[\ \beta_{2k+1} \to \beta_{2(k+1)}\ ]_0 \\
[\ \gamma_{2k+1} \to \gamma_{2(k+1)}\ ]_2 \\
\left.\begin{array}{l} [\ x_{i,j,2k+1} \to x_{i,j,2k+2}\ ]_2 \\ [\ \overline{x}_{i,j,2k+1} \to \overline{x}_{i,j,2k+2}\ ]_2 \\ [\ x^*_{i,j,2k+1} \to x^*_{i,j,2k+2}\ ]_2 \end{array}\right\} \text{for } 1 \le i \le n, 1 \le j \le p
\end{array}
$$

  Therefore, the following holds

- $\mathcal{C}_{2(k+1)}(0) = \{\alpha_{2(k+1)}, \beta_{2(k+1)}\}$
- In $\mathcal{C}_{2(k+1)}$ there are $2^{k+1}$ empty membranes labelled by 1.
- In $\mathcal{C}_{2(k+1)}$ there are $2^{k+1}$ membranes labelled by 2 such that each of them contains
  - $\star$ the input multiset $cod_{2(k+1)}(\varphi)$;
  - $\star$ an object $\gamma_{2(k+1)}$;
  - $\star$ $p$ copies of $T_i$ and $F_i$, $1 \le i \le n$.
  - $\star$ objects $a_{i,2(k+1)+1}$, $k+1 \le i \le n$; and
  - $\star$ a different subset $\{r_{1,j}, \dots, r_{k+1,j}\}$, $k+1 \le j \le 2(k+1)+1$.

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

- In order to prove $(b)$ it is enough to notice that, on the one hand, from $(a)$ configuration $\mathcal{C}_{2n-1}$ holds:
  - $\mathcal{C}_{2n-1}(0) = \{\alpha_{2n-1}, \beta_{2n-1}\}$
  - In $\mathcal{C}_{2n-1}$ there are $2^{n-1}$ empty membranes labelled by 1.
  - In $\mathcal{C}_{2n-1}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ the input multiset $cod_{2n-1}p(\varphi)$;
    - ⋆ an object $\gamma_{2n-1}$;
    - ⋆ $p$ copies of $T_i$ and $F_i$, $1 \le i \le n$; and
    - ⋆ a different subset of objects $r_{i,2n-i}$, $1 \le i \le n$.

  Then, configuration $\mathcal{C}_{2n-1}$ yields $\mathcal{C}_{2n}$ by applying the rules:

$$\left.\begin{array}{l} [\ t_{i,2n-i} \to t_{i,2n+1-i}\ ]_2 \\ [\ f_{i,2n-i} \to f_{i,2n+1-1}\ ]_2 \end{array}\right\} \text{for } 1 \le i \le n$$
$$[\ [\quad ]_2\ [\quad ]_2 ]_1 \to [\ [\quad ]_2 ]_1\ [\ [\quad ]_2 ]_1$$
$$[\ \alpha_{2n-1} \to \alpha_{2n}\ ]_0$$
$$[\ \beta_{2n-1} \to \beta_{2n}\ ]_0$$
$$[\ \gamma_{2n-1} \to \gamma_{2n}\ ]_2$$
$$\left.\begin{array}{l} [\ x_{i,j,2n-1} \to x_{i,j,2n}\ ]_2 \\ [\ \overline{x}_{i,j,2n-1} \to \overline{x}_{i,j,2n}\ ]_2 \\ [\ x^*_{i,j,2n-1} \to x^*_{i,j,2n}\ ]_2 \end{array}\right\} \text{for } 1 \le i \le n, 1 \le j \le p$$

  Then, we have $\mathcal{C}_{2n}(0) = \{\alpha_{2n}, \beta_{2n}\}$, and there exist $2^n$ empty membranes labelled by 1; and $2^n$ membranes labelled by 2 containing the input multiset $cod_{2n}(\varphi)$, an object $\gamma_{2n}$, $p$ copies of $T_i$ and $F_i$, being $1 \le i \le n$ and a different multiset of objects $r_{i,2n+1-i}$, being $1 \le i \le n$.

□

When the tree structure is created, we start assigning a truth assignment to each branch. It is executed in the next $2np-n$ steps. The last $n$ steps are different from the previous ones, so they deserve another proposition of the following one.

**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 $cod(\varphi)$.*

$(a_0)$ *For each $k$ $(1 \le k \le n)$ and $l$ $(0 \le l \le p-1)$ at configuration $\mathcal{C}_{2n+2ln+k}$ we have the following:*
  - $\mathcal{C}_{2n+2ln+k}(0) = \{\alpha_{2n+2ln+k}, \beta_{2n+2ln+k}\}$
  - *There are $2^n$ membranes labelled by 1 such that each of them contains a different subset of objects $r_{i,2n+2ln+k-i+1}$, $1 \le i \le k$, being $r \in \{t, f\}$.*
  - *There are $2^n$ membranes labelled by 2 such that each of them contains*
    - ⋆ *the input multiset $cod_{2n+2ln+k}(\varphi)$;*
    - ⋆ *an object $\gamma_{2n+2ln+k}$;*
    - ⋆ *$p$ copies of every $T_i$ and $F_i$, $1 \le i \le n$ if the truth assignment associated to the branch contains its corresponding $t_i$ or $f_i$ object; otherwise, there are $p-l$ copies if $k+1 \le i \le n$, $p-l-1$ otherwise; and*

- ⋆ *objects* $r_{i,2n+2ln+k-i+1}$, $k+1 \leq i \leq n$, *being* $r \in \{t,f\}$.

$(a_1)$ *For each* $k$ $(1 \leq k \leq n)$ *and* $l$ $(0 \leq l \leq p-1)$ *at configuration* $\mathcal{C}_{3n+2ln+k}$ *we have the following:*

- $\mathcal{C}_{2n+2ln+k}(0) = \{\alpha_{2n+2ln+k}, \beta_{2n+2ln+k}\}$
- *There are* $2^n$ *membranes labelled by 1 such that each of them contains a different subset of objects* $r_{i,3n+ln+k-i+1}$, $k+1 \leq i \leq n$, *being* $r \in \{t,f\}$.
- *There are* $2^n$ *membranes labelled by 2 such that each of them contains*
  - ⋆ *the input multiset* $cod_{3n+2ln+k}(\varphi)$*;*
  - ⋆ *an object* $\gamma_{3n+2ln+k}$*;*
  - ⋆ $p$ *copies of every* $T_i$ *and* $F_i$, $1 \leq i \leq n$ *if the truth assignment associated to the branch contains its corresponding* $t_i$ *or* $f_i$ *object, and* $p-l$ *copies otherwise; and*
  - ⋆ *objects* $r_{i,3n+2ln+k-i+1}$, $1 \leq i \leq k$, *being* $r \in \{t,f\}$.

$(b)$ $\mathcal{C}_{2np}(0) = \{\alpha_{2np}, \beta_{2np}\}$*, and in* $\mathcal{C}_{2np}$ *there are* $2^n$ *empty membranes labelled by 1; and* $2^n$ *membranes labelled by 2 such that each of them contains the input multiset* $cod_{2np}(\varphi)$*, an object* $\gamma_{2np}$*,* $p$ *copies of every* $T_i$ *and* $F_i$*,* $1 \leq i \leq n$ *if the truth assignment associated to the branch contains its corresponding* $t_i$ *or* $f_i$ *object, and 1 object otherwise and objects* $r_{i,2np-i+1}$*,* $1 \leq i \leq n$*, being* $r \in \{t,f\}$*, that is, the truth assignment associated with the branch.*

*Proof.* $(a)$ is going to be demonstrated by induction on $l$

- The base case $l = 0$ is going to be demonstrated by induction on $k$

  $(a_0)$ The base case $k = 1$ is trivial because:

  - at configuration $\mathcal{C}_{2n}$ we have: $\mathcal{C}_{2n}(0) = \{\alpha_{2n}, \beta_{2n}\}$ and there exist $2^n$ empty membranes labelled by 1 containing ; and $2^n$ membranes labelled by 2 containing the input multiset $cod_{2n}(\varphi)$, an object $\gamma_{2n}$, $p$ copies of $T_i$ and $F_i$, being $1 \leq i \leq n$ and a different subset of objects $r_{i,2n-i+1}$, $1 \leq i \leq n$, being $r \in \{t,f\}$, the corresponding truth assignment of the branch. Then, configuration $\mathcal{C}_{2n}$ yields configuration $\mathcal{C}_{2n+1}$ by applying the rules:

$$
\begin{array}{l}
[\ t_{i,2n}\ F_i\ ]_2 \rightarrow t_{i,2n+1}[\quad]_2 \\
[\ f_{i,2n}\ T_i\ ]_2 \rightarrow f_{i,2n+1}[\quad]_2 \\
\left.\begin{array}{l}
[\ t_{i,2n+1-i} \rightarrow t_{i,2n+2-i}\ ]_2 \\
[\ f_{i,2n+1-i} \rightarrow f_{i,2n+2-1}\ ]_2
\end{array}\right\} \text{for } 2 \leq i \leq n \\
[\ \alpha_{2n} \rightarrow \alpha_{2n+1}\ ]_0 \\
[\ \beta_{2n} \rightarrow \beta_{2n+1}\ ]_0 \\
[\ \gamma_{2n} \rightarrow \gamma_{2n+1}\ ]_2 \\
\left.\begin{array}{l}
[\ x_{i,j,2n} \rightarrow x_{i,j,2n+1}\ ]_2 \\
[\ \overline{x}_{i,j,2n} \rightarrow \overline{x}_{i,j,2n+1}\ ]_2 \\
[\ x^*_{i,j,2n} \rightarrow x^*_{i,j,2n+1}\ ]_2
\end{array}\right\} \text{for } 1 \leq i \leq n, 1 \leq j \leq p
\end{array}
$$

  Thus, $\mathcal{C}_{2n+1}(0) = \{\alpha_{2n+1}, \beta_{2n+1}\}$, and there exist $2^n$ membranes labelled by 1 containing an object $r_{1,2n+1}$, being $r \in \{t,f\}$; and $2^n$ membranes labelled by 2 containing the input multiset $cod_{2n+1}(\varphi)$, an object $\gamma_{2n+1}$, $p$ copies of $T_i$ and $F_i$, being $2 \leq i \leq n$, and $p-1$ copies of $T_1$

(respectively, $F_i$) if object $f_{1,2n}$ (resp., $t_{1,2n}$) was within membrane labelled by 2 at configuration $\mathcal{C}_{2n}$, and $p$ copies of $F_1$ (resp., $T_1$), and a different subset of objects $r_{i,2n-i+2}$, $2 \le i \le n$, being $r \in \{t, f\}$.

- Supposing, by induction, result is true for $k$ $(1 \le k \le n)$
  - $\mathcal{C}_{2n+k}(0) = \{\alpha_{2n+k}, \beta_{2n+k}\}$
  - In $\mathcal{C}_{2n+k}$ there are $2^n$ membranes labelled by 1 such that each of them contains objects $r_{i,2n+k-i+1}$, $1 \le i \le k$, being $r \in \{t, f\}$.
  - In $\mathcal{C}_{2n+k}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ the input multiset $cod_{2n+k}(\varphi)$;
    - ⋆ an object $\gamma_{2n+k}$;
    - ⋆ $p$ copies of every $T_i$ and $F_i$, $1 \le i \le n$ if the truth assignment associated to the branch contains its corresponding $t_i$ or $f_i$ object, $p-1$ objects $T_i$ and $F_i$ $(1 \le i \le k)$ otherwise; and
    - ⋆ a different subset of objects $r_{i,2n+k-i+1}$, $k+1 \le i \le n$, being $r \in \{t, f\}$.

  Then, configuration $\mathcal{C}_{2n+k}$ yields configuration $\mathcal{C}_{2n+k+1}$ by applying the rules:

$$
\begin{array}{l}
[\ t_{k+1,2n}\ F_{k+1}\ ]_2 \to t_{k+1,2n+1}[\quad]_2 \\
[\ f_{k+1,2n}\ T_{k+1}\ ]_2 \to f_{k+1,2n+1}[\quad]_2 \\
\left.\begin{array}{l}
[\ t_{i,2n+k-i+1} \to t_{i,2n+k-i+2}\ ]_2 \\
[\ f_{i,2n+k-i+1} \to f_{i,2n+k-i+2}\ ]_2
\end{array}\right\} \text{for } k+2 \le i \le n \\
\left.\begin{array}{l}
[\ t_{i,2n+k-i+1} \to t_{i,2n+k-i+2}\ ]_1 \\
[\ f_{i,2n+k-i+1} \to f_{i,2n+k-i+2}\ ]_1
\end{array}\right\} \text{for } 1 \le i \le k \\
[\ \alpha_{2n+k} \to \alpha_{2n+k+1}\ ]_0 \\
[\ \beta_{2n+k} \to \beta_{2n+k+1}\ ]_0 \\
[\ \gamma_{2n+k} \to \gamma_{2n+k+1}\ ]_2 \\
\left.\begin{array}{l}
[\ x_{i,j,2n+k} \to x_{i,j,2n+k+1}\ ]_2 \\
[\ \overline{x}_{i,j,2n+k} \to \overline{x}_{i,j,2n+k+1}\ ]_2 \\
[\ x^*_{i,j,2n+k} \to x^*_{i,j,2n+k+1}\ ]_2
\end{array}\right\} \text{for } 1 \le i \le n, 1 \le j \le p
\end{array}
$$

  Therefore, the following holds
  - $\mathcal{C}_{2n+k+1}(0) = \{\alpha_{2n+k+1}, \beta_{2n+k+1}\}$
  - In $\mathcal{C}_{2n+k+1}$ there are $2^n$ membranes labelled by 1 such that each of them contains objects $r_{i,2n+k-i+2}$, $1 \le i \le k+1$, being $r \in \{t, f\}$.
  - In $\mathcal{C}_{2n+k+1}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ the input multiset $cod_{2n+k+1}(\varphi)$;
    - ⋆ an object $\gamma_{2n+k+1}$;
    - ⋆ $p$ copies of every $T_i$ and $F_i$, $1 \le i \le n$ if the truth assignment associated to the branch contains its corresponding $t_i$ or $f_i$ object, $p-1$ objects $T_i$ and $F_i$ $(1 \le i \le k+1)$ otherwise; and
    - ⋆ a different subset of objects $r_{i,2n+k-i+2}$, $k+2 \le i \le n$, being $r \in \{t, f\}$.

