Computational Efficiency of P Systems with Symport/Antiport Rules and Membrane Separation

Luis Valencia-Cabrera[1], Bosheng Song[2], Luis F. Macías-Ramos[1], Linqiang Pan[2], Agustín Riscos-Núñez[1], and Mario J. Pérez-Jiménez[1]

[1] 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@us.es, lfmaciasr@us.es, ariscosn@us.es, marper@us.es

[2] Key Laboratory of Image Information Processing and Intelligent Control,
School of Automation, Huazhong University of Science and Technology,
Wuhan 430074, Hubei, China
E-mail: boshengsong@163.com, lqpan@mail.hust.edu.cn

Summary. Membrane fission is a process by which a biological membrane is split into two new ones in such a way that the contents of the initial membrane is separated and distributed between the new membranes. Inspired by this biological phenomenon, membrane separation rules were considered in membrane computing. In this paper we deal with cell-like P systems with membrane separation rules that use symport/antiport rules (such systems compute by changing the places of objects with respect to the membranes, and not by changing the objects themselves) as communication rules. Specifically we study a lower bound on the length of communication rules with respect to the computational efficiency of such kind of membrane systems; that is, their ability to solve computationally hard problems in polynomial time by trading space for time. The main result of this paper is the following: communication rules involving at most three objects is enough to achieve the computational efficiency of P systems with membrane separation. Thus, a polynomial time solution to SAT problem is provided in this computing framework. It is known that only problems in **P** can be solved in polynomial time by using minimal cooperation in communication rules and membrane separation, so the lower bound of the efficiency obtained is an optimal bound.

1 Introduction

In a eukaryotic cell, the lipid membranes serve as concentration barriers allowing to incorporate material from its environment (in the case of the cell membrane), or exchange material between compartments. This is done by means of a simple three-step process whose last step is *membrane fission*, consisting in splitting it into two new membranes [6].

The biological phenomenon of membrane fission process was incorporated in *membrane computing* [11] as a new kind of computational rules, called *membrane separation rules*, in the framework of polarizationless P systems with active membranes [1]. These rules were associated with different subsets of the working alphabet. In [7], a new definition of separation rules in the framework of P systems with active membranes was introduced, where there exists a distinguished partition of the working alphabet into two subsets such that each separation rule is associated with that predefined partition. By applying such a rule, two new membranes are created, the object triggering it is consumed and the remaining objects are *distributed* among the newly created membranes. A uniform and polynomial time solution to `SAT` problem by a family of P systems with active membranes and membrane separation rules was given in [1].

Networks of membranes, which compute by communication only in the form of symport/antiport rules, were considered in [9]. These networks aim to abstract the biological phenomenon of trans-membrane transport of couples of chemical substances, in the same or opposite directions. Such rules are used both for communication with the environment and for direct communication between different membranes. Membrane fission was introduced into tissue-like P systems with symport/antiport rules through *cell separation* rules yielding *tissue P systems with cell separation* [8]. The computational efficiency of these systems was investigated and a tractability border in terms of the length of communication rules was obtained: passing from 1 to 8 amounts to passing from tractability to **NP**–hardness [8]. Furthermore, in [15], that frontier was refined in an optimal sense with respect to communication rules length (passing from 2 to 3).

Cell-like P systems with symport/antiport rules were introduced in [10], and their computational completeness (five membranes are enough if at most two objects are used in the rules) was shown. In this work, we investigate the computational efficiency of this kind of P systems when membrane separation rules are allowed. Specifically, a polynomial time solution to `SAT` problem by using a family of such systems that use communication rules with length at most 3, is provided. The hardness of the design is high and a P-Lingua simulator [4] has been helpful to check the validity of some modules in which the solution was structured

The paper is organized as follows. Section 2 briefly describes some preliminaries in order to make the paper self-contained. In Section 3, the modeling framework of P systems with symport/antiport rules and membrane separation is introduced. Section 4 describes in detail the design of a family solving `SAT` problem efficiently. The solution presented is informally outlined in Section 5. Then, a formal verification of the solution is exhaustively presented in Section 6. The paper ends with a summary of the results and some conclusions.

2 Preliminaries

2.1 Languages and Multisets

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

A *multiset* over an alphabet Γ is an ordered pair (Γ, f), where f is a mapping from Γ onto the set of natural numbers $\mathbb{N}$. For each $x \in \Gamma$ we say that $f(x)$ is the *multiplicity* of x in that multiset. The *support* of a multiset $m = (\Gamma, f)$ is defined as $supp(m) = \{x \in \Gamma \mid f(x) > 0\}$. A multiset is finite if its support is a finite set. We denote by $\emptyset$ the empty multiset. Let us note that a set is a particular case of a multiset when each symbol of the support has multiplicity 1.

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

2.2 Graphs

Let us recall that a *free tree* (*tree*, for short) is a connected, acyclic, undirected graph. A *rooted tree* is a tree in which one of the vertices (called *the root of the tree*) is distinguished from the others. In a rooted tree the concepts of ascendants and descendants are defined in a usual way. Given a node x (different from the root), if the last edge on the (unique) path from the root of the tree to the node x is $\{x, y\}$ (in this case, $x \neq y$), then y is **the** *parent* of node x and x is **a** *child* of node y. The root is the only node in the tree with no parent (see [2] for details).

2.3 Encoding ordered pairs of natural numbers

The *pair function* $\langle n, m\rangle = ((n + m)(n + m + 1)/2) + n$ is a polynomial–time computable function from $\mathbb{N} \times \mathbb{N}$ onto $\mathbb{N}$ which is also a primitive recursive and bijective function.

3 P systems with symport/antiport rules with membrane separation

In this section we introduce a kind of cell-like P systems that use communication rules capturing the biological phenomenon of trans-membrane transport of chemical substances. Specifically, two processes have been considered. The first one allows a multiset of chemical substances to pass through a membrane in the same direction. In the second one, two multisets of chemical substances (located in different biological membranes) only pass with the help of each other (i.e., an *exchange* of objects between both membranes happens).

Next, we introduce an abstraction of these operations in the framework of P systems with symport/antiport rules following [10]. In these models, the membranes are not polarized.

Definition 1. *A P system with symport/antiport rules and membrane separation (SAS P system, for short) of degree* $q \geq 1$ *is a tuple*

$$\Pi = (\Gamma, \Gamma_0, \Gamma_1, \mathcal{E}, \Sigma, \mu, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}_1, \cdots, \mathcal{R}_q, i_{in}, i_{out}),$$

where

1. Γ *is a finite alphabet;*
2. $\{\Gamma_0, \Gamma_1\}$ *is a partition of* Γ*, that is,* $\Gamma = \Gamma_0 \cup \Gamma_1$*,* $\Gamma_0, \Gamma_1 \neq \emptyset$*,* $\Gamma_0 \cap \Gamma_1 = \emptyset$*;*
3. $\mathcal{E} \subsetneq \Gamma$*;*
4. Σ *is an (input) alphabet strictly contained in* Γ *such that* $\mathcal{E} \subseteq \Gamma \setminus \Sigma$*;*
5. μ *is a rooted tree whose nodes are injectively labelled with* $1, \dots, q$ *(the root of the tree is labelled with* 1*);*
6. $\mathcal{M}_1, \dots, \mathcal{M}_q$ *are finite multisets over* $\Gamma \setminus \Sigma$*;*
7. $\mathcal{R}_i$*,* $1 \leq i \leq q$*, are finite sets of communication rules over* Γ *of the form:*
   (a) Communication rules*:*
      (a) Symport rules*:* (u, out) *or* (u, in)*, where* u *is a finite multiset over* Γ *such that* $|u| > 0$*;*
      (b) Antiport rules*:* $(u, out; v, in)$*, where* u, v *are finite multisets over* Γ *such that* $|u| > 0$ *and* $|v| > 0$*;*
   (b) Separation rules*:* $[\, a\,]_i \to [\, \Gamma_0\,]_i\, [\, \Gamma_1\,]_i$*, where* $a \in \Gamma$*,* $i \in \{2, \dots, q\}$*, with* $i \neq i_{out}$ *the label of a leaf of the tree;*
8. $i_{in} \in \{1, \dots, q\}$ *and* $i_{out} \in \{0, 1, \dots, q\}$*.*

A P system with symport/antiport rules and membrane separation of degree $q \geq 1$

$$\Pi = (\Gamma, \Gamma_0, \Gamma_1, \mathcal{E}, \Sigma, \mu, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}_1, \cdots, \mathcal{R}_q, i_{in}, i_{out})$$

can be viewed as a set of q membranes, labelled with $1, \dots, q$, arranged in a hierarchical structure μ given by a rooted tree whose root is called the *skin membrane*, such that: (a) $\mathcal{M}_1, \dots, \mathcal{M}_q$ represent the finite multisets of *objects* (symbols of the working alphabet Γ) initially placed into the q membranes of the system; (b) $\mathcal{E}$ is the set of objects initially located in the environment of the system (labelled with 0), all of them available in an arbitrary number of copies; (c) $\mathcal{R}_1, \cdots, \mathcal{R}_q$ are finite sets of communication rules over Γ ($\mathcal{R}_i$ is associated with the membrane i of μ); and (d) i_{out} represents a distinguished *region* which will encode the output of the system. We use the term *region* i ($0 \leq i \leq q$) to refer to membrane i in the case

$1 \le i \le q$ and to refer to the environment in the case $i = 0$. The length of rule (u, out) or (u, in) (resp. $(u, out; v, in)$) is defined as $|u|$ (resp. $|u| + |v|$).

For each membrane $i \in \{2, \dots, q\}$ (different from the skin membrane) we denote by $p(i)$ the parent of membrane i in the rooted tree μ. We define $p(1) = 0$, that is, by convention the "parent" of the skin membrane is the environment.

An *instantaneous description* or a *configuration* at an instant t of a SA P system is described by the membrane structure at instant t, all multisets of objects over Γ associated with all the membranes present in the system, and the multiset of objects over $\Gamma - \mathcal{E}$ associated with the environment at that moment. Recall that there are infinite copies of objects from $\mathcal{E}$ in the environment, so that this set is not properly changed along the computation. The *initial configuration* of the system is $(\mu, \mathcal{M}_1, \cdots, \mathcal{M}_q; \emptyset)$.

A symport rule $(u, out) \in \mathcal{R}_i$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if there exists a membrane labelled with i in $\mathcal{C}_t$ such that multiset u is contained in such membrane. When applying a rule $(u, out) \in \mathcal{R}_i$ to such a membrane, the objects specified by u are sent out of that membrane into the region immediately outside (the parent $p(i)$ of i). Note that this can be the environment in the case of the skin membrane.

A symport rule $(u, in) \in \mathcal{R}_i$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if multiset u is contained in the parent of i. When applying a rule $(u, in) \in \mathcal{R}_i$ to a membrane labelled with i, the multiset of objects u leaves the parent of such membrane and enters into the region defined by that membrane.

An antiport rule $(u, out; v, in) \in \mathcal{R}_i$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t if there exists a membrane labelled with i in $\mathcal{C}_t$ such that multiset u is contained in such membrane, and multiset v is contained in the parent of i. When applying a rule $(u, out; v, in) \in \mathcal{R}_i$ to such a membrane, the objects specified by u are sent out of it into the parent of i and, at the same time, the objects specified by v are brought into that membrane i.

A separation rule $[a]_i \to [\Gamma_0]_i[\Gamma_1]_i \in \mathcal{R}_i$ is *applicable* to a configuration $\mathcal{C}_t$ at an instant t, if there exists an elementary membrane labelled with i in $\mathcal{C}_t$, different from the skin membrane, such that it contains object a. When applying a separation rule $[a]_i \to [\Gamma_0]_i\,[\Gamma_1]_i \in \mathcal{R}_i$ to such a membrane in a configuration $\mathcal{C}_t$, triggered by object a, that membrane is separated into two membranes with the same label; at the same time, object a is consumed; the objects (from the original membrane) belonging to Γ_0 are placed in the first membrane, while those from belonging to Γ_1 are placed in the second membrane. This way, several membranes with the same label i can be present in the new membrane structure μ' of the system: for each membrane labelled with $i \neq 1$ we have an arc $(p(i), i)$ in μ' as a result of the application of a membrane separation rule $[a]_i \to [\Gamma_0]_i[\Gamma_1]_i$.

Regarding the semantics of these variants, the rules of such P systems are applied in a non-deterministic maximally parallel manner with the following important remark: when a membrane i is separated, the membrane separation rule is the only one from $\mathcal{R}_i$ which is applied for that membrane at that step. The new membranes resulting from separation could participate in the interaction with other membranes or the environment by means of communication rules at the next step – providing that they are not separated once again. The label of a membrane precisely identify the rules which can be applied to it.

Let Π be a P system with symport/antiport rules and membrane separation. We say that configuration $\mathcal{C}_t$ yields configuration $\mathcal{C}_{t+1}$ in one *transition step*, denoted by $\mathcal{C}_t \Rightarrow_\Pi \mathcal{C}_{t+1}$, if we can pass from $\mathcal{C}_t$ to $\mathcal{C}_{t+1}$ by applying the rules from the system following the above semantics. A *computation* of Π is a (finite or infinite) sequence of configurations such that: (a) the first term is the initial configuration of the system; (b) for each $n \geq 2$, the n-th configuration of the sequence is obtained from the previous configuration in one transition step; and (c) if the sequence is finite (called a *halting computation*) then the last term is a *halting configuration* (a configuration where no rule of the system is applicable). All the computations start from an initial configuration and proceed as stated above; only a halting computation gives a result, which is encoded by the objects present in the output region i_{out} associated with the halting configuration. For each finite multiset w over the input alphabet Σ, a *computation of Π with input multiset w* starts from the configuration of the form $(\mu, \mathcal{M}_1, \ldots, \mathcal{M}_{i_{in}} + w, \ldots, \mathcal{M}_q, \emptyset)$, where the input multiset w is added to the content of the input membrane i_{in}. That is, we have an initial configuration associated with each input multiset w over Σ in recognizer P systems with symport/antiport rules. We denote by $\Pi + w$ the P system Π with input multiset w.