$(a_1)$ The base case $k = 1$ is trivial because:

- at configuration $\mathcal{C}_{3n}$ we have $\mathcal{C}_{3n}(0) = \{\alpha_{3n}, \beta_{3n}\}$ and there exist $2^n$ membranes labelled by 1 containing and a different subset of objects $r_{i,3n+1-i}$, $1 \leq i \leq n$, being $r \in \{t, f\}$, that is, the corresponding truth assignment of the branch; and $2^n$ membranes labelled by 2 containing the input multiset $cod_{3n}(\varphi)$, an object $\gamma_{3n}$, $p$ copies of $T_i$ and $F_i$, being $1 \leq i \leq n$ if the truth assignment associated to the branch contains its corresponding object $t_i$ or $f_i$, $p-1$ objects otherwise. Then, configuration $\mathcal{C}_{3n}$ yields configuration $\mathcal{C}_{3n+1}$ by applying the rules:

$$
\begin{array}{l}
t_{1,3n}\ [\quad]_2 \rightarrow [\ t_{1,3n+1}\ ]_2 \\
f_{1,3n}\ [\quad]_2 \rightarrow [\ f_{1,3n+1}\ ]_2 \\
\left.\begin{array}{l}
[\ t_{i,3n-i+1} \rightarrow t_{i,3n-i+2}\ ]_1 \\
[\ f_{i,3n-i+1} \rightarrow f_{i,3n-i+2}\ ]_1
\end{array}\right\} \text{for } 2 \leq i \leq n \\
[\ \alpha_{3n} \rightarrow \alpha_{3n+1}\ ]_0 \\
[\ \beta_{3n} \rightarrow \beta_{3n+1}\ ]_0 \\
[\ \gamma_{3n} \rightarrow \gamma_{3n+1}\ ]_2 \\
\left.\begin{array}{l}
[\ x_{i,j,3n} \rightarrow x_{i,j,3n+1}\ ]_2 \\
[\ \overline{x}_{i,j,3n} \rightarrow \overline{x}_{i,j,3n+1}\ ]_2 \\
[\ x^*_{i,j,3n} \rightarrow x^*_{i,j,3n+1}\ ]_2
\end{array}\right\} \text{for } 1 \leq i \leq n, 1 \leq j \leq p
\end{array}
$$

  Thus, $\mathcal{C}_{3n+1}(0) = \{\alpha_{3n+1}, \beta_{3n+1}\}$, and there exist $2^n$ membranes labelled by 1 containing a different subset of objects $r_{i,3n-i+2}$, $2 \leq i \leq n$, being $r \in \{t, f\}$; and $2^n$ membranes labelled by 2 containing the input multiset $cod_{3n+1}(\varphi)$, an object $\gamma_{3n+1}$, $p$ copies of $T_i$ and $F_i$, being $1 \leq i \leq n$ if the truth assignment associated to the branch contains its corresponding object $t_i$ or $f_i$, $p-1$ objects otherwise and an object $r_{1,3n+1}$, being $r \in \{t, f\}$.

- Supposing, by induction, result is true for $k$ $(1 \leq k \leq n)$
  - $\mathcal{C}_{3n+k}(0) = \{\alpha_{3n+k}, \beta_{3n+k}\}$
  - In $\mathcal{C}_{3n+k}$ there are $2^n$ membranes labelled by 1 such that each of them contains objects $r_{i,3n+k-i+1}$, $k+1 \leq i \leq n$, being $r \in \{t, f\}$.
  - In $\mathcal{C}_{3n+k}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ the input multiset $cod_{3n+k}(\varphi)$;
    - ⋆ an object $\gamma_{3n+k}$;
    - ⋆ $p$ copies of every $T_i$ and $F_i$ for $1 \leq i \leq n$ or their corresponding $t_i$ or $f_i$ is assigned to that branch, $p-l$ copies otherwise; and
    - ⋆ a different subset of objects $r_{i,3n+k-i+1}$, $1 \leq i \leq k$, being $r \in \{t, f\}$.

  Then, configuration $\mathcal{C}_{3n+k}$ yields configuration $\mathcal{C}_{3n+k+1}$ by applying the rules:

$$
\begin{array}{l}
t_{k+1,3n}\ [\quad]_2 \rightarrow [\ t_{k+1,3n+1}\ ]_2 \\
f_{k+1,3n}\ [\quad]_2 \rightarrow [\ f_{k+1,3n+1}\ ]_2 \\
\left.\begin{array}{l}
[\ t_{i,3n+k-i+1} \rightarrow t_{i,3n+k-i+2}\ ]_1 \\
[\ f_{i,3n+k-i+1} \rightarrow f_{i,3n+k-i+2}\ ]_1
\end{array}\right\} \text{for } k+2 \leq i \leq n \\
\left.\begin{array}{l}
[\ t_{i,3n+k-i+1} \rightarrow t_{i,3n+k-i+2}\ ]_2 \\
[\ f_{i,3n+k-i+1} \rightarrow f_{i,3n+k-i+2}\ ]_2
\end{array}\right\} \text{for } 1 \leq i \leq k
\end{array}
$$

$[\ \alpha_{3n+k} \to \alpha_{3n+k+1}\ ]_0$
$[\ \beta_{3n+k} \to \beta_{3n+k+1}\ ]_0$
$[\ \gamma_{3n+k} \to \gamma_{3n+k+1}\ ]_2$
$$\left.\begin{array}{l}[\ x_{i,j,3n+k} \to x_{i,j,3n+k+1}\ ]_2 \\ [\ \overline{x}_{i,j,3n+k} \to \overline{x}_{i,j,3n+k+1}\ ]_2 \\ [\ x^*_{i,j,3n+k} \to x^*_{i,j,3n+k+1}\ ]_2\end{array}\right\} \text{for } 1 \le i \le n, 1 \le j \le p$$

Therefore, the following holds

- $\mathcal{C}_{3n+k+1}(0) = \{\alpha_{3n+k+1}, \beta_{3n+k+1}\}$
- In $\mathcal{C}_{3n+k+1}$ there are $2^n$ membranes labelled by 1 such that each of them contains objects $r_{i,3n+k-i+2}$, $k+2 \le i \le n$, being $r \in \{t, f\}$.
- In $\mathcal{C}_{3n+k+1}$ there are $2^n$ membranes labelled by 2 such that each of them contains
  - ⋆ the input multiset $cod_{3n+k+1}(\varphi)$;
  - ⋆ an object $\gamma_{3n+k+1}$;
  - ⋆ $p$ copies of every $T_i$ and $F_i$ for $1 \le i \le n$ or the corresponding $t_i$ or $f_i$ is assigned to that branch, $p-l$ copies otherwise; and
  - ⋆ a different subset of objects $r_{i,3n+k-i+2}$, $1 \le i \le k+1$, being $r \in \{t, f\}$.

- Supposing, by induction, result is true for $l$ ($0 \le l \le p-1$)

  $(a_0)$ The base case $k = 1$ is trivial because:
  - at configuration $\mathcal{C}_{2n+(l+1)n}$[1] we have: $\mathcal{C}_{2n+(l+1)n}(0) = \{\alpha_{2n+(l+1)n}, \beta_{2n+(l+1)n}\}$ and there exist $2^n$ empty membranes labelled by 1; and $2^n$ membranes labelled by 2 containing the input multiset $cod_{2n+(l+1)n}(\varphi)$, an object $\gamma_{2n+(l+1)n}$, $p$ copies of $T_i$ and $F_i$, being $1 \le i \le n$, and $p-l$ copies for $T_i$ (resp. $F_i$) objects that are in a branch with an object $f_i$ (resp. $t_i$) and a different subset of objects $r_{i,2n+(l+1)n-i+1}$, $1 \le i \le n$, being $r \in \{t, f\}$, the corresponding truth assignment of the branch. Then, configuration $\mathcal{C}_{2n+(l+1)n}$ yields configuration $\mathcal{C}_{2n+(l+1)n+1}$ by applying the rules:

$[\ t_{i,2n+(l+1)n}\ F_i\ ]_2 \to t_{i,2n+(l+1)n+1}[\quad]_2$
$[\ f_{i,2n+(l+1)n}\ T_i\ ]_2 \to f_{i,2n+(l+1)n+1}[\quad]_2$
$$\left.\begin{array}{l}[\ t_{i,2n+(l+1)n+1-i} \to t_{i,2n+(l+1)n+2-i}\ ]_2 \\ [\ f_{i,2n+(l+1)n+1-i} \to f_{i,2n+(l+1)n+2-i}\ ]_2\end{array}\right\} \text{for } 2 \le i \le n$$
$[\ \alpha_{2n+(l+1)n} \to \alpha_{2n+(l+1)n+1}\ ]_0$
$[\ \beta_{2n+(l+1)n} \to \beta_{2n+(l+1)n+1}\ ]_0$
$[\ \gamma_{2n+(l+1)n} \to \gamma_{2n+(l+1)n+1}\ ]_2$
$$\left.\begin{array}{l}[\ x_{i,j,2n+(l+1)n} \to x_{i,j,2n+(l+1)n+1}\ ]_2 \\ [\ \overline{x}_{i,j,2n+(l+1)n} \to \overline{x}_{i,j,2n+(l+1)n+1}\ ]_2 \\ [\ x^*_{i,j,2n+(l+1)n} \to x^*_{i,j,2n+(l+1)n+1}\ ]_2\end{array}\right\} \text{for } 1 \le i \le n, 1 \le j \le p$$

Thus, $\mathcal{C}_{2n+(l+1)n+1}(0) = \{\alpha_{2n+(l+1)n+1}, \beta_{2n+(l+1)n+1}\}$, and there exist $2^n$ membranes labelled by 1 containing and an object $r_{1,2n+(l+1)n+1}$,

[1] Note that $(l+1)n = ln + n$, and it has been demonstrated in the first step of the induction that it is correct.

being $r \in \{t, f\}$; and $2^n$ membranes labelled by 2 containing the input multiset $cod_{2n+(l+1)n+1}(\varphi)$, an object $\gamma_{2n+(l+1)n+1}$, $p$ copies of $T_i$ (resp. $F_i$) being $1 \le i \le n$ if the corresponding $t_i$ (resp. $f_i$) object exists in that branch, otherwise $p-l$ copies of $F_i$ (resp. $T_i$) if $2 \le i \le n$, $p-l-1$ otherwise and a different subset of objects $r_{i,2n+(l+1)n-i+2}$, $2 \le i \le n$, being $r \in \{t, f\}$.

- Supposing, by induction, result is true for $k$ $(1 \le k \le n)$
  - $\mathcal{C}_{2n+(l+1)n+k}(0) = \{\alpha_{2n+(l+1)n+k}, \beta_{2n+(l+1)n+k}\}$
  - In $\mathcal{C}_{2n+(l+1)n+k}$ there are $2^n$ membranes labelled by 1 such that each of them contains objects $r_{i,2n+(l+1)n+k-i+1}$, $1 \le i \le k$, being $r \in \{t, f\}$.
  - In $\mathcal{C}_{2n+(l+1)n+k}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ the input multiset $cod_{2n+(l+1)n+k}(\varphi)$;
    - ⋆ an object $\gamma_{2n+(l+1)n+k}$;
    - ⋆ $p$ copies of $T_i$ (resp. $F_i$) being $1 \le i \le n$ if the corresponding $t_i$ (resp. $f_i$) object exists in that branch, otherwise $p-l$ copies of $F_i$ (resp. $T_i$) if $k+1 \le i \le n$, $p-l-1$ otherwise; and
    - ⋆ a different subset of objects $r_{i,2n+(l+1)n+k-i+1}$, $k+1 \le i \le n$, being $r \in \{t, f\}$.