If $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_r)$ of Π is a halting computation, then the *length* of $\mathcal{C}$, denoted by $|\mathcal{C}|$, is r. For each i $(1 \leq i \leq q)$, we denote by $\mathcal{C}_t(i)$ the finite multiset of objects over Γ contained in all membranes labelled with i (by applying separation rules different membranes with the same label can be created) at configuration $\mathcal{C}_t$.

3.1 Recognizer P systems with symport/antiport rules

Recognizer P systems were introduced in [14], and they provide a natural framework to solve decision problems by means of computational devices in membrane computing (i.e., P systems).

Definition 2. *A recognizer P system with symport/antiport rules and membrane separation of degree $q \geq 1$ is a P system with symport/antiport rules and membrane separation of degree q such that:*

1. *The working alphabet has two distinguished symbols* `yes` *and* `no`*;*
2. *initial multisets are finite multisets over $\Gamma \setminus \Sigma$ such that at least one copy of* `yes` *or* `no` *is present in some of them;*
3. *the output region is the environment ($i_{out} = 0$);*
4. *all computations halt;*

5. *if $\mathcal{C}$ is a computation of the system, then either symbol* `yes` *or symbol* `no` *(but not both) must have been released into the environment, and only at the last step of the computation.*

Let us notice that if a recognizer P system has a symport rule of the type $(u, in) \in \mathcal{R}_1$ then the multiset u must contain some object from $\Gamma \setminus \mathcal{E}$ because on the contrary, it might exist non-halting computations of Π.

We say that a computation $\mathcal{C}$ of a recognizer P system is an *accepting computation* (respectively, *rejecting computation*) if object `yes` (respectively, object `no`) appears in the environment associated with the corresponding halting configuration of $\mathcal{C}$, and neither object `yes` nor `no` appears in the environment associated with any non–halting configuration of $\mathcal{C}$.

We denote by $\mathbf{CSC}(k)$ the class of all recognizer P systems with symport/antiport rules and membrane separation (for elementary membranes) such that the length of the communication rules of the system is at most k.

3.2 Polynomial complexity classes of recognizer P systems with symport/antiport rules

Next, according to [13], we define what it means to solve a decision problem by a family of recognizer P systems with symport/antiport rules and membrane separation.

Definition 3. *A decision problem $X = (I_X, \theta_X)$ is solvable in polynomial time by a family $\mathbf{\Pi} = \{\Pi(n) \mid n \in \mathbb{N}\}$ of recognizer P systems with symport/antiport rules and membrane separation or membrane separation, if the following holds:*

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

According to Definition 3, we say that the family $\mathbf{\Pi}$ provides a *uniform solution* to the decision problem X. We also say that ordered pair (cod, s) is a polynomial encoding from X in $\mathbf{\Pi}$ and s is the size mapping associated with that solution. It is worth pointing out that for each instance $u \in I_X$, the P system $\Pi(s(u)) + cod(u)$ is *confluent*, in the sense that all possible computations of the system must give the same answer.

If $\mathbf{R}$ is a class of recognizer P systems, then we denote by $\mathbf{PMC_R}$ the set of all decision problems which can be solved in polynomial time (and in a uniform way) by means of recognizer P systems from $\mathbf{R}$. The class $\mathbf{PMC_R}$ is closed under complement and polynomial–time reductions (see [13] for details). Besides, we have $\mathbf{P} \subseteq \mathbf{PMC_R}$. Indeed, if $X \in \mathbf{P}$ then we consider the family $\mathbf{\Pi} = \{\Pi(n) \mid n \in \mathbb{N}\}$ where $\Pi(n) = \Pi(0)$, for each $n \in \mathbb{N}$, and $\Pi(0)$ is a P system from $\mathbf{R}$ of degree 1 containing only two rules $(\mathtt{yes}, out)$ and $(\mathtt{no}, out)$. Let us consider the polynomial encoding from X in $\mathbf{\Pi}$ defined as follows: (a) $s(u) = 0$, for each $u \in I_X$; and (b) $cod(u) = \mathtt{yes}$ if $\theta_X(u) = 1$ and $cod(u) = \mathtt{no}$ if $\theta_X(u) = 0$. Then, the family $\mathbf{\Pi}$ solves X according to Definition 3.

4 On Efficiency of CSC(3)

The limitations on the efficiency of P systems with membrane separation whose symport/antiport rules involve at most two objects, have been established [5]. Specifically, it has been proved that the polynomial complexity class $\mathbf{PMC_{CSC(2)}}$ is equal to class $\mathbf{P}$: only tractable problems can be efficiently solved by using families of P systems with membrane separation which make use of symport/antiport rules with length at most 2. In this Section we analyze the computational efficiency of familes of P systems from $\mathbf{CSC(3)}$, and it is given a polynomial time solution to `SAT` problem by means of a family of such P systems, in a uniform way, according to Definition 3.

4.1 A polynomial time solution to `SAT` problem in CSC(3)

Let us recall that `SAT` problem is the following: *given a Boolean formula in conjunctive normal form (CNF), to determine whether or not there exists an assignment to its variables on which it evaluates true.* This is a well known **NP**-complete problem [3].

We consider a family $\mathbf{\Pi} = \{\Pi(t) \mid t \in \mathbb{N}\}$ of recognizer P system from $\mathbf{CSC}(3)$, such that each system $\Pi(t)$, with $t = \langle n, m\rangle$, will process all instances of `SAT` problem (an instance is a Boolean formula φ in conjunctive normal form with n variables and m clauses) provided that the appropriate input multiset $cod(\varphi)$ is supplied to the system.

For each $n, m \in \mathbb{N}$, we consider the recognizer P system from $\mathbf{CSC}(3)$

$$\Pi(\langle n,m\rangle) = (\Gamma, \Gamma_0, \Gamma_1, \mathcal{E}, \Sigma, \mu, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}_1, \cdots, \mathcal{R}_q, i_{in}, i_{out})$$

defined as follows:

(1) Working alphabet:

$$\begin{aligned}\Gamma = \Sigma \cup \mathcal{E} \cup \{&\alpha_{i,0,k}, \alpha'_{i,0,k} \mid 1 \le i \le n-1 \wedge 0 \le k \le 1\} \cup \\ \{&A_1, B_1, b_1, b'_1, c_1, c'_1, v_1, q_{1,1}, \beta_0, \beta'_0, \beta''_0, \gamma_0, \gamma'_0, \gamma''_0, \gamma'''_0, f_0, \mathtt{yes}, \mathtt{no}\} \cup \\ \{&f'_i \mid 0 \le i \le 3n+2m+1\}, \cup \{\rho_{i,0}, \tau_{i,0} \mid 1 \le i \le n\}, \cup \{\delta_{j,0} \mid 0 \le j \le m\}\end{aligned}$$

where the input alphabet is $\Sigma = \{x_{i,j}, \overline{x}_{i,j} \mid 1 \le i \le n \wedge 1 \le j \le m\}$, and the alphabet of the environment is:

$$\begin{aligned}\mathcal{E} = \{&\alpha_{i,j,k}, \alpha'_{i,j,k} \mid 1 \le i \le n-1 \wedge 1 \le j \le 3(n-1) \wedge 0 \le k \le 1\} \cup \\ \{&\beta_j, \beta'_j, \beta''_j, \gamma_j, \gamma'_j, \gamma''_j, \gamma'''_j \mid 1 \le j \le 3(n-1)\} \cup \\ \{&\rho_{i,j}, \tau_{i,j} \mid 1 \le i \le n \wedge 1 \le j \le 3n-1\} \cup \\ \{&T_{i,j}, T'_{i,j}, F_{i,j}, F'_{i,j} \mid 1 \le i < j \wedge 1 \le j \le n\} \cup \\ \{&T_{i,i}, F'_{i,i}, T_i, F_i \mid 1 \le i \le n\} \cup \{A_i, A'_i, B_i, B'_i \mid 2 \le i \le n+1\} \cup \\ \{&b_i, b'_i, c_i, c'_i \mid 2 \le i \le n\} \cup \{v_i \mid 2 \le i \le n-1\} \cup \\ \{&y_i, a_i, w_i \mid 1 \le i \le n-1\} \cup \{z_i \mid 1 \le i \le n-2\} \cup \\ \{&q_{i,j} \mid 1 \le i \le j \wedge 2 \le j \le n-1\} \cup \{u_{i,j} \mid 1 \le i \le j \wedge 1 \le j \le n-2\} \cup \\ \{&t_{i,j}, f_{i,j}, r_{i,j}, s_{i,j} \mid 1 \le i \le j \wedge 1 \le j \le n-1\} \cup \\ \{&d_{i,j,k}, \overline{d}_{i,j,k} \mid 1 \le i \le n \wedge 1 \le j \le m \wedge 1 \le k \le n-1\} \cup \\ \{&f_r \mid 1 \le r \le 3n+2m\} \cup \{e_{i,j}, \overline{e}_{i,j} \mid 1 \le i \le n \wedge 1 \le j \le m\} \cup \\ \{&\delta_{j,r} \mid 0 \le j \le m \wedge 1 \le r \le 3n\} \cup \{E_j \mid 0 \le j \le m\} \cup \{S\}\end{aligned}$$

(2) The partition is $\{\Gamma_0, \Gamma_1\}$, where $\Gamma_0 = \Gamma \setminus \Gamma_1$ and

$$\begin{aligned}\Gamma_1 = \{&T'_{i,j}\, F'_{i,j} \mid 1 \le i < j \wedge 1 \le j \le n\} \cup \{F'_{i,i} \mid 1 \le i \le n\} \cup \\ \{&A'_i, B'_i \mid 2 \le i \le n+1\}\end{aligned}$$

(3) Membrane structure: $\mu = [\,[\ \ ]_2\,[\ \ ]_3]_1$. The input membrane is the membrane labelled with 1.

(4) Initial multisets:

$$\begin{aligned}\mathcal{M}_1 &= \{\alpha_{i,0,k}, \alpha'_{i,0,k} \mid 1 \le i \le n-1 \wedge 0 \le k \le 1\} \cup \{\rho_{i,0}, \tau_{i,0} \mid 1 \le i \le n\} \cup \\ &\quad\ \{\beta_0, \beta'_0, \beta''_0, \gamma_0, \gamma'_0, \gamma''_0, \gamma'''_0, c_1, c'_1, b_1, b'_1, v_1, q_{1,1}, f_0, \mathtt{yes}\} \cup \\ &\quad\ \{\delta_{j,0} \mid 0 \le j \le m\} \cup \{f'_p \mid 1 \le p \le 3n+2m+1\} \\ \mathcal{M}_2 &= \{A_1, B_1\} \\ \mathcal{M}_3 &= \{f'_0, \mathtt{no}\}\end{aligned}$$

(5) $\boxed{\text{Rules in } \mathcal{R}_1}$:

1.1 Rules to generate in the membrane 1 of configuration $\mathcal{C}_{3p+1}$ ($p = 1, \dots, n-1$) the objects $T^{2^{p-1}}_{i,p+1}, T'^{2^{p-1}}_{i,p+1}, F^{2^{p-1}}_{i,p+1}, F'^{2^{p-1}}_{i,p+1}$:

$$\left.\begin{array}{l}(\alpha_{i,0,k}, out; \alpha_{i,1,k}, in)\\(\alpha'_{i,0,k}, out; \alpha'_{i,1,k}, in)\\(\alpha_{i,1,k}, out; \alpha_{i,2,k}, in)\\(\alpha'_{i,1,k}, out; \alpha'_{i,2,k}, in)\\(\alpha_{i,2,k}, out; \alpha_{i,3,k}, in)\\(\alpha'_{i,2,k}, out; \alpha'_{i,3,k}, in)\end{array}\right\} 1 \le i \le n-1 \wedge 0 \le k \le 1$$

$$\begin{array}{l}(\alpha_{i,3p,k}\,, out; \alpha_{i,3p+1,k}\ \varDelta^{k}_{i,p+1}\,, in) : 1 \le i \le p \wedge 1 \le p \le n-2 \wedge 0 \le k \le 1\\(\alpha'_{i,3p,k}\,, out; \alpha'_{i,3p+1,k}\ \varDelta'^{k}_{i,p+1}\,, in) : 1 \le i \le p \wedge 1 \le p \le n-2 \wedge 0 \le k \le 1\\(\alpha_{i,3p,k}\,, out; \alpha_{i,3p+1,k}\,, in) : p+1 \le i \le n-1 \wedge 1 \le p \le n-2 \wedge 0 \le k \le 1\\(\alpha'_{i,3p,k}\,, out; \alpha'_{i,3p+1,k}\,, in) : p+1 \le i \le n-1 \wedge 1 \le p \le n-2 \wedge 0 \le k \le 1\\(\alpha_{i,3p+1,k}\,, out; \alpha_{i,3p+2,k}\,, in) : 1 \le i \le n-1 \wedge 1 \le p \le n-2 \wedge 0 \le k \le 1\\(\alpha'_{i,3p+1,k}\,, out; \alpha'_{i,3p+2,k}\,, in) : 1 \le i \le n-1 \wedge 1 \le p \le n-2 \wedge 0 \le k \le 1\\(\alpha_{i,3p+2,k}\,, out; \alpha^{2}_{i,3p+3,k}\,, in) : 1 \le i \le n-1 \wedge 1 \le p \le n-2 \wedge 0 \le k \le 1\\(\alpha'_{i,3p+2,k}\,, out; \alpha'^{2}_{i,3p+3,k}\,, in) : 1 \le i \le n-1 \wedge 1 \le p \le n-2 \wedge 0 \le k \le 1\\(\alpha_{i,3(n-1),k}\,, out;\ \varDelta^{k}_{i,n}\,, in) : 1 \le i \le n-1 \wedge 0 \le k \le 1\\(\alpha'_{i,3(n-1),k}\,, out;\ \varDelta'^{k}_{i,n}\,, in) : 1 \le i \le n-1 \wedge 0 \le k \le 1\end{array}$$

where $\varDelta^0_{i,j} = F_{i,j}$, $\varDelta'^0_{i,j} = F'_{i,j}$, $\varDelta^1_{i,j} = T_{i,j}$, $\varDelta'^1_{i,j} = T'_{i,j}$.