  Then, configuration $\mathcal{C}_{2n+k}$ yields configuration $\mathcal{C}_{2n+(l+1)n+k+1}$ by applying the rules:

$$
\begin{array}{l}
[\ t_{k+1,2n+(l+1)n}\ F_{k+1}\ ]_2 \rightarrow t_{k+1,2n+(l+1)n+1}[\quad]_2 \\
[\ f_{k+1,2n+(l+1)n}\ T_{k+1}\ ]_2 \rightarrow f_{k+1,2n+(l+1)n+1}[\quad]_2 \\
\left.\begin{array}{l}
[\ t_{i,2n+(l+1)n+k-i+1} \rightarrow t_{i,2n+k-i+2}\ ]_2 \\
[\ f_{i,2n+(l+1)n+k-i+1} \rightarrow f_{i,2n+k-i+2}\ ]_2
\end{array}\right\} \text{for } k+2 \le i \le n \\
\left.\begin{array}{l}
[\ t_{i,2n+(l+1)n+k-i+1} \rightarrow t_{i,2n+(l+1)n+k-i+2}\ ]_1 \\
[\ f_{i,2n+(l+1)n+k-i+1} \rightarrow f_{i,2n+(l+1)n+k-i+2}\ ]_1
\end{array}\right\} \text{for } 1 \le i \le k \\
[\ \alpha_{2n+(l+1)n+k} \rightarrow \alpha_{2n+(l+1)n+k+1}\ ]_0 \\
[\ \beta_{2n+(l+1)n+k} \rightarrow \beta_{2n+(l+1)n+k+1}\ ]_0 \\
[\ \gamma_{2n+(l+1)n+k} \rightarrow \gamma_{2n+(l+1)n+k+1}\ ]_2 \\
\left.\begin{array}{l}
[\ x_{i,j,2n+(l+1)n+k} \rightarrow x_{i,j,2n+(l+1)n+k+1}\ ]_2 \\
[\ \overline{x}_{i,j,2n+(l+1)n+k} \rightarrow \overline{x}_{i,j,2n+(l+1)n+k+1}\ ]_2 \\
[\ x^*_{i,j,2n+(l+1)n+k} \rightarrow x^*_{i,j,2n+(l+1)n+k+1}\ ]_2
\end{array}\right\} \text{for } 1 \le i \le n, 1 \le j \le p
\end{array}
$$

  Therefore, the following holds
  - $\mathcal{C}_{2n+(l+1)n+k+1}(0) = \{\alpha_{2n+(l+1)n+k+1}, \beta_{2n+(l+1)n+k+1}\}$
  - In $\mathcal{C}_{2n+(l+1)n+k+1}$ there are $2^n$ membranes labelled by 1 such that each of them contains objects $r_{i,2n+(l+1)n+k-i+2}$, $1 \le i \le k+1$, being $r \in \{t, f\}$.
  - In $\mathcal{C}_{2n+(l+1)n+k+1}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ the input multiset $cod_{2n+(l+1)n+k+1}(\varphi)$;
    - ⋆ an object $\gamma_{2n+(l+1)n+k+1}$;

- ⋆ $p$ copies of $T_i$ (resp. $F_i$) being $1 \leq i \leq n$ if the corresponding $t_i$ (resp. $f_i$) object exists in that branch, otherwise $p-l$ copies of $F_i$ (resp. $T_i$) if $k+2 \leq i \leq n$, $p-l-1$ otherwise; and
- ⋆ a different subset of objects $r_{i,2n+(l+1)n+k-i+2}$, $k+2 \leq i \leq n$, being $r \in \{t,f\}$.

$(a_1)$ The base case $k=1$ is trivial because:

- at configuration $\mathcal{C}_{3n+(l+1)n}$ we have $\mathcal{C}_{3n+(l+1)n}(0) = \{\alpha_{3n+(l+1)n}, \beta_{3n+(l+1)n}\}$ and there exist $2^n$ membranes labelled by 1 containing a different subset of objects $r_{i,3n+(l+1)n-i+1}$, $1 \leq i \leq n$, being $r \in \{t,f\}$, that is, the corresponding truth assignment of the branch; and $2^n$ membranes labelled by 2 containing the input multiset $cod_{3n+(l+1)n}(\varphi)$, an object $\gamma_{3n+(l+1)n}$ and $p$ copies of $T_i$ (resp. $F_i$) being $1 \leq i \leq n$ if the corresponding $t_i$ (resp. $f_i$) object exists in that branch, and $p-l$ copies of $F_i$ (resp. $T_i$). Then, configuration $\mathcal{C}_{3n+(l+1)n}$ yields configuration $\mathcal{C}_{3n+(l+1)n+1}$ by applying the rules:

$$
\begin{array}{l}
t_{1,3n+(l+1)n}\ [\ \ ]_2 \rightarrow [\ t_{1,3n+(l+1)n+1}\ ]_2 \\
f_{1,3n+(l+1)n}\ [\ \ ]_2 \rightarrow [\ f_{1,3n+(l+1)n+1}\ ]_2 \\
\left.\begin{array}{l}
[\ t_{i,3n+(l+1)n-i+1} \rightarrow t_{i,3n+(l+1)n-i+2}\ ]_1 \\
[\ f_{i,3n+(l+1)n-i+1} \rightarrow f_{i,3n+(l+1)n-i+2}\ ]_1
\end{array}\right\} \text{for } 2 \leq i \leq n \\
[\ \alpha_{3n+(l+1)n} \rightarrow \alpha_{3n+(l+1)n+1}\ ]_0 \\
[\ \beta_{3n+(l+1)n} \rightarrow \beta_{3n+(l+1)n+1}\ ]_0 \\
[\ \gamma_{3n+(l+1)n} \rightarrow \gamma_{3n+(l+1)n+1}\ ]_2 \\
\left.\begin{array}{l}
[\ x_{i,j,3n+(l+1)n} \rightarrow x_{i,j,3n+(l+1)n+1}\ ]_2 \\
[\ \overline{x}_{i,j,3n+(l+1)n} \rightarrow \overline{x}_{i,j,3n+(l+1)n+1}\ ]_2 \\
[\ x^*_{i,j,3n+(l+1)n} \rightarrow x^*_{i,j,3n+(l+1)n+1}\ ]_2
\end{array}\right\} \text{for } 1 \leq i \leq n, 1 \leq j \leq p
\end{array}
$$

Thus, $\mathcal{C}_{3n+(l+1)n+1}(0) = \{\alpha_{3n+(l+1)n+1}, \beta_{3n+(l+1)n+1}\}$, and there exist $2^n$ membranes labelled by 1 containing a different subset of objects $r_{i,3n+(l+1)n-i+2}$, $2 \leq i \leq n$, being $r \in \{t,f\}$; and $2^n$ membranes labelled by 2 containing the input multiset $cod_{3n+(l+1)n+1}(\varphi)$, an object $\gamma_{3n+(l+1)n+1}$, $p$ copies of $T_i$ (resp. $F_i$) being $1 \leq i \leq n$ if the corresponding $t_i$ (resp. $f_i$) object exists in that branch, and $p-l$ copies of $F_i$ (resp. $T_i$) and an object $r_{1,3n+(l+1)n+1}$, being $r \in \{t,f\}$.

- Supposing, by induction, result is true for $k$ $(1 \leq k \leq n)$
  - $\mathcal{C}_{3n+(l+1)n+k}(0) = \{\alpha_{3n+(l+1)n+k}, \beta_{3n+(l+1)n+k}\}$
  - In $\mathcal{C}_{3n+(l+1)n+k}$ there are $2^n$ membranes labelled by 1 such that each of them contains objects $r_{i,3n+k-i+1}$, $k+1 \leq i \leq n$, being $r \in \{t,f\}$.
  - In $\mathcal{C}_{3n+(l+1)n+k}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ the input multiset $cod_{3n+(l+1)n+k}(\varphi)$;
    - ⋆ an object $\gamma_{3n+(l+1)n+k}$;
    - ⋆ $p$ copies of $T_i$ (resp. $F_i$) being $1 \leq i \leq n$ if the corresponding $t_i$ (resp. $f_i$) object exists in that branch, and $p-l$ copies of $F_i$ (resp. $T_i$)

- ⋆ a different subset of objects $r_{i,3n+(l+1)n-i+1}$, $1 \le i \le k$, being $r \in \{t, f\}$.

Then, configuration $\mathcal{C}_{3n+(l+1)n+k}$ yields configuration $\mathcal{C}_{3n+(l+1)n+k+1}$ by applying the rules:

$$
\begin{array}{l}
t_{k+1,3n+(l+1)n}\ [\ \ ]_2 \rightarrow [\ t_{k+1,3n+(l+1)n+1}\ ]_2 \\
f_{k+1,3n+(l+1)n}\ [\ \ ]_2 \rightarrow [\ f_{k+1,3n+(l+1)n+1}\ ]_2 \\
\left.\begin{array}{l}
[\ t_{i,3n+(l+1)n+k-i+1} \rightarrow t_{i,3n+(l+1)n+k-i+2}\ ]_1 \\
[\ f_{i,3n+(l+1)n+k-i+1} \rightarrow f_{i,3n+(l+1)n+k-i+2}\ ]_1
\end{array}\right\} \text{for } k+2 \le i \le n \\
\left.\begin{array}{l}
[\ t_{i,3n+(l+1)n+k-i+1} \rightarrow t_{i,3n+(l+1)n+k-i+2}\ ]_2 \\
[\ f_{i,3n+(l+1)n+k-i+1} \rightarrow f_{i,3n+(l+1)n+k-i+2}\ ]_2
\end{array}\right\} \text{for } 1 \le i \le k \\
[\ \alpha_{3n+(l+1)n+k} \rightarrow \alpha_{3n+(l+1)n+k+1}\ ]_0 \\
[\ \beta_{3n+(l+1)n+k} \rightarrow \beta_{3n+(l+1)n+k+1}\ ]_0 \\
[\ \gamma_{3n+(l+1)n+k} \rightarrow \gamma_{3n+(l+1)n+k+1}\ ]_2 \\
\left.\begin{array}{l}
[\ x_{i,j,3n+(l+1)n+k} \rightarrow x_{i,j,3n+(l+1)n+k+1}\ ]_2 \\
[\ \overline{x}_{i,j,3n+(l+1)n+k} \rightarrow \overline{x}_{i,j,3n+(l+1)n+k+1}\ ]_2 \\
[\ x^*_{i,j,3n+(l+1)n+k} \rightarrow x^*_{i,j,3n+(l+1)n+k+1}\ ]_2
\end{array}\right\} \text{for } 1 \le i \le n, 1 \le j \le p
\end{array}
$$

Therefore, the following holds

- $\mathcal{C}_{3n+(l+1)n+k+1}(0) = \{\alpha_{3n+(l+1)n+k+1}, \beta_{3n+(l+1)n+k+1}\}$
- In $\mathcal{C}_{3n+(l+1)n+k+1}$ there are $2^n$ membranes labelled by 1 such that each of them contains objects $r_{i,3n+(l+1)n+k-i+2}$, $k+2 \le i \le n$, being $r \in \{t, f\}$.
- In $\mathcal{C}_{3n+(l+1)n+k+1}$ there are $2^n$ membranes labelled by 2 such that each of them contains
  - ⋆ the input multiset $cod_{3n+(l+1)n+k+1}(\varphi)$;
  - ⋆ an object $\gamma_{3n+(l+1)n+k+1}$;
  - ⋆ $p$ copies of $T_i$ (resp. $F_i$) being $1 \le i \le n$ if the corresponding $t_i$ (resp. $f_i$) object exists in that branch, and $p - l$ copies of $F_i$ (resp. $T_i$)
  - ⋆ a different subset of objects $r_{i,3n+(l+1)n+k-i+2}$, $1 \le i \le k+1$, being $r \in \{t, f\}$.

- In order to prove $(b)$ it is enough to notice that, on the one hand, from $(a)$ configuration $\mathcal{C}_{2np-1}$[2] holds:
  - $\mathcal{C}_{2np-1}(0) = \{\alpha_{2np-1}, \beta_{2np-1}\}$
  - In $\mathcal{C}_{2np-1}$ there are $2^n$ membranes labelled by 1 such that each of them contains an object $r_{n,2np}$, being $r \in \{t, f\}$.
  - In $\mathcal{C}_{n+2np-1}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ the input multiset $cod_{2np-1}(\varphi)$;
    - ⋆ an object $\gamma_{2np-1}$;
    - ⋆ $p$ copies of $T_i$ (resp. $F_i$) being $1 \le i \le n$ if the corresponding $t_i$ (resp. $f_i$) object exists in that branch, and 1 copy otherwise; and
    - ⋆ a different subset of objects $r_{i,2np-i}$, $1 \le i \le n-1$.