1.2 Rules to generate in the membrane 1 of configuration $\mathcal{C}_{3p+1}$ ($p = 0, 1, \ldots, n-1$) the objects $B^{2^p}_{p+2}, B'^{2^p}_{p+2}, S^{2^p}$:

$$\left.\begin{array}{r}(\beta_{3p}\,, out; \beta_{3p+1}\ B_{p+2}\,, in)\\(\beta'_{3p}\,, out; \beta'_{3p+1}\ B'_{p+2}\,, in)\\(\beta''_{3p}\,, out; \beta''_{3p+1}\ S\,, in)\\(\beta_{3p+1}\,, out; \beta_{3p+2}\,, in)\\(\beta'_{3p+1}\,, out; \beta'_{3p+2}\,, in)\\(\beta''_{3p+1}\,, out; \beta''_{3p+2}\,, in)\\(\beta_{3p+2}\,, out; \beta^{2}_{3p+3}\,, in)\\(\beta'_{3p+2}\,, out; \beta'^{2}_{3p+3}\,, in)\\(\beta''_{3p+2}\,, out; \beta''^{2}_{3p+3}\,, in)\end{array}\right\} 0 \le p \le n-3$$

$$\left.\begin{array}{r}(\beta_{3(n-2)}\,, out; \beta_{3(n-2)+1}\ B_{n}\,, in)\\(\beta'_{3(n-2)}\,, out; \beta'_{3(n-2)+1}\ B'_{n}\,, in)\\(\beta''_{3(n-2)}\,, out; \beta''_{3(n-2)+1}\ S\,, in)\\(\beta_{3(n-2)+1}\,, out; \beta_{3(n-2)+2}\,, in)\\(\beta'_{3(n-2)+1}\,, out; \beta'_{3(n-2)+2}\,, in)\\(\beta''_{3(n-2)+1}\,, out; \beta''_{3(n-2)+2}\,, in)\\(\beta_{3(n-2)+2}\,, out; \beta^{2}_{3(n-2)+3}\,, in)\\(\beta'_{3(n-2)+2}\,, out; \beta'^{2}_{3(n-2)+3}\,, in)\\(\beta''_{3(n-2)+2}\,, out; \beta''^{2}_{3(n-2)+3}\,, in)\end{array}\right\}$$

$$\left.\begin{array}{r}(\beta_{3(n-1)}\,, out; B_{n+1}\,, in)\\(\beta'_{3(n-1)}\,, out; B'_{n+1}\,, in)\\(\beta''_{3(n-1)}\,, out; S\,, in)\end{array}\right\}$$

1.3 Rules to generate in the membrane 1 of configuration $\mathcal{C}_{3p+1}$ ($p = 0, 1, \ldots, n-1$) the objects $T^{2^p}_{p+1,p+1}, T'^{2^p}_{p+1,p+1}, A^{2^p}_{p+2}, A'^{2^p}_{p+2}$:

$$\left.\begin{array}{r}
(\gamma_{3p}\,, out; \gamma_{3p+1}\ T_{p+1,p+1}\,, in) \\
(\gamma'_{3p}\,, out; \gamma'_{3p+1}\ F'_{p+1,p+1}\,, in) \\
(\gamma''_{3p}\,, out; \gamma''_{3p+1}\ A_{p+2}\,, in) \\
(\gamma'''_{3p}\,, out; \gamma'''_{3p+1}\ A'_{p+2}\,, in) \\
(\gamma_{3p+1}\,, out; \gamma_{3p+2}\,, in) \\
(\gamma'_{3p+1}\,, out; \gamma'_{3p+2}\,, in) \\
(\gamma''_{3p+1}\,, out; \gamma''_{3p+2}\,, in) \\
(\gamma'''_{3p+1}\,, out; \gamma'''_{3p+2}\,, in) \\
(\gamma_{3p+2}\,, out; \gamma^2_{3p+3}\,, in) \\
(\gamma'_{3p+2}\,, out; \gamma'^2_{3p+3}\,, in) \\
(\gamma''_{3p+2}\,, out; \gamma''^2_{3p+3}\,, in) \\
(\gamma'''_{3p+2}\,, out; \gamma'''^2_{3p+3}\,, in)
\end{array}\right\} 0 \le p \le n-3$$

$$\left.\begin{array}{r}
(\gamma_{3(n-2)}\,, out; \gamma_{3(n-2)+1}\ T_{n-1,n-1}\,, in) \\
(\gamma'_{3(n-2)}\,, out; \gamma'_{3(n-2)+1}\ F'_{n-1,n-1}\,, in) \\
(\gamma''_{3(n-2)}\,, out; \gamma''_{3(n-2)+1}\ A_{n}\,, in) \\
(\gamma'''_{3(n-2)}\,, out; \gamma''_{3(n-2)+1}\ A'_{n}\,, in) \\
(\gamma_{3(n-2)+1}\,, out; \gamma_{3(n-2)+2}\ in) \\
(\gamma'_{3(n-2)+1}\,, out; \gamma'_{3(n-2)+2}\ in) \\
(\gamma''_{3(n-2)+1}\,, out; \gamma''_{3(n-2)+2}\ in) \\
(\gamma'''_{3(n-2)+1}\,, out; \gamma'''_{3(n-2)+2}\ in) \\
(\gamma_{3(n-2)+2}\,, out; \gamma^2_{3(n-2)+3}\ in) \\
(\gamma'_{3(n-2)+2}\,, out; \gamma'^2_{3(n-2)+3}\ in) \\
(\gamma''_{3(n-2)+2}\,, out; \gamma''^2_{3(n-2)+3}\ in) \\
(\gamma'''_{3(n-2)+2}\,, out; \gamma'''^2_{3(n-2)+3}\ in)
\end{array}\right\}$$

$$\left.\begin{array}{r}
(\gamma_{3(n-1)}\,, out;\ T_{n,n}\,, in) \\
(\gamma'_{3(n-1)}\,, out;\ F'_{n,n}\,, in) \\
(\gamma''_{3(n-1)}\,, out;\ A_{n+1}\,, in) \\
(\gamma'''_{3(n-1)}\,, out;\ A'_{n+1}\,, in)
\end{array}\right\}$$

1.4 Rules to generate in the membrane 1 of configuration $\mathcal{C}_{3n}$ the objects $T^{2^{n-1}}_i, F^{2^{n-1}}_i$ ($1 \le i \le n$):

$$\left.\begin{array}{r}
(\rho_{i,0}\,, out; \rho_{i,1}\,, in) \\
(\tau_{i,0}\,, out; \tau_{i,1}\,, in) \\
(\rho_{i,1}\,, out; \rho_{i,2}\,, in) \\
(\tau_{i,1}\,, out; \tau_{i,2}\,, in) \\
(\rho_{i,2}\,, out; \rho_{i,3}\,, in) \\
(\tau_{i,2}\,, out; \tau_{i,3}\,, in)
\end{array}\right\} 1 \le i \le n$$

$$\left.\begin{array}{r}(\rho_{i,3p}\,, out; \rho_{i,3p+1}\,, in)\\(\tau_{i,3p}\,, out; \tau_{i,3p+1}\,, in)\\(\rho_{i,3p+1}\,, out; \rho^2_{i,3p+2}\,, in)\\(\tau_{i,3p+1}\,, out; \tau^2_{i,3p+2}\,, in)\\(\rho_{i,3p+2}\,, out; \rho_{i,3p+3}\,, in)\\(\tau_{i,3p+2}\,, out; \tau_{i,3p+3}\,, in)\end{array}\right\} 1 \le i \le n \wedge 1 \le p \le n-2$$

$$\left.\begin{array}{r}(\rho_{i,3(n-1)}\,, out; \rho_{i,3(n-1)+1}\,, in)\\(\tau_{i,3(n-1)}\,, out; \tau_{i,3(n-1)+1}\,, in)\\(\rho_{i,3(n-1)+1}\,, out; \rho^2_{i,3(n-1)+2}\,, in)\\(\tau_{i,3(n-1)+1}\,, out; \tau^2_{i,3(n-1)+2}\,, in)\\(\rho_{i,3(n-1)+2}\,, out; T_i\,, in)\\(\tau_{i,3(n-1)+2}\,, out; F_i\,, in)\end{array}\right\} 1 \le i \le n$$

$$\left.\begin{array}{r}(A_i\,, out; a_i\,, in)\\(A'_i\,, out; a_i\,, in)\\(B_i\,, out; a_i\,, in)\\(B'_i\,, out; a_i\,, in)\end{array}\right\} 1 \le i \le n-1$$

$$\left.\begin{array}{r}(y_i\,, out; z_i\, w_i\,, in) : 1 \le i \le n-2\\(y_{n-1}\,, out; w_{n-1}\,, in) : \qquad\qquad\quad\end{array}\right\}$$

$$\left.\begin{array}{r}(w_i\,, out; c_{i+1}\, c'_{i+1}\,, in) : 1 \le i \le n-1\\(z_i\,, out; v_{i+1}\,, in) : 1 \le i \le n-2\end{array}\right\}$$

$$\left.\begin{array}{r}(v_i\,, out; y^2_i\,, in)\\(a_i\,, out; b_{i+1}\, b'_{i+1}\,, in)\end{array}\right\} 1 \le i \le n-1$$

$$\left.\begin{array}{l}(q_{1,1}\,, out; r_{1,1}\,, in)\\\;(q_{i,j}\,, out; r^2_{i,j}\,, in) : 1 \le i \le n-1 \wedge i \le j \le n-1\end{array}\right\}$$

$$\left.\begin{array}{r}(r_{i,j}\,, out; s_{i,j}\, u_{i,j}\,, in) : 1 \le i \le n-2 \wedge i \le j \le n-2\\(r_{i,n-1}\,, out; s_{i,n-1}\,, in) : \qquad\quad 1 \le i \le n-1 \qquad\end{array}\right\}$$

$$(s_{i,j}\,, out; t_{i,j}\, f_{i,j}\,, in) \;:\; 1 \le i \le n-1 \wedge i \le j \le n-1$$

$$\left.\begin{array}{r}(u_{1,j}\,, out; q_{1,j+1}\, q_{2,j+1}\,, in) : \qquad 1 \le j \le n-2 \qquad\quad\\(u_{i,j}\,, out; q_{i+1,j+1}\,, in) : 2 \le i \le j \wedge 2 \le j \le n-2\end{array}\right\}$$

$$\left.\begin{array}{r}(T_{i,j}\, t_{i,j}, out)\\(T'_{i,j}\, t_{i,j}, out)\\(F_{i,j}\, f_{i,j}, out)\\(F'_{i,j}\, f_{i,j}, out)\end{array}\right\} 1 \le i \le j \wedge 1 \le j \le n$$

1.5 Rules allowing that each object $x_{i,j}$ (meaning that $x_i \in C_j$) and $\overline{x}_{i,j}$ (meaning that $\neg x_i \in C_j$) results in the corresponding $e_{i,j}$ and $\overline{e}_{i,j}$ objects with multiplicity 2^{n-1} in membrane 1 of configuration $\mathcal{C}_{n+1}$.

$$\left.\begin{array}{r}(x_{i,j}\,, out; d^2_{i,j,1}\,; in)\\(\overline{x}_{i,j}\,, out; \overline{d}^2_{i,j,1}\,; in)\end{array}\right\} 1 \le i \le n \wedge 1 \le j \le m$$

$$\left.\begin{array}{r}(d_{i,j,k}\,, out; d^2_{i,j,k+1}\,, in)\\(\overline{d}_{i,j,k}\,, out; \overline{d}^2_{i,j,k+1}\,, in)\end{array}\right\} 1 \le i \le n \wedge 1 \le j \le m \wedge 1 \le k \le n-2$$

$$\left.\begin{array}{r}(d_{i,j,n-1}\,, out; e_{i,j}\,, in)\\(\overline{d}_{i,j,n-1}\,, out; \overline{e}_{i,j}\,, in)\end{array}\right\} 1 \le i \le n \wedge 1 \le j \le m$$

1.6 Output rule with **affirmative** answer: $(E_0\, f_{3n+2m}\, \mathtt{yes}\,; out)$.

1.7 Output rule with **negative** answer: $(f_{3n+2m}\, \mathtt{no}\,; out)$.

1.8 Rules to generate in the membrane 1 of configuration $\mathcal{C}_{3n}$ the objects $E_1^{2^n}$, and in the membrane 1 of configuration $\mathcal{C}_{3n+1}$ the objects $E_0^{2^n}, E_2^{2^n}, \ldots, E_m^{2^n}$:

$$\left.\begin{array}{r}(\delta_{j,3p}\,, out; \delta_{j,3p+1}, in)\\(\delta_{j,3p+1}\,, out; \delta^2_{j,3p+2}, in)\end{array}\right\} 0 \le j \le m \wedge 0 \le p \le n-1$$

$$(\delta_{j,3p+2}\,, out; \delta_{j,3p+3}, in)\ 0 \le j \le m \wedge 0 \le p \le n-2$$