[2] Note that $2np - 1 = n + 2n(p-1) + (n-1)$

Then, configuration $\mathcal{C}_{n+2np-1}$ yields $\mathcal{C}_{n+2np}$ by applying the rules:

$$t_{n,2np}[\quad]_2 \rightarrow [\ t_{n,2np+1}\ ]_2$$
$$f_{n,2np}[\quad]_2 \rightarrow [\ f_{n,2np+1}\ ]_2$$
$$\left.\begin{array}{l}[\ t_{i,n+2np-i} \rightarrow t_{i,n+2np-i+1}\ ]_2 \\ [\ f_{i,n+2np-i} \rightarrow f_{i,n+2np-i}\ ]_2\end{array}\right\} \text{for } 1 \leq i \leq n-1$$
$$[\ \alpha_{n+2np-1} \rightarrow \alpha_{n+2np}\ ]_0$$
$$[\ \beta_{n+2np-1} \rightarrow \beta_{n+2np}\ ]_0$$
$$[\ \gamma_{n+2np-1} \rightarrow \gamma_{n+2np}\ ]_2$$
$$\left.\begin{array}{l}[\ x_{i,j,n+2np-1} \rightarrow x_{i,j,n+2np}\ ]_2 \\ [\ \overline{x}_{i,j,n+2np-1} \rightarrow \overline{x}_{i,j,n+2np}\ ]_2 \\ [\ x^*_{i,j,n+2np-1} \rightarrow x^*_{i,j,n+2np}\ ]_2\end{array}\right\} \text{for } 1 \leq i \leq n, 1 \leq j \leq p$$

Then, we have $\mathcal{C}_{2np}(0) = \{\alpha_{2np}, \beta_{2np}\}$, and there exist $2^n$ empty membranes labelled by 1; and $2^n$ membranes labelled by 2 containing containing the input multiset $cod_{2np}(\varphi)$, an object $\gamma_{2np}$, $p$ copies of $T_i$ (resp. $F_i$) being $1 \leq i \leq n$ if the corresponding $t_i$ (resp. $f_i$) object exists in that branch, and 1 copy otherwise and a different multiset of objects $r_{i,2np-i+1}$, $1 \leq i \leq n$, being $r \in \{t, f\}$, that is, the truth assignment associated with the branch.

□

**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* $k$ *(*$1 \leq k \leq n-1$*) at configuration* $\mathcal{C}_{2np+k}$ *we have the following:*
- $\mathcal{C}_{2np+k}(0) = \{\alpha_{2np+k}, \beta_{2np+k}\}$
- *There are* $2^n$ *membranes labelled by 1 such that each of them contains* $k$ *objects* $\#$*.*
- *there are* $2^n$ *membranes labelled by 2 such that each of them contains*
  - ⋆ *the input multiset* $cod_{2np+k}(\varphi)$*;*
  - ⋆ *an object* $\gamma_{2np+k}$*;*
  - ⋆ $p$ *copies of* $T_i$ *(resp.* $F_i$*) being* $1 \leq i \leq n$ *if the corresponding* $t_i$ *(resp.* $f_i$*) object exists in that branch, and 1 copy of* $F_i$ *(resp.* $T_i$*) if* $k+1 \leq i \leq n$*; and*
  - ⋆ *objects* $r_{i,2np+k-i+1}$*,* $k+1 \leq i \leq n$*.*

$(b)$ $\mathcal{C}_{n+2np}(0) = \{\alpha_{n+2np}, \beta_{n+2np}\}$*, and in* $\mathcal{C}_{n+2np}$ *there are* $2^n$ *membranes labelled by 1, such that each of them contains* $n$ *objects* $\#$*; and* $2^n$ *membranes labelled by 2, such that each of them contains the input multiset* $cod_{n+2np}(\varphi)$*, an object* $\gamma_{n+2np}$*,* $p$ *copies of every* $T_i$ *and* $F_i$*,* $1 \leq i \leq n$ *if the truth assignment associated to the branch contains its corresponding* $t_i$ *or* $f_i$ *object.*

*Proof.* $(a)$ is going to be demonstrated by induction on $k$

- the base case $k = 1$ is trivial because:
  - at $\mathcal{C}_{2np}$ we have $\mathcal{C}_{2np}(0) = \{\alpha_{2np}, \beta_{2np}\}$ and there exist $2^n$ empty membranes labelled by 1; and $2^n$ membranes labelled by 2 containing the input multiset $cod_{2np}(\varphi)$, an object $\gamma_{2np}$ $p$ copies of $T_i$ (resp. $F_i$) being $1 \leq i \leq n$ if

the corresponding $t_i$ (resp. $f_i$) object exists in that branch, and 1 copy otherwise and a different multiset of objects $r_{i,2np-i+1}$, $1 \leq i \leq n$, being $r \in \{t, f\}$, that is, the truth assignment associated with the branch. Then, configuration $\mathcal{C}_{2np}$ yields $\mathcal{C}_{2np+1}$ by applying the rules.

$$\begin{array}{l}
[\ t_{1,2np}\ F_1\ ]_2 \rightarrow \#[\quad]_2 \\
[\ f_{1,2np}\ T_1\ ]_2 \rightarrow \#[\quad]_2 \\
\left.\begin{array}{l} [\ t_{i,2np-i+1} \rightarrow t_{i,2np-i+2}\ ]_2 \\ [\ f_{i,2np-i+1} \rightarrow f_{i,2np-i+2}\ ]_2 \end{array}\right\} \text{for } 2 \leq i \leq n \\
[\ \alpha_{2np} \rightarrow \alpha_{2np+1}\ ]_0 \\
[\ \beta_{2np} \rightarrow \beta_{2np+1}\ ]_0 \\
[\ \gamma_{2np} \rightarrow \gamma_{2np+1}\ ]_2 \\
\left.\begin{array}{l} [\ x_{i,j,2np} \rightarrow x_{i,j,2np+1}\ ]_2 \\ [\ \overline{x}_{i,j,2np} \rightarrow \overline{x}_{i,j,2np+1}\ ]_2 \\ [\ x^*_{i,j,2np} \rightarrow x^*_{i,j,2np+1}\ ]_2 \end{array}\right\} \text{for } 1 \leq i \leq n, 1 \leq j \leq p
\end{array}$$

Thus, $\mathcal{C}_{2np+1}(0) = \{\alpha_{2np+1}, \beta_{2np+1}\}$, and there exist $2^n$ membranes labelled by 1 containing an object #; and $2^n$ membranes labelled by 2 containing the input multiset $cod_{2np+1}(\varphi)$, an object $\gamma_{2np+1}$, $p$ copies of $T_i$ (resp. $F_i$) being $1 \leq i \leq n$ if their corresponding $t_i$ (resp. $f_i$) object exists in that branch, and 1 copy of $F_i$ (resp. $T_i$) if $k+2 \leq i \leq n$ and objects $r_{i,2np-i+2}$, $k+2 \leq i \leq n$, being $r \in \{t, f\}$.

- Supposing, by induction, result is true for $k$ ($1 \leq k \leq n-1$)
  - $\mathcal{C}_{2np+k}(0) = \{\alpha_{2np+k}, \beta_{2np+k}\}$
  - In $\mathcal{C}_{2np+k}$ there are $2^n$ membranes labelled by 1 such that each of them contains $k$ objects #.
  - In $\mathcal{C}_{2np+k}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ the input multiset $cod_{2np+k}(\varphi)$;
    - ⋆ an object $\gamma_{2np+k}$;
    - ⋆ $p$ copies of $T_i$ (resp. $F_i$) being $1 \leq i \leq n$ if their corresponding $t_i$ (resp. $f_i$) object exists in that branch, and 1 copy of $F_i$ (resp. $T_i$) if $k+1 \leq i \leq n$; and
    - ⋆ objects $r_{i,2np+k-i+1}$, $k+1 \leq i \leq n$, being $r \in \{t, f\}$.

  Then, configuration $\mathcal{C}_{2np+k}$ yields configuration $\mathcal{C}_{2np+k+1}$ by applying the rules:

$$\begin{array}{l}
[\ t_{k+1,2np}\ F_1\ ]_2 \rightarrow \#[\quad]_2 \\
[\ f_{k+1,2np}\ T_1\ ]_2 \rightarrow \#[\quad]_2 \\
\left.\begin{array}{l} [\ t_{i,2np+k-i+1} \rightarrow t_{i,2np+k-i+2}\ ]_2 \\ [\ f_{i,2np+k-i+1} \rightarrow f_{i,2np+k-i+2}\ ]_2 \end{array}\right\} \text{for } 2 \leq i \leq n \\
[\ \alpha_{2np+k} \rightarrow \alpha_{2np+k+1}\ ]_0 \\
[\ \beta_{2np+k} \rightarrow \beta_{2np+k+1}\ ]_0 \\
[\ \gamma_{2np+k} \rightarrow \gamma_{2np+k+1}\ ]_2 \\
\left.\begin{array}{l} [\ x_{i,j,2np+k} \rightarrow x_{i,j,2np+k+1}\ ]_2 \\ [\ \overline{x}_{i,j,2np+k} \rightarrow \overline{x}_{i,j,2np+k+1}\ ]_2 \\ [\ x^*_{i,j,2np+k} \rightarrow x^*_{i,j,2np+k+1}\ ]_2 \end{array}\right\} \text{for } 1 \leq i \leq n, 1 \leq j \leq p
\end{array}$$

Therefore, the following holds

- $\mathcal{C}_{2np+k+1}(0) = \{\alpha_{2np+k+1}, \beta_{2np+k+1}\}$
- In $\mathcal{C}_{2np+k+1}$ there are $2^n$ membranes labelled by 1 such that each of them contains $k+1$ objects $\#$.
- In $\mathcal{C}_{2np+k+1}$ there are $2^n$ membranes labelled by 2 such that each of them contains
  - ⋆ the input multiset $cod_{2np+k+1}(\varphi)$;
  - ⋆ an object $\gamma_{2np+k+1}$;
  - ⋆ $p$ copies of $T_i$ (resp. $F_i$) being $1 \leq i \leq n$ if their corresponding $t_i$ (resp. $f_i$) object exists in that branch, and 1 copy of $F_i$ (resp. $T_i$) if $k+2 \leq i \leq n$; and
  - ⋆ objects $r_{i,2np+k-i+2}$, $k+2 \leq i \leq n$, being $r \in \{t, f\}$.

- In order to prove $(b)$ it is enough to notice that, on the one hand, from $(a)$ configuration $\mathcal{C}_{n+2np-1}$[3] holds:
  - $\mathcal{C}_{n+2np-1}(0) = \{\alpha_{n+2np-1}, \beta_{n+2np-1}\}$
  - In $\mathcal{C}_{n+2np-1}$ there are $2^n$ membranes labelled by 1 such that each of them contains $n-1$ objects $\#$.
  - In $\mathcal{C}_{n+2np-1}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ the input multiset $cod_{n+2np-1}(\varphi)$;
    - ⋆ an object $\gamma_{n+2np-1}$;
    - ⋆ $p$ copies of $T_i$ (resp. $F_i$) being $1 \leq i \leq n$ if the corresponding $t_i$ (resp. $f_i$) object exists in that branch, and 1 copy of $F_n$ (resp. $T_n$); and
    - ⋆ an object $r_{n,2np}$, being $r \in \{t, f\}$.

  Then, configuration $\mathcal{C}_{n+2np-1}$ yields configuration $\mathcal{C}_{n+2np}$ by applying the rules:

$$
\begin{array}{l}
[\ t_{n,2np}\ F_1\ ]_2 \to \#[\quad]_2 \\
[\ f_{n,2np}\ T_1\ ]_2 \to \#[\quad]_2 \\
[\ \alpha_{n+2np-1} \to \alpha_{n+2np}\ ]_0 \\
[\ \beta_{n+2np-1} \to \beta_{n+2np}\ ]_0 \\
[\ \gamma_{n+2np-1} \to \gamma_{n+2np}\ ]_2 \\
\left.\begin{array}{l}
[\ x_{i,j,n+2np-1} \to x_{i,j,n+2np}\ ]_2 \\
[\ \overline{x}_{i,j,n+2np-1} \to \overline{x}_{i,j,n+2np}\ ]_2 \\
[\ x^*_{i,j,n+2np-1} \to x^*_{i,j,n+2np}\ ]_2
\end{array}\right\} \text{for } 1 \leq i \leq n, 1 \leq j \leq p
\end{array}
$$

  Therefore, the following holds
  - $\mathcal{C}_{n+2np}(0) = \{\alpha_{n+2np}, \beta_{n+2np}\}$
  - In $\mathcal{C}_{n+2np}$ there are $2^n$ membranes labelled by 1 such that each of them contains $n$ objects $\#$.
  - In $\mathcal{C}_{n+2np}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ the input multiset $cod_{n+2np}(\varphi)$;
    - ⋆ an object $\gamma_{n+2np}$; and

[3] Note that $n + 2np - 1 = 2np + (n-1)$

- ⋆ $p$ copies of $T_i$ (resp. $F_i$) being $1 \leq i \leq n$ if their corresponding $t_i$ (resp. $f_i$) object exists in that branch.

□

### 5.2 First checking stage

At this stage, we try to determine the clauses satisfied for the truth assignment encoded by each branch. For that, rules from **5.5** will be applied in such manner that in the $m$-th step, being $m = ln+k$ ($1 \leq k \leq n$, $0 \leq l \leq p-1$), clause $C_{l+1}$ will be evaluated with the $k$-th variable of the formula. This stage will take exactly $np$ steps.