$$(\delta_{1,3(n-1)+2}\,, out; E_1, in)$$

$$\left.\begin{array}{r}(\delta_{j,3(n-1)+2}\,, out; \delta_{j,3(n-1)+3}, in)\\(\delta_{j,3n}\,, out; E_j, in)\end{array}\right\} 0 \le j \le m \wedge j \ne 1$$

$$(f_p\,, out; f_{p+1}\,; in)\ 0 \le p \le 3n+2m-1$$

1.9 Rules to remove a part of the garbage:

$$\left.\begin{array}{r}(t_{i,k}\ T_{i,k}\,, out)\\(t_{i,k}\ T'_{i,k}\,, out)\\(f_{i,k}\ F_{i,k}\,, out)\\(f_{i,k}\ F'_{i,k}\,, out)\end{array}\right\} 1 \le i < k \wedge 2 \le k \le n$$

$$\left.\begin{array}{r}(t_{i,i}\ T_{i,i}\,, out)\\(f_{i,i}\ F'_{i,i}\,, out)\end{array}\right\} 1 \le i \le n$$

$$\left.\begin{array}{r}(b_k\ B_{k+1}\,, out)\\(b'_k\ B'_{k+1}\,, out)\\(c_k\ A_{k+1}\,, out)\\(c'_k\ A'_{k+1}\,, out)\end{array}\right\} n-1 \le k \le n$$

(6) $\boxed{\text{Rules in } \mathcal{R}_2}$:

2.1 Separation rule: $[\, S\,]_2 \rightarrow [\, \Gamma_0\,]_2\ [\, \Gamma_1\,]_2$.

2.2 Rules to produce objects $T_{i,i}, A_{i+1}, F'_{i,i}, A'_{i+1}$ in each membrane 2:

$$\left.\begin{array}{r} (A_i, out; c_i\, c'_i, in) \\ (A'_i, out; c_i\, c'_i, in) \\ (B_i, out; b_i\, b'_i, in) \\ (B'_i, out; b_i\, b'_i, in) \\ (b_i, out; B_{i+1}\, S, in) \\ (b'_i, out; B'_{i+1}, in) \\ (c_i, out; T_{i,i}\, A_{i+1}, in) \\ (c'_i, out; F'_{i,i}\, A'_{i+1}, in) \end{array}\right\} 1 \leq i \leq n$$

2.3 Rules to produce an object E_1 in each membrane 2 of configuration $\mathcal{C}_{3n+1}$ and an object E_0 in each membrane 2 of configuration $\mathcal{C}_{3n+2}$:

$$\begin{array}{c} (B_{n+1}, out; E_1, in) \\ (B'_{n+1}, out; E_1, in) \\ (A_{n+1}, out; E_0, in) \\ (A'_{n+1}, out; E_0, in) \end{array}$$

2.4 Rules to produce a truth assignment in each membrane 2 of configuration $\mathcal{C}_{3n+1}$:

$$\left.\begin{array}{r} (T_{i,j}, out; t_{i,j}, in) \\ (T'_{i,j}, out; t_{i,j}, in) \\ (F_{i,j}, out; f_{i,j}, in) \\ (F'_{i,j}, out; f_{i,j}, in) \end{array}\right\} 1 \leq i \leq j \wedge 1 \leq j \leq n$$

$$\left.\begin{array}{r} (t_{i,j}, out; T_{i,j+1}\, T'_{i,j+1}, in) \\ (f_{i,j}, out; F_{i,j+1}\, F'_{i,j+1}, in) \end{array}\right\} 1 \leq i \leq j \wedge 1 \leq j \leq n-1$$

$$\left.\begin{array}{r} (T_{i,n}, out; T_i, in) \\ (T'_{i,n}, out; T_i, in) \\ (F_{i,n}, out; F_i, in) \\ (F'_{i,n}, out; F_i, in) \end{array}\right\} 1 \leq i \leq n$$

2.5 Rules to check clause C_j through the truth assignment encoded by a membrane 2:

$$\left.\begin{array}{r} (E_j\, T_i, out; e_{i,j}, in) \\ (E_j\, F_i, out; \overline{e}_{i,j}, in) \end{array}\right\} 1 \leq i \leq n \wedge 1 \leq j \leq m$$

2.6 Rules to restore the truth assignment encoded by a membrane 2 which makes clause C_j true:

$$\left.\begin{array}{r} (e_{i,j}, out, E_{j+1}\, T_i, in) \\ (\overline{e}_{i,j}, out, E_{j+1}\, F_i, in) \end{array}\right\} 1 \leq i \leq n \wedge 1 \leq j \leq m-1$$

2.7 Rules to send an object E_0 to membrane 1 of configuration $\mathcal{C}_{3n+2m+1}$, meaning that some truth assignment encoded by a membrane labelled with 2 makes the input formula φ true:

$$\left.\begin{array}{l}(e_{i,m}\, E_0\,;out)\\(\overline{e}_{i,m}\, E_0\,;out)\end{array}\right\} 1 \leq i \leq n$$

(7) $\boxed{\text{Rules in } \mathcal{R}_3}$:

3.1 Rules to produce objects $f'_{3n+2m+1}$ and no in the membrane 1 of configuration $\mathcal{C}_{3n+2m+2}$.

$$(f'_p, out; f'_{p+1}, in)\ 0 \leq p \leq 3n+2m$$
$$(f'_{3n+2m+1}\, \texttt{no}\,; out)$$

5 An overview of the computations

A family of recognizer P systems with symport/antiport rules and membrane separation is constructed above. For an instance of SAT problem $\varphi = C_1 \wedge \cdots \wedge C_m$, consisting of m clauses $C_j = l_{j,1} \vee \cdots \vee l_{j,r_j},\ 1 \leq j \leq m$, where $Var(\varphi) = \{x_1, \cdots, x_n\}$, and $l_{j,k} \in \{x_i, \neg x_i \mid 1 \leq i \leq n\},\ 1 \leq j \leq m, 1 \leq k \leq r_j$. Let us assume that the number of variables, n, and the number of clauses, m, of the input formula φ, are greater or equal to 2.

The *size* mapping on the set of instances is defined as $s(\varphi) = \langle m, n \rangle$, for each $\varphi \in I_{\texttt{SAT}}$, and the encoding of the instance φ is the multiset

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

That is, $x_{i,j}$ (respectively, $\overline{x}_{i,j}$) denotes variable x_i (respectively, $\neg x_i$) belonging to clause C_j. Then, the Boolean formula φ will be processed by the system $\Pi(s(\varphi))$ with input multiset $cod(\varphi)$.

Next, we informally describe how the system $\Pi(s(\varphi)) + cod(\varphi)$ works, in order to process the instance φ of SAT problem. The solution proposed follows a brute force algorithm in the framework of recognizer P systems with symport/antiport rules and membrane separation, and it consists of the following phases:

- *Generation phase*: using separation rules, all truth assignments for the variables associated with the Boolean formula $\varphi(x_1, \ldots, x_n)$ are produced. This phase exactly takes $3n + 1$ computation steps.
- *Checking phase*: checking whether or not the input formula φ is satisfied by some truth assignment generated in the previous phase. This phase takes, exactly, $3m + 1$ steps, being m the number of clauses of the formula φ.