**Proposition 4.** *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_q)$ be a computation of the system $\Pi(s(\varphi))$ with input multiset $cod(\varphi)$.*

(*a*) *For each $k$ ($1 \leq k \leq n$) and $l$ ($0 \leq l \leq p-1$) at configuration $\mathcal{C}_{n+2np+ln+k}$ we have the following:*
  - $\mathcal{C}_{n+2np+ln+k}(0) = \{\alpha_{n+2np+ln+k}, \beta_{n+2np+ln+k}\}$
  - *There are $2^n$ membranes labelled by 1 such that each of them contains*
    - ⋆ *$m$ objects $c_{j,t}$ ($1 \leq j \leq l+1$, $0 \leq t \leq ln+k-1$), that is, clauses that have been satisfied by any variable; and*
    - ⋆ *$n+ln+k-m$ objects #.*
  - *There are $2^n$ membranes labelled by 2 such that each of them contains*
    - ⋆ *the $(n-k)$-th last elements of $cod_{n+2np+ln+k}(\varphi)_{l+1}^{l+1}$;*
    - ⋆ *the input multiset $cod_{n+2np+ln+k}(\varphi)_{l+2}^{p}$;*
    - ⋆ *an object $\gamma_{n+2np+ln+k}$; and*
    - ⋆ *$p-l$ copies of objects $T_i$ or $F_i$, $k+1 \leq i \leq n$, $p-l-1$ copies otherwise, corresponding to the truth assignment assigned to the branch.*

(*b*) *$\mathcal{C}_{n+3np}(0) = \{\alpha_{n+3np}, \beta_{n+3np}\}$, and in $\mathcal{C}_{n+3np}$ there are $2^n$ membranes labelled by 1, such that each of them contains $m$ objects $c_{j,t}$ ($1 \leq j \leq p$, $0 \leq t \leq np-1$), that is, the clauses satisfied by any variable and $n+np-m$ objects #; and $2^n$ membranes labelled by 2 such that each of them contains an object $\gamma_{n+3np}$.*

*Proof.* $(a)$ is going to be demonstrated by induction on $l$

- The base case $l = 0$ is goig to be demonstrated by induction on $k$
  - The base case $k = 1$ is trivial because:
    - at configuration $\mathcal{C}_{n+2np}$ we have: $\mathcal{C}_{n+2np}(0) = \{\alpha_{n+2np}, \beta_{n+2np}\}$ and there exist $2^n$ membranes labelled by 1, such that each of them contains; and $2^n$ membranes labelled by 2 such that each of them contains $n$ objects # the input multiset $cod_{n+2np}(\varphi)$, an object $\gamma_{n+2np}$ and $p$ copies of objects $T_i$ and $F_i$, $1 \leq i \leq n$, representing the correspondent truth assignment to the branch. Then, configuration $\mathcal{C}_{n+2np}$ yields configuration $\mathcal{C}_{n+2np+1}$ by applying the rules:

$$\left.\begin{array}{l}[\ T_1\ x_{1,1,n+2np}\ ]_2 \longrightarrow c_{1,0}[\quad]_2 \\ [\ T_1\ \overline{x}_{1,1,n+2np}\ ]_2 \longrightarrow \#[\quad]_2 \\ [\ T_1\ x^*_{1,1,n+2np}\ ]_2 \longrightarrow \#[\quad]_2 \quad {}^4 \\ [\ F_1\ x_{1,1,n+2np}\ ]_2 \longrightarrow \#[\quad]_2 \\ [\ F_1\ \overline{x}_{1,1,n+2np}\ ]_2 \longrightarrow c_{1,0}[\quad]_2 \\ [\ F_1\ x^*_{1,1,n+2np}\ ]_2 \longrightarrow \#[\quad]_2 \\ [\ \alpha_{n+2np} \to \alpha_{n+2np+1}\ ]_0 \\ [\ \beta_{n+2np} \to \beta_{n+2np+1}\ ]_0 \\ [\ \gamma_{n+2np} \to \gamma_{n+2np+1}\ ]_2 \end{array}\right.$$

$$\left.\begin{array}{l}[\ x_{i,j,n+2np} \to x_{i,j,n+2np+1}\ ]_2 \\ [\ \overline{x}_{i,j,n+2np} \to \overline{x}_{i,j,n+2np+1}\ ]_2 \\ [\ x^*_{i,j,n+2np} \to x^*_{i,j,n+2np+1}\ ]_2 \end{array}\right\} \text{for } 1 \le i \le n, 1 \le j \le p$$

Thus, $\mathcal{C}_{n+2np+1}(0) = \{\alpha_{n+2np+1}, \beta_{n+2np+1}\}$, and there exist $2^n$ membranes labelled by 1 containing $n$ objects # and an object $c_{1,0}$ if the corresponding truth assignment makes true clause 1 with variable 1, another object # otherwise; and $2^n$ membranes labelled by 2 containing the last $n-1$ elements of $cod_{n+2np+1}(\varphi)^1_1$, the input multiset $cod_{n+2np+1}(\varphi)^p_2$, $p$ copies of $T_i$ or $F_i$, being $2 \le i \le n$, and $p-1$ copies of $T_1$ or $F_1$.

- Supposing, by induction, result is true for $k$ $(1 \le k \le n)$
  - $\mathcal{C}_{n+2np+k}(0) = \{\alpha_{n+2np+k}, \beta_{n+2np+k}\}$
  - In $\mathcal{C}_{n+2np+k}$ there are $2^n$ membranes labelled by 1 such that each of them contains
    - ⋆ $m$ objects $c_{1,t}$ $(0 \le t \le k-1)$, that is, the number of variables with the corresponding truth assignment that makes true the input formula $\varphi$; and
    - ⋆ $n+k-m$ objects #.
  - In $\mathcal{C}_{n+2np+k}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ the $(n-k)$-th last elements of $cod_{n+2np+k}(\varphi)^1_1$;
    - ⋆ the input multiset $cod_{n+2np+k}(\varphi)^p_2$;
    - ⋆ an object $\gamma_{n+2np+k}$; and
    - ⋆ $p$ copies of objects $T_i$ or $F_i$, $k+1 \le i \le n$, $p-1$ copies if $1 \le i \le k$, corresponding to the truth assignment assigned to the branch.

  Then, configuration $\mathcal{C}_{n+2np+k}$ yields configuration $\mathcal{C}_{n+2np+k+1}$ by applying the rules:

$$\left.\begin{array}{l}[\ T_k\ x_{1,1,n+2np+k}\ ]_2 \longrightarrow c_{1,0}[\quad]_2 \\ [\ T_k\ \overline{x}_{1,1,n+2np+k}\ ]_2 \longrightarrow \#[\quad]_2 \\ [\ T_k\ x^*_{1,1,n+2np+k}\ ]_2 \longrightarrow \#[\quad]_2 \quad {}^5 \\ [\ F_k\ x_{1,1,n+2np+k}\ ]_2 \longrightarrow \#[\quad]_2 \\ [\ F_k\ \overline{x}_{1,1,n+2np+k}\ ]_2 \longrightarrow c_{1,0}[\quad]_2 \\ [\ F_k\ x^*_{1,1,n+2np+k}\ ]_2 \longrightarrow \#[\quad]_2 \end{array}\right.$$

[4] If $k = 1, l = 0$, then $i = 1, j = 1$, so $2np + n + n(j-1) + (i-1) = n + 2np$

[5] If $l = 0$, then $i = k+1, j = 1$, so $2np + 2n + n(j-1) + (i-1) = 2n + 2np + k$

$[\ \alpha_{n+2np+k} \to \alpha_{n+2np+k+1}\ ]_0$
$[\ \beta_{n+2np+k} \to \beta_{n+2np+k+1}\ ]_0$
$[\ \gamma_{n+2np+k} \to \gamma_{n+2np+k+1}\ ]_2$

$$\left.\begin{array}{l}[\ x_{i,j,n+2np+k} \to x_{i,j,n+2np+k+1}\ ]_2\\ [\ \overline{x}_{i,j,n+2np+k} \to \overline{x}_{i,j,n+2np+k+1}\ ]_2\\ [\ x^*_{i,j,n+2np+k} \to x^*_{i,j,n+2np+k+1}\ ]_2\end{array}\right\}\text{for } 1 \le i \le n, 1 \le j \le p$$

$[\ c_{1,t} \to c_{1,t+1}\ ]_1$ for $0 \le t \le k-1$

Therefore, the following holds

- $\mathcal{C}_{n+2np+k+1} = \{\alpha_{n+2np+k+1}, \beta_{n+2np+k+1}\}$
- In $\mathcal{C}_{n+2np+k+1}$ there are $2^n$ membranes labelled by 1 such that each of them contains
  - ⋆ $m$ objects $c_{1,t}$ $(0 \le t \le k)$, that is, the number of variables with the corresponding truth assignment that makes true the clause $\mathcal{C}_1$; and
  - ⋆ $n + k + 1 - m$ objects $\#$.
- In $\mathcal{C}_{n+2np+k+1}$ there are $2^n$ membranes labelled by 2 such that each of them contains
  - ⋆ the $(n - k + 1)$-th last elements of $cod_{n+2np+k+1}(\varphi)^1_1$;
  - ⋆ the input multiset $cod_{n+2np+k+1}(\varphi)^p_2$,
  - ⋆ an object $\gamma_{n+2np+k+1}$; and
  - ⋆ $p$ copies of objects $T_i$ or $F_i$, $k+2 \le i \le n$, $p-1$ copies if $1 \le i \le k+1$, corresponding to the truth assignment assigned to the branch.

- Supposing, by induction, result is true for $l$ $(0 \le l \le p-1)$
  - The base case $k = 1$ is trivial because:
    - at configuration $\mathcal{C}_{n+2np+(l+1)n}$ we have: $\mathcal{C}_{n+2np+(l+1)n}(0) = \{\alpha_{n+2np+(l+1)n}, \beta_{n+2np+(l+1)n}\}$ and there exist $2^n$ membranes labelled by 1 containing $m$ objects $c_{j,t}$ $(1 \le j \le l,\ 0 \le t \le ln-1)$, that is, the number of variables with the corresponding truth assignment that makes true the clauses from $\mathcal{C}_1$ to $\mathcal{C}_l$ and $n + (l+1)n - m$ objects $\#$; and $2^n$ membranes labelled by 2 containing the input multiset $cod_{n+2np+(l+1)n}(\varphi)^p_{l+1}$, an object $\gamma_{n+2np+(l+1)n}$ and $p - l$ copies of objects $T_i$ or $F_i$, $1 \le i \le n$. Then, configuration $\mathcal{C}_{n+2np+(l+1)n}$ yields configuration $\mathcal{C}_{n+2np+(l+1)n+1}$ by applying the rules:

$[\ T_1\ x_{1,1,n+2np+(l+1)n}\ ]_2 \longrightarrow c_{l+1,0}[\ \ ]_2$
$[\ T_1\ \overline{x}_{1,1,n+2np+(l+1)n}\ ]_2 \longrightarrow \#[\ \ ]_2$
$[\ T_1\ x^*_{1,1,n+2np+(l+1)n}\ ]_2 \longrightarrow \#[\ \ ]_2$
$[\ F_1\ x_{1,1,n+2np+(l+1)n}\ ]_2 \longrightarrow c_{l+1,0}[\ \ ]_2$
$[\ F_1\ \overline{x}_{1,1,n+2np+(l+1)n}\ ]_2 \longrightarrow \#[\ \ ]_2$
$[\ F_1\ x^*_{1,1,n+2np+(l+1)n}\ ]_2 \longrightarrow \#[\ \ ]_2$

$[\ \alpha_{n+2np+(l+1)n} \to \alpha_{n+2np+(l+1)n+1}\ ]_0$
$[\ \beta_{n+2np+(l+1)n} \to \beta_{n+2np+(l+1)n+1}\ ]_0$
$[\ \gamma_{n+2np+(l+1)n} \to \gamma_{n+2np+(l+1)n+1}\ ]_2$

$$\left.\begin{array}{l}[\ x_{i,j,n+2np+(l+1)n} \to x_{i,j,n+2np+(l+1)n+1}\ ]_2\\ [\ \overline{x}_{i,j,n+2np+(l+1)n} \to \overline{x}_{i,j,n+2np+(l+1)n+1}\ ]_2\\ [\ x^*_{i,j,n+2np+(l+1)n} \to x^*_{i,j,n+2np+(l+1)n+1}\ ]_2\end{array}\right\}\text{for } \begin{array}{l}1 \le i \le n\\ 1 \le j \le p\end{array}$$

$$[\ c_{j,t} \to c_{1,t+1}\ ]_1 \text{ for } 1 \le j \le l+1, 0 \le t \le ln-1$$

Thus, $\mathcal{C}_{n+2np+(l+1)n+1}(0) = \{\alpha_{n+2np+(l+1)n+1}, \beta_{n+2np+(l+1)n+1}\}$, and there exist $2^n$ membranes labelled by 1 containing $m$ objects $c_{j,t}$ ($1 \le j \le l+1$, $0 \le t \le ln$), that is, the number of variables with the corresponding truth assignment that makes true the clauses from $\mathcal{C}_1$ to $\mathcal{C}_{l+1}$ and $n+(l+1)n+1-m$ objects #; and $2^n$ membranes labelled by 2 containing the last $n-1$ elements of $cod_{n+2np+(l+1)n+1}(\varphi)^{l+1}_{l+1}$, the input multiset $cod_{n+2np+(l+1)n+1}(\varphi)^{p}_{l+2}$, $p-l$ copies of $T_i$ or $F_i$, being $2 \le i \le n$, and $p-l-1$ copies of $T_1$ or $F_1$.

- Supposing, by induction, result is true for $k$ ($1 \le k \le n$)
  - $\mathcal{C}_{n+2np+(l+1)n+k}(0) = \{\alpha_{n+2np+(l+1)n+k}, \beta_{n+2np+(l+1)n+k}\}$
  - In $\mathcal{C}_{n+2np+(l+1)n+k}$ there are $2^n$ membranes labelled by 1 such that each of them contains
    - ⋆ $m$ objects $c_{j,t}$ ($1 \le j \le l+1$, $0 \le t \le ln+k-1$), that is, the number of variables with the corresponding truth assignment that makes true clauses from $\mathcal{C}_1$ to $\mathcal{C}_{l+1}$; and
    - ⋆ $n+(l+1)n+k+1-m$ objects #.
  - In $\mathcal{C}_{n+2np+(l+1)n+k}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ the $(n-k)$-th last elements of $cod_{n+2np+(l+1)n+k}(\varphi)^{l+1}_{l+1}$;
    - ⋆ the input multiset $cod_{n+2np+(l+1)n+k}(\varphi)^{p}_{l+2}$,
    - ⋆ an object $\gamma_{n+2np+(l+1)n+k}$; and
    - ⋆ $p-l$ copies of objects $T_i$ or $F_i$, $k+1 \le i \le n$, $p-l-1$ copies if $1 \le i \le k$, corresponding to the truth assignment assigned to the branch.

  Then, configuration $\mathcal{C}_{n+2np+(l+1)n+k}$ yields configuration $\mathcal{C}_{n+2np+(l+1)n+k+1}$ by applying the rules:

$$\begin{array}{l}[\ T_k\ x_{1,1,n+2np+(l+1)n+k}\ ]_2 \longrightarrow c_{l+1}[\quad]_2\\ [\ T_k\ \overline{x}_{1,1,n+2np+(l+1)n+k}\ ]_2 \longrightarrow \#[\quad]_2\\ [\ T_k\ x^*_{1,1,n+2np+(l+1)n+k}\ ]_2 \longrightarrow \#[\quad]_2\\ [\ F_k\ x_{1,1,n+2np+(l+1)n+k}\ ]_2 \longrightarrow \#[\quad]_2\\ [\ F_k\ \overline{x}_{1,1,n+2np+(l+1)n+k}\ ]_2 \longrightarrow c_{l+1}[\quad]_2\\ [\ F_k\ x^*_{1,1,n+2np+(l+1)n+k}\ ]_2 \longrightarrow \#[\quad]_2\end{array}$$

$$\begin{array}{l}[\ \alpha_{n+2np+(l+1)n+k} \to \alpha_{n+2np+(l+1)n+k+1}\ ]_0\\ [\ \beta_{n+2np+(l+1)n+k} \to \beta_{n+2np+(l+1)n+k+1}\ ]_0\\ [\ \gamma_{n+2np+(l+1)n+k} \to \gamma_{n+2np+(l+1)n+k+1}\ ]_2\end{array}$$

$$\left.\begin{array}{l}[\ x_{i,j,n+2np+(l+1)n+k} \to x_{i,j,n+2np+(l+1)n+k+1}\ ]_2\\ [\ \overline{x}_{i,j,n+2np+(l+1)n+k} \to \overline{x}_{i,j,n+2np+(l+1)n+k+1}\ ]_2\\ [\ x^*_{i,j,n+2np+(l+1)n+k} \to x^*_{i,j,n+2np+(l+1)n+k+1}\ ]_2\end{array}\right\}\text{for } \begin{array}{l}1 \le i \le n\\ 1 \le j \le p\end{array}$$

$$[\ c_{j,t} \to c_{j,t+1}\ ]_1 \text{ for } 1 \le j \le l+1, 0 \le t \le ln+k-1$$

Therefore, the following holds

- $\mathcal{C}_{n+2np+(l+1)n+k+1}(0) = \{\alpha_{n+2np+(l+1)n+k+1}, \beta_{n+2np+(l+1)n+k+1}\}$
- In $\mathcal{C}_{n+2np+(l+1)n+k+1}$ there are $2^n$ membranes labelled by 1 such that each of them contains
  - $\star$ $m$ objects $c_{j,t}$ $(1 \leq j \leq l+1,\ 0 \leq t \leq ln+k)$, that is, the number of variables with the corresponding truth assignment that makes true clauses from $\mathcal{C}_1$ to $\mathcal{C}_{l+1}$; and
  - $\star$ $n + (l+1)n + k + 1 - m$ objects $\#$.
- In $\mathcal{C}_{n+2np+(l+1)n+k+1}$ there are $2^n$ membranes labelled by 2 such that each of them contains
  - $\star$ the $(n-(k+1))$-th last elements of $cod_{n+2np+(l+1)n+k+1}(\varphi)^{l+1}_{l+1}$,
  - $\star$ the input multiset $cod_{n+2np+(l+1)n+k+1}(\varphi)^{p}_{l+1}$,
  - $\star$ an object $\gamma_{n+2np+(l+1)n+k+1}$;
  - $\star$ $p-l$ copies of objects $T_i$ or $F_i$, $k+2 \leq i \leq n$, $p-l-1$ copies if $1 \leq i \leq k+1$, corresponding to the truth assignment assigned to the branch.

- In order to prove $(b)$ it is enough to notice that, on the one hand, from $(a)$ configuration $\mathcal{C}_{n+3np-1}$ [6] holds:
  - $\mathcal{C}_{n+3np-1}(0) = \{\alpha_{n+3np-1}, \beta_{n+3np-1}\}$
  - In $\mathcal{C}_{n+3np-1}$ there are $2^n$ membranes labelled by 1 such that each of them contains
    - $\star$ $m$ objects $c_{j,t}$ $(1 \leq j \leq p, 0 \leq t \leq np-2)$, that is, the number of variables with the corresponding truth assignment that makes true clauses from $\mathcal{C}_1$ to $\mathcal{C}_p$; and
    - $\star$ $n + np - 1 - m$ objects $\#$.
  - In $\mathcal{C}_{n+3np-1}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - $\star$ the last element of $cod_{n+3np-1}(\varphi)^p_p$;
    - $\star$ an object $\gamma_{n+3np-1}$; and
    - $\star$ an object $T_n$ or $F_n$ corresponding to the truth assignment assigned to the branch.

  Then, configuration $\mathcal{C}_{n+3np-1}$ yields $\mathcal{C}_{n+3np}$ by applying the rules:

  $$\begin{array}{l}
  [\ T_n\ x_{n,p,n+3np-1}\ ]_2 \longrightarrow c_{p,0}[\quad]_2 \\
  [\ T_n\ \overline{x}_{n,p,n+3np-1}\ ]_2 \longrightarrow \#[\quad]_2 \\
  [\ T_n\ x^*_{n,p,n+3np-1}\ ]_2 \longrightarrow \#[\quad]_2 \\
  [\ F_n\ x_{n,p,n+3np-1}\ ]_2 \longrightarrow \#[\quad]_2 \\
  [\ F_n\ \overline{x}_{n,p,n+3np-1}\ ]_2 \longrightarrow c_{p,0}[\quad]_2 \\
  [\ F_n\ x^*_{n,p,n+3np-1}\ ]_2 \longrightarrow \#[\quad]_2 \\
  [\ \alpha_{n+2np+(l+1)n+k} \to \alpha_{n+2np+(l+1)n+k+1}\ ]_0 \\
  [\ \beta_{n+2np+(l+1)n+k} \to \beta_{n+2np+(l+1)n+k+1}\ ]_0 \\
  [\ \gamma_{n+2np+(l+1)n+k} \to \gamma_{n+2np+(l+1)n+k+1}\ ]_2
  \end{array}$$

[6] Note that $n + 3np - 1 = n + 3n(p-1) + (n-1)$

$$\left.\begin{array}{l}[\ x_{i,j,n+2np+(l+1)n+k} \to x_{i,j,n+2np+(l+1)n+k+1}\ ]_2\\ [\ \overline{x}_{i,j,n+2np+(l+1)n+k} \to \overline{x}_{i,j,n+2np+(l+1)n+k+1}\ ]_2\\ [\ x^*_{i,j,n+2np+(l+1)n+k} \to x^*_{i,j,n+2np+(l+1)n+k+1}\ ]_2\end{array}\right\}\text{for}\ \begin{array}{c}1 \le i \le n\\ 1 \le j \le p\end{array}$$

$[\ c_{j,t} \to c_{j,t+1}\ ]_2$ for $1 \le j \le l+1, 0 \le t \le ln+k-1$

Therefore, the following holds
- $\mathcal{C}_{n+3np}(0) = \{\alpha_{n+3np}, \beta_{n+3np}\}$
- In $\mathcal{C}_{n+3np}$ there are $2^n$ membranes labelled by 1 such that each of them contains
  - ⋆ $m$ objects $c_{j,t}$ $(1 \le j \le p,\ 0 \le t \le np-1)$, that is, the number of variables with the corresponding truth assignment that makes true clauses from $\mathcal{C}_1$ to $\mathcal{C}_p$; and
  - ⋆ $n+np-m$ objects $\#$.
- In $\mathcal{C}_{n+3np}$ there are $2^n$ membranes labelled by 2 such that each of them contains an object $\gamma_{n+3np}$.

□

### 5.3 Second checking stage

At this stage, started at configuration $\mathcal{C}_{n+3np}$, we try to determine the truth assignments that make true the input formula $\varphi$, using rules from **5.6**. We are going to divide this stage in two phases. The first one will be devoted to send in all the objects $c_j$, for $1 \le j \le p$ in order to get them ready for the next phase.

Let $k = ln+i$ $(0 \le l \le p-1, 1 \le i \le n)$, so we can refer to each clause $(l-1)$ when we are doing the verification. Let $m = \sum_{j=1}^{p} m_j$, being $m_j$ the number of objects $c_{j,k}$ in each membrane 1 at step $\mathcal{C}_{n+3np}$. In this stage, we cannot be sure of how many objects $c_{l+1,k}$ are present at each membrane when $i \neq 0$ [7], so if we cannot be sure of that, we are going to say that there are $\widetilde{m}_j$ (remember that $\widetilde{m}_j$ is always less than or equal to $m_j$) objects within membrane 1. We will ignore objects $\#$ since they have no effect from here.

**Proposition 5.** *Let $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_q)$ be a computation of the system $\Pi(s(\varphi))$ with input multiset $cod(\varphi)$.*

*(a) For each $k$ $(1 \le k \le np-1)$ at configuration $\mathcal{C}_{n+3np+k}$ we have the following:*
- $\mathcal{C}_{n+3np+k}(0) = \{\alpha_{n+3np+k}, \beta_{n+3np+k}\}$
- *There are $2^n$ membranes labelled by 1 such that each of them contains $\widetilde{m}_{l+1}$ objects $c_{l+1,t}$ $((p-1)n+1 \le t \le np-1)$ and $m_j$ objects $c_{j,t}$ $(l+2 \le j \le p, ln+i \le t \le np-1)$*
- *There are $2^n$ membranes labelled by 2 such that each of them contains*
  - ⋆ *an object $\gamma_{n+3np+k}$; and*
  - ⋆ *$m_j$ objects $c_j$ for $1 \le j \le l$ and $m_{l+1} - \widetilde{m}_{l+1}$ objects $c_{l+1}$*

[7] That is because objects $c_{j,k}$ do not have to be created consecutively.

*(b)* $\mathcal{C}_{n+4np}(0) = \{\alpha_{n+4np}, \beta_{n+4np}\}$*, there are* $2^n$ *empty membranes labelled by 1; and* $2^n$ *membranes labelled by 2, such that each of them contains* $m$ *objects* $c_j$ *(*$1 \leq j \leq p$*) and an object* $\gamma_{n+4np}$*.*

*Proof.* $(a)$ is going to be demonstrated by induction on $k$

- The base case $k = 1$ is trivial because: At configuration $\mathcal{C}_{n+3np}$ we have: $\mathcal{C}_{n+3np}(0) = \{\alpha_{n+3np}, \beta_{n+3np}\}$ and there exist $2^n$ membranes labelled by 1 containing $m$ objects $c_{j,t}$ $(1 \leq j \leq k, 0 \leq t \leq np-1)$; and $2^n$ membranes labelled by 2 containing an object $\gamma_{n+3np}$. Then, configuration $\mathcal{C}_{n+3np}$ yields configuration $\mathcal{C}_{n+3np+1}$ by applying the rules:
  $[\ \alpha_{n+3np} \to \alpha_{n+3np+1}\ ]_0$
  $[\ \beta_{n+3np} \to \beta_{n+3np+1}\ ]_0$
  $[\ \gamma_{n+3np} \to \gamma_{n+3np+1}\ ]_2$
  $[c_{j,t} \longrightarrow c_{j,t+1}]_1$, for $1 \leq j \leq p$, $0 \leq k \leq np-2$
  $c_{1,np-1}[\quad]_2 \longrightarrow [c_1]_2$
  Thus, $\mathcal{C}_{n+3np+1}(0) = \{\alpha_{n+3np+1}, \beta_{n+3np+1}\}$, and there exist $2^n$ membranes labelled by 2 containing $\widetilde{m}_1$ objects $c_1$ and $m_j$ objects $c_j$ $(2 \leq j \leq p)$; and $2^n$ membranes labelled by 1 containing an object $\gamma_{n+3np+1}$ and $m_1 - \widetilde{m}_1$ objects $c_1$ [8]. Hence, the result holds for $k = 1$.
- Supposing, by induction, result is true for $k$ $(1 \leq k \leq np-1)$
  - $\mathcal{C}_{n+3np+k}(0) = \{\alpha_{n+3np+k}, \beta_{n+3np+k}\}$
  - In $\mathcal{C}_{n+3np+k}$ there are $2^n$ membranes labelled by 1 such that each of them contains $\widetilde{m}_{l+1}$ objects $c_{l+1,t}$ $((p-1)n+1 \leq t \leq np-1)$ and $m_j$ objects $c_{j,t}$ $(l+2 \leq j \leq p, ln+i \leq t \leq np-1)$.
  - In $\mathcal{C}_{n+3np+k}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ an object $\gamma_{n+3np+k}$; and
    - ⋆ $m_j$ objects $c_j$ for $1 \leq j \leq l$ and $m_{l+1} - \widetilde{m}_{l+1}$ objects $c_{l+1}$.

    Then, configuration $\mathcal{C}_{n+3np+k}$ yields configuration $\mathcal{C}_{n+3np+k}$ by applying the rules:
    $[\ \alpha_{n+3np+k} \to \alpha_{n+3np+k+1}\ ]_0$
    $[\ \beta_{n+3np+k} \to \beta_{n+3np+k+1}\ ]_0$
    $[\ \gamma_{n+3np+k} \to \gamma_{n+3np+k+1}\ ]_2$
    $[c_{j,t} \longrightarrow c_{j,t+1}]_1$, for $l+1 \leq j \leq p$, $0 \leq k \leq np-2$
    $c_{l+1,np-1}[\quad]_2 \longrightarrow [c_{l+1}]_2$
    Therefore, the following holds
  - $\mathcal{C}_{n+3np+k+1}(0) = \{\alpha_{n+3np+k+1}, \beta_{n+3np+k+1}\}$
  - In $\mathcal{C}_{n+3np+k+1}$ there are $2^n$ membranes labelled by 1 such that each of them contains $\widetilde{m}_{l+1}$ objects $c_{l+1,t+1}$ $((p-1)n+1 \leq t \leq np-1)$ and $m_j$ objects $c_{j,t+1}$ $(l+2 \leq j \leq p, ln+i \leq t \leq np-1)$.

[8] That is, if the truth assignment of variable 1 made true clause 1, then an object $c_{1,0}$ were created at $(2n+2np+1)$-th step, and it is going to be sent to the corresponding membrane 2. So, $m_1 - \widetilde{m}_1$ can be 0 or 1 in this step.

- In $\mathcal{C}_{n+3np+k+1}$ there are $2^n$ membranes labelled by 2 such that each of them contains
  - $\star$ an object $\gamma_{n+3np+k+1}$; and
  - $\star$ $m_j$ objects $c_j$ for $1 \leq j \leq l$ and $m_{l+1} - \widetilde{m}_{l+1}$ objects $c_{l+1}$.

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

- In order to prove $(b)$ it is enough to notice that, on the one hand, from $(a)$ configuration $\mathcal{C}_{n+4np-1}$ holds:
  - $\mathcal{C}_{n+4np-1}(0) = \{\alpha_{n+4np-1}, \beta_{n+4np-1}\}$
  - In $\mathcal{C}_{n+4np-1}$ there are $2^n$ membranes labelled by 1 such that each of them contains
  - In $\mathcal{C}_{n+4np-1}$ there are $2^n$ membranes labelled by 2 such that each of them contains $\widetilde{m}_p$ objects $c_{p,np}$.
    - $\star$ an object $\gamma_{n+4np-1}$; and
    - $\star$ $m_j$ objects $c_j$ for $1 \leq j \leq p-1$ and $m_p - \widetilde{m}_p$ [9] objects $c_p$.

  Then, configuration $\mathcal{C}_{n+4np-1}$ yields configuration $\mathcal{C}_{n+4np}$ by applying the rules:

  $[\ \alpha_{n+4np-1} \rightarrow \alpha_{n+4np}\ ]_0$
  $[\ \beta_{n+4np-1} \rightarrow \beta_{n+4np}\ ]_0$
  $[\ \gamma_{n+4np-1} \rightarrow \gamma_{n+4np}\ ]_2$
  $c_{p,np}[\quad]_2 \longrightarrow [c_p]_2$

  Then, we have $\mathcal{C}_{n+4np}(0) = \{\alpha_{n+4np}, \beta_{n+4np}\}$, and there exist $2^n$ empty membranes labelled by 1; and there exist $2^n$ membranes labelled by 2 containing an object $\gamma_{n+4np}$ and $m$ objects $c_j$ $(1 \leq j \leq p)$.

$\square$

When objects $c_j$ are within the membranes labelled by 2, we can start to check if all the clauses of the input formula $\varphi$ are satisfied by any truth assignment. As we use objects $c_j$ to denote that clause $C_j$ has been satisfied by some variable, it can be possible that some $c_j$ are missing, that is, that for some $j$, $1 \leq j \leq p$, $c_j$ does not appear in any membrane labelled by 2 in $\mathcal{C}_{2n+4np}$. Let $\widetilde{j}$ be the index $j$ [10] of that clause. It is going to take $2p$ steps.

**Proposition 6.** *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 $cod(\varphi)$.*

$(a_0)$ *For each $2k+1$ $(0 \leq k \leq p-1)$ at configuration $\mathcal{C}_{n+4np+2k+1}$ we have the following:*
- $\mathcal{C}_{n+4np+2k+1}(0) = \{\alpha_{n+4np+2k+1}, \beta_{n+4np+2k+1}\}$
- *There are $2^n$ membranes labelled by 1 such that each of them contains an object $d_{k+1}$ if and only if the truth assignment associated to the branch makes true the first $k+1$ clauses.*
- *There are $2^n$ membranes labelled by 2 such that each of them contains*

[9] In this case, $\widetilde{m}_p$ can only take two values: 0 or 1.
[10] If $\widetilde{j}$ is not defined, we are going to suposse that it is equal to $p+1$.

- ⋆ *an object* $\gamma_{n+4np}$ *or* $d_{\widetilde{j}-1}$ *(respectively, an object* $d_k$*) if the corresponding truth assignment does not make true (resp., makes true) the clause* $C_1$ *or* $C_j$ $(2 \leq j \leq p)$ *(resp., the first* $k$ *clauses); and*
- ⋆ $m_j - 1$ *objects* $c_j$ *for* $1 \leq j \leq min(\widetilde{j}, k+1)$ *and* $m_j$ *objects* $c_j$ *for* $min(\widetilde{j}, k+2) \leq j \leq p$.

$(a_1)$ *For each* $2k$ $(1 \leq k \leq p-2)$ *at configuration* $\mathcal{C}_{n+4np+2k}$ *we have the following:*
- $\mathcal{C}_{n+4np+2k}(0) = \{\alpha_{n+4np+2k}, \beta_{n+4np+2k}\}$
- *There are* $2^n$ *empty membranes labelled by 1.*
- *There are* $2^n$ *empty membranes labelled by 2 such that each of them contains*
  - ⋆ *an object* $\gamma_{n+4np}$ *or* $d_{\widetilde{j}-1}$ *if the corresponding truth assignment does not make true the clause* $C_1$ *or* $C_j$ $(2 \leq j \leq p)$*; and*
  - ⋆ $m_j - 1$ *objects* $c_j$ *for* $1 \leq j \leq min(\widetilde{j}, k)$ *and* $m_j$ *objects* $c_j$ *for* $min(\widetilde{j}, k+1) \leq j \leq p$.

$(b)$ $\mathcal{C}_{n+4np+2p-1}(0) = \{\alpha_{n+4np+2p-1}, \beta_{n+4np+2p-1}\}$*, and in* $\mathcal{C}_{n+4np+2p-1}$ *there are* $2^n$ *membranes labelled by 1, such that each of them contains an object* $d_p$ *if and only if the corresponding truth assignment makes true the input formula* $\varphi$ *(*$d_{\widetilde{j}-1}$ *otherwise); and* $2^n$ *membranes labelled by 2, such that each of them contains* $m_j - 1$ *objects* $c_j$ *for* $1 \leq j \leq min(\widetilde{j}, p+1)$*,* $m_j$ *objects* $c_j$ *for* $min(\widetilde{j}, p+1) \leq j \leq p$ *and an object* $\gamma_{n+4np}$ *(respectively,* $d_{\widetilde{j}}$*) if clause* $C_1$ *(resp.,* $C_j$*) is not satisfied by the corresponding truth assignment.*

*Proof.* $(a)$ is going to be demonstrated by induction on $k$

- The base case $k = 1$ is trivial because:

  $(a_0)$ at configuration $\mathcal{C}_{n+4np}$ we have: $\mathcal{C}_{n+4np}(0) = \{\alpha_{n+4np}, \beta_{n+4np}\}$ and there exist $2^n$ empty membranes labelled by 1; and there exist $2^n$ membranes labelled by 2 containing an object $\gamma_{n+4np}$ and $m$ objects $c_j$ $(1 \leq j \leq p)$. Then, configuration $\mathcal{C}_{n+4np}$ yields configuration $\mathcal{C}_{n+4np+1}$ by applying the rules:

  $[\ \alpha_{n+4np} \rightarrow \alpha_{n+4np+1}\ ]_0$
  $[\ \beta_{n+4np} \rightarrow \beta_{n+4np+1}\ ]_0$
  $[\ \gamma_{4np+2n}\ c_1\ ]_2 \longrightarrow d_1[\quad]_2$

  $(a_1)$ at $\mathcal{C}_{n+4np+1}(0) = \{\alpha_{n+4np+1}, \beta_{n+4np+1}\}$ and there exist $2^n$ membranes labelled by 1 containing an object $d_1$ if and only if there was at least one object $c_1$ within membrane labelled by 1 at configuration $\mathcal{C}_{n+4np}$; and $2^n$ membranes labelled by 2 containing an object $\gamma_{n+4np}$ if and only if there were no objects $c_1$ at configuration $\mathcal{C}_{n+4np}$, $m_1 - 1$ (respectively, $m_1$) objects $c_1$ if there was any object $c_j$ in this membrane in the previous configuration (resp., $m_1$) and $m_j$ objects $c_j$ for $2 \leq j \leq p$. Then, the configuration $\mathcal{C}_{n+4np+1}$ yields configuration $\mathcal{C}_{n+4np+2}$ by applying the rules:

  $[\ \alpha_{n+4np+1} \rightarrow \alpha_{n+4np+2}\ ]_0$
  $[\ \beta_{n+4np+1} \rightarrow \beta_{n+4np+2}\ ]_0$
  $d_1\ [\quad]_2 \longrightarrow [\ d_1\ ]_2$

  Thus, $\mathcal{C}_{n+4np+2}(0) = \{\alpha_{n+4np+2}, \beta_{n+4np+2}\}$, and there exist $2^n$ empty membranes labelled by 1; and there exist $2^n$ membranes labelled by 2 containing an object $d_1$ (respectively, $\gamma_{n+4np}$) if the corresponding truth assignment makes true (resp., doesn't make true) clause $C_1$, $m_1 - 1$ (resp., $m_1$) objects $c_1$ and $m_j$ objects $c_j$ for $1 \leq j \leq p$. Hence, the result holds for $k = 1$.

- Supposing, by induction, result is true for $k$ $(0 \leq k \leq p-1)$
  - $\mathcal{C}_{n+4np+2k}(0) = \{\alpha_{n+4np+2k}, \beta_{n+4np+2k}\}$
  - In $\mathcal{C}_{n+4np+2k}$ there are $2^n$ empty membranes labelled by 1.
  - In $\mathcal{C}_{n+4np+2k}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ an object $\gamma_{n+4np}$ or $d_{\widetilde{j}-1}$ (respectively, an object $d_k$) if the corresponding truth assignment does not make true (resp., makes true) the clause $C_1$ or $C_j$ $(2 \leq j \leq p)$ (resp., the first $k$ clauses); and
    - ⋆ $m_j - 1$ objects $c_j$ for $1 \leq j \leq min(\widetilde{j}, k+1)$ and $m_j$ objects $c_j$ for $min(\widetilde{j}, k+2) \leq j \leq p$.

  Then, configuration $\mathcal{C}_{n+4np+2k}$ yields configuration $\mathcal{C}_{n+4np+2k+1}$ by applying the rules:

  $[\ \alpha_{n+4np+2k} \to \alpha_{n+4np+2k+1}\ ]_0$
  $[\ \beta_{n+4np+2k} \to \beta_{n+4np+2k+1}\ ]_0$
  $[\ d_k\ c_{k+1}\ ]_2 \longrightarrow d_{k+1}[\quad]_2$

  Therefore, the following holds
  - $\mathcal{C}_{n+4np+2k+1}(0) = \{\alpha_{n+4np+2k+1}, \beta_{n+4np+2k+1}\}$
  - In $\mathcal{C}_{n+4np+2k+1}$ there are $2^n$ membranes labelled by 1 such that each of them contains an object $d_{k+1}$ if and only if the corresponding truth assignment makes true the first $k+1$ clauses.
  - In $\mathcal{C}_{n+4np+2k+1}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ an object $\gamma_{n+4np}$ or $d_{\widetilde{j}-1}$ if the corresponding truth assignment does not make true the clause $C_1$ or $C_j$ $(2 \leq j \leq p)$; and
    - ⋆ $m_j - 1$ objects $c_j$ for $1 \leq j \leq min(\widetilde{j}, k)$ and $m_j$ objects $c_j$ for $min(\widetilde{j}, k+1) \leq j \leq p$.

  Then, configuration $\mathcal{C}_{n+4np+2k+1}$ yields $\mathcal{C}_{n+4np+2k+2}$ by applying the rules:

  $[\ \alpha_{n+4np+2k+1} \to \alpha_{n+4np+2k+2}\ ]_0$
  $[\ \beta_{n+4np+2k+1} \to \beta_{n+4np+2k+2}\ ]_0$
  $d_{k+1}[\quad]_2 \longrightarrow [\ d_{k+1}\ ]_2$

  Therefore, the following holds
  - $\mathcal{C}_{n+4np+2k+2}(0) = \{\alpha_{n+4np+2k+2}, \beta_{n+4np+2k+2}\}$
  - In $\mathcal{C}_{n+4np+2k+2}$ there are $2^n$ empty membranes labelled by 1.
  - In $\mathcal{C}_{n+4np+2k+2}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - ⋆ an object $\gamma_{n+4np}$ or $d_{\widetilde{j}-1}$ (respectively, an object $d_{k+1}$) if the corresponding truth assignment does not make true (resp., makes true) the clause $C_1$ or $C_j$ $(2 \leq j \leq p)$ (resp., the first $k+1$ clauses); and

  * $\star$ $m_j - 1$ objects $c_j$ for $1 \leq j \leq min(\widetilde{j}, k+2)$ and $m_j$ objects $c_j$ for $min(\widetilde{j}, k+3) \leq j \leq p$.

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

- In order to prove $(b)$ it is enough to notice that, on the one han, from $(a)$ configuration $\mathcal{C}_{n+4np+2p-2}$ holds:
  - $\mathcal{C}_{n+4np+2p-2}(0) = \{\alpha_{n+4np+2p-2}, \beta_{n+4np+2p-2}\}$
  - In $\mathcal{C}_{n+4np+2p-2}$ there are $2^n$ empty membranes labelled by 1.
  - In $\mathcal{C}_{n+4np+2p-2}$ there are $2^n$ membranes labelled by 2 such that each of them contains
    - an object $\gamma_{n+4np}$ or $d_{\widetilde{j}-1}$ (respectively, $d_{p-1}$) if the corresponding truth assignment does not make true the clause $C_1$ or $C_j$ $(2 \leq j \leq p-1)$ (resp., makes true clauses $C_j$ $(1 \leq j \leq p-1)$); and
    - $m_j - 1$ objects $c_j$ for $1 \leq j \leq min(\widetilde{j}, p-1)$ and $m_j$ objects $c_j$ for $min(\widetilde{j}, p) \leq j \leq p$

  Then, configuration $\mathcal{C}_{n+4np+2p-2}$ yields configuration $\mathcal{C}_{n+4np+2p-1}$ by applying the rules:

  $[\ \alpha_{n+4np+2p-2} \rightarrow \alpha_{n+4np+2p-1}\ ]_0$
  $[\ \beta_{n+4np+2p-2} \rightarrow \beta_{n+4np+2p-1}\ ]_0$
  $[\ d_{p-1}\ c_p\ ]_2 \longrightarrow d_p\ [\quad]_2$

  Then, we have $\mathcal{C}_{n+4np+2p-1}(0) = \{\alpha_{n+4np+2p-1}, \beta_{n+4np+2p-1}\}$, and in $\mathcal{C}_{n+4np+2p-1}$ there are $2^n$ membranes labelled by 1, such that each of them contains an object $d_p$ if and only if the corresponding truth assignment makes true the input formula $\varphi$ ($d_{\widetilde{j}-1}$ otherwise); and $2^n$ membranes labelled by 2, such that each of them contains $m_j - 1$ objects $c_j$ for $1 \leq j \leq min(\widetilde{j}, p+1)$, $m_j$ objects $c_j$ for $min(\widetilde{j}, p+1) \leq j \leq p$ and an object $\gamma_{n+4np}$ (respectively, $d_{\widetilde{j}}$) if clause $C_1$ (resp., $C_j$) is not satisfied by the corresponding truth assignment.

$\square$

### 5.4 Output stage

The output phase starts at configuration $\mathcal{C}_{n+4np+2p-1}$, and takes exactly two steps when there is an affirmative answer and three steps when there is a negative one. Rules from **5.7** are devoted to compute this stage.

- *Affirmative answer*: In this case, at configuration $\mathcal{C}_{n+4np+2p-1}$, in some membrane 1 there is an object $d_p$. By applying the rule $[\ d_p\ ]_1 \longrightarrow d_p[\quad]_1$ (at the same time that $[\ \alpha_{n+4np+2p-1} \rightarrow \alpha_{n+4np+2p}\ ]_0$ and $[\ \beta_{n+4np+2p-1} \rightarrow \beta_{n+4np+2p}\ ]_0$ are executed), an object $d_p$ is produced in membrane 0. Then by applying the rules $[\ \alpha_{4np+n+2p}\ d_p\ ]_0 \longrightarrow \mathtt{yes}[\quad]_0$ and $[\ \beta_{n+4np+2p} \rightarrow \beta_{n+4np+2p+1}\ ]_0$, an object yes is released to environment and the computation halts.

- *Negative answer*: In this case, at configuration $\mathcal{C}_{n+4np+2p-1}$, there are no membranes labelled by 1 that contains an object $d_p$, so the only rules executed are $[\ \alpha_{n+4np+2p-1} \to \alpha_{n+4np+2p}\ ]_0$ and $[\ \beta_{n+4np+2p-1} \to \beta_{n+4np+2p}\ ]_0$. Rule $[\ \beta_{n+4np+2p} \to \beta_{n+4np+2p+1}\ ]_0$ is executed in the next step. Thus, at configuration $\mathcal{C}_{n+4np+2p+1}$ in membrane labelled by 0 we execute have a copy of object $\alpha_{n+4np+2p}$ and a copy of object $\beta_{n+4np+2p+1}$. By applying the rule $[\ \alpha_{4np+n+2p}\ \beta_{4np+n+2p+1}]_0 \longrightarrow \texttt{no}[\quad]_0$ an object `no` is released to the environment and then the computation halts.

### 5.5 Result

**Theorem 1.** $\texttt{SAT} \in \mathbf{PMC}_{\mathcal{DAM}^0(+e_s, mcmp_{out}, -d, +n)}$.

*Proof.* The family $\mathbf{\Pi}$ of P systems previously constructed verifies the following:

(a) 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: $\frac{15n^2p^2}{2} + 3n^2p + 3n^2 + np^2 + \frac{35np}{2} + 5n + 6p + 6 \in \Theta(n^2p^2)$.
  - Initial number of membranes: $3 \in \Theta(1)$.
  - Initial number of objects in membranes: $3np + n + 3 \in \Theta(np)$.
  - Number of rules: $\frac{15n^2p^2}{2} + 7n^2p + np^2 + \frac{33np}{2} + 4n + 6p + 4 \in \Theta(n^2p^2)$.
  - Maximal number of objects involved in any rule: $3 \in \Theta(1)$.

(b) The family $\mathbf{\Pi}$ is polynomially bounded with regard to ($\texttt{SAT}$, $cod, s$): indeed for each instance $\varphi$ of the $\texttt{SAT}$ problem, any computation of the system $\mathbf{\Pi}(s(\varphi))$ with input multiset $cod(\varphi)$ takes at most $2n + 4np + 2p + 5$ computation steps.

(e) The family $\mathbf{\Pi}$ is sound with regard to ($\texttt{SAT}$, $cod, s$): indeed for each instance $\varphi$ of the $\texttt{SAT}$ problem, if the computation of $\mathbf{\Pi}(s(\varphi)) + cod(\varphi)$ is an accepting computation, then $\varphi$ is satisfiable.

(f) The family $\mathbf{\Pi}$ is complete with regard to ($\texttt{SAT}$, $cod, s$): indeed, for each instance $\varphi$ of the $\texttt{SAT}$ problem such that $\varphi$ is satisfiable, any computation of $\mathbf{\Pi}(s(\varphi)) + cod(\varphi)$ is an accepting computation.

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

**Corollary 1.** $\mathbf{NP} \cup \mathbf{co-NP} \subseteq \mathbf{PMC}_{\mathcal{DAM}^0(+e_s, mcmp_{out}, -d, +n)}$.

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

## 6 Conclusions

From a computational complexity point of view and assuming that $\mathbf{P} \neq \mathbf{NP}$, dissolution rules play a crucial role in classical polarizationless P systems with active membranes where there is no cooperation, no changing labels neither priorities. In that framework, **PSPACE**-complete problems can be solved in polynomial time when dissolution rules and division for elementary and non-elementary membranes are permitted. However, dissolution rules and division rules for non-elementary membranes can be replaced by minimal cooperation (the left-hand side of the rules has at most two objects) and minimal production (the right-hand side of the rules has at most two objects) in object evolution rules in order to obtain the computational efficiency [11].

In this paper, the ingredient of minimal cooperation and minimal production in object evolution rules is replaced by minimal cooperation and minimal production in send-out communication rules but we have need to use division for non-elementary membranes. The new systems considered are able to efficiently solve computational hard problems even by considering *simple object evolution rules*, that is, these kind of rules only produce one object. An analogous result can be obtained if minimal cooperation and minimal production are considered only for send-in rules, instead of send-out rules ([12]).

The case where only elementary division is allowed, while keeping the restriction that minimal cooperation and minimal production are used in communication rules of the same direction (only *out* or only *in*) remains as future work, as well as the case where division rules are replaced by separation rules.

What about the class $\mathcal{SAM}^0(+e_s, mcmp_{out}, -d, +n)$? That is, what happens if we revisit the framework studied in this paper but replacing division rules by separation rules? We can adapt the reasoning used in the proof of $\mathbf{P} = \mathbf{PMC}_{\mathcal{SAM}^0_{bmc}(-d,-n)}$ (see [10]), and we can prove that by using families of recognizer membrane systems belonging to this class, only problems in class $\mathbf{P}$ can be solved in polynomial time.

## Acknowledgements

This work was partially supported by Grant numbers 61472328 and 61320106005 of the National Natural Science Foundation of China.

## References

1. A. Alhazov, L. Pan. Polarizationless P systems with active membranes. *Grammars*, **7** (2004), 141-159.
2. A. Alhazov, L. Pan, Gh. Păun. Trading polarizations for labels in P systems with active membranes. *Acta Informaticae*, **41**, 2-3 (2004), 111-144.

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, 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. In R. Freund, Gh. Păun, Gr. Rozenberg, A. Salomaa (eds.). *Membrane Computing, 6th International Workshop, WMC 2005, Vienna, Austria, July 18-21, 2005, Revised Selected and Invited Papers, Lecture Notes in Computer Science*, **3850** (2006), 224–240.
6. Gh. Păun. P systems with active membranes: Attacking **NP**–complete problems, *Journal of Automata, Languages and Combinatorics*, **6** (2001), 75–90. A preliminary version in *Centre for Discrete Mathematics and Theoretical Computer Science Research Reports Series*, CDMTCS-102, May 1999.
7. M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini. Complexity classes in models of cellular computing with membranes. *Natural Computing*, **2**, 3 (2003), 265–285.
8. P. Sosík, A. Rodríguez-Patón. Membrane computing and complexity theory: A characterization of PSPACE. *Journal of Computer and System Sciences*, **73** (2007), 137152.
9. L. Valencia-Cabrera, D. Orellana-Martín, M.A. Martínez-del-Amor, A. Riscos-Núñez, M.J. Pérez-Jiménez. Polarizationless P systems with active membranes: Computational complexity aspects. *Journal of Automata, Languages and Combinatorics*, **21**, 1-2 (2016), 107123
10. L. Valencia-Cabrera, D. Orellana-Martín, A. Riscos-Núñez, M.J. Pérez-Jiménez. Minimal cooperation in polarizationless P systems with active membranes. In C. Graciani, Gh. Păun, D. Orellana-Martn, A. Riscos-Nez, L. Valencia-Cabrera (eds.) *Proceedings of the Fourteenth Brainstorming Week on Membrane Computing*, 1-5 February, 2016, Sevilla, Spain, Fénix Editora, pp. 327-356.
11. L. Valencia-Cabrera, D. Orellana-Martín, M.A. Martínez-del-Amor, A. Riscos-Núñez, M.J. Pérez-Jiménez. Reaching efficiency through collaboration in membrane systems: dissolution, polarization and cooperation. *Theoretical Computer Science*, in press, 2017.
12. L. Valencia-Cabrera, D. Orellana-Martín, M.A. Martínez-del-Amor, A. Riscos-Núñez, M.J. Pérez-Jiménez. Restricted polarizationless P systems with active membranes: minimal cooperation only inwards. In this volume, 2017 (manuscript).
13. C. Zandron, C. Ferretti, G. Mauri. Solving **NP**-complete problems using P systems. In I. Antoniou, C.S. Calude, M.J. Dinneen (eds.) *Unconventional Models of Computation, UMC'2K*, Springer, London, 2000, pp. 153-164.