- *Output phase*: the system sends the right answer to the environment depending on the results of the previous phase. This phase takes, exactly, 1 step if the answer affirmative, and 2 steps if the answer is negative.

Generation phase

In this phase, all truth assignments for the variables associated with the Boolean formula $\varphi(x_1, \ldots, x_n)$ are generated, by applying separation rules in membranes labelled with 2. This way, after completing the phase, there will exist 2^n membranes labelled with 2 such that each of them encodes a different truth assignment of the variables $\{x_1, \ldots, x_n\}$.

This phase consists in a loop with n iterations and one additional final step. Each iteration of the loop takes three steps and, consequently, this phase takes $3n+1$ steps.

To do this, in the configurations of the kind $\mathcal{C}_{3p+2}$ $(0 \leq p \leq n-1)$ there exist 2^p membranes labelled with 2 containing objects

$$A_{p+2},\ A'_{p+2},\ B_{p+2},\ B'_{p+2},\ T_{p+1,p+1},\ F'_{p+1,p+1},\ S$$

along with $2p$–tuples of objects $(\pi_{1,p+1}, \pi'_{1,p+1}, \ldots, \pi_{p,p+1}, \pi'_{p,p+1})$, with $\pi \in \{T, F\}$, in such a way that the corresponding tuples are all different in the different membranes.

Thus, a separation rule can be applied to each membrane labelled with 2. As a consequence, in configuration $\mathcal{C}_{3p+3}$ $(0 \leq p \leq n-2)$ there will exist 2^{p+1} membranes labelled with 2. 2^p of them will contain objects A_{p+2} and B_{p+2}, as well as $(p+1)$–tuples $(\pi_{1,p+1}, \ldots, \pi_{p+1,p+1})$, with $\pi \in \{T, F\}$, in such a way that $\pi_{p+1,p+1} = T_{p+1,p+1}$, and the corresponding tuples of these membranes are all different. The other 2^p membranes labelled with 2 contain the objects A'_{p+2} and B'_{p+2}, as well as $(p+1)$–tuples $(\pi'_{1,p+1}, \ldots, \pi'_{p+1,p+1})$ with $\pi \in \{T, F\}$, in such a way that $\pi'_{p+1,p+1} = F'_{p+1,p+1}$ and the corresponding tuples of these membranes are all different.

Finally, in configuration $\mathcal{C}_{3n}$ there exist 2^n membranes labelled with 2. 2^{n-1} of them contain the objects A_{n+1} and B_{n+1}, as well as n–tuples $(\pi_{1,n}, \ldots, \pi_{n,n})$ with $\pi \in \{T, F\}$, in such a way that $\pi_{n,n} = T_{n,n}$ and the corresponding tuples of these membranes are all different. The other 2^{n-1} membranes labelled with 2 contain the objects A'_{n+1} and B'_{n+1}, as well as n–tuples $(\pi'_{1,n}, \ldots, \pi'_{n,n})$ with $\pi \in \{T, F\}$, in such a way that $\pi'_{n,n} = F'_{n,n}$ and the corresponding tuples of these membranes are all different.

This phase ends in the step $3n+1$, where configuration $\mathcal{C}_{3n+1}$ contains 2^n membranes labelled with 2, each one of them containing the objects A_{n+1} and E_1, as well as n–tuples $(\pi_1, \ldots, \pi_n)$ with $\pi \in \{T, F\}$, and the corresponding tuples of these membranes are all different.

Simultaneously, during the generation phase, from the input multiset placed initially in the skin membrane, 2^{n-1} copies of each object of that multiset are generated in that membrane, corresponding to configuration $\mathcal{C}_n$. Due to technical reasons, we will change variables $x_{i,j}$ and $\overline{x}_{i,j}$ by $e_{i,j}$ and $\overline{e}_{i,j}$, respectively. This is accomplished by using the following rules from $\mathcal{R}_1$:

$$\left.\begin{array}{r}
(x_{i,j}\,, out; d^2_{i,j,1}\,; in) \;:\; 1 \le i \le n \wedge 1 \le j \le m \\
(\overline{x}_{i,j}\,, out; \overline{d}^2_{i,j,1}\,; in) \;:\; 1 \le i \le n \wedge 1 \le j \le m \\
(d_{i,j,k}\,, out; d^2_{i,j,k+1}\,, in) \;:\; 1 \le i \le n \wedge 1 \le j \le m \wedge 1 \le k \le n-2 \\
(\overline{d}_{i,j,k}\,, out; \overline{d}^2_{i,j,k+1}\,, in) \;:\; 1 \le i \le n \wedge 1 \le j \le m \wedge 1 \le k \le n-2 \\
(d_{i,j,n-1}\,, out; e_{i,j}\,, in) \;:\; 1 \le i \le n \wedge 1 \le j \le m \\
(\overline{d}_{i,j,n-1}\,, out; \overline{e}_{i,j}\,, in) \;:\; 1 \le i \le n \wedge 1 \le j \le m
\end{array}\right\}$$

The cited multiset that codifies the input formula will be denoted by $(cod(\varphi))_e^{2^{n-1}}$.

Checking phase

This phase begins at computation step $3n+2$ and consists in a loop with m iterations, taking each of them 2 steps. Hence, the checking phase takes $2m$ steps.

In the configuration $\mathcal{C}_{3n+1}$, the presence of an object E_1 in each membrane labelled with 2, along with the code of a truth assignment, marks the beginning of this phase. In the first iteration of the loop, the truth assignments making clause C_1 of φ true are found. To do this, the following rules of $\mathcal{R}_2$ are applied:

$$\left.\begin{array}{r}
(E_1\, T_i, out; e_{i,1}, in) \\
(E_1\, F_i, out; \overline{e}_{i,1}, in)
\end{array}\right\} 1 \le i \le n$$

Simultaneously, in the computation step $(3n+1)+2$, the object E_0 is incorporated to each of the membranes labelled with 2 by means of the application of the following rules of $\mathcal{R}_2$: (A_{n+1}, out, E_0, in) and (A'_{n+1}, out, E_0, in).

At this point, the presence of an object $e_{i,1}$ or an object $\overline{e}_{i,1}$ in a membrane 2 of the configuration $\mathcal{C}_{(3n+1)+1}$ indicates that this membrane codifies a truth assignment making the first clause true.

In the next computation step, those membranes will incorporate an object E_2 coming from the skin by applying the following rules from $\mathcal{R}_2$:

$$\left.\begin{array}{r}
(e_{i,1}, out, E_2\, T_i, in) \\
(\overline{e}_{i,1}, out, E_2\, F_i, in)
\end{array}\right\} 1 \le i \le n$$

This way, the presence of an object E_2 in a membrane 2 of the configuration $\mathcal{C}_{(3n+1)+2}$ indicates that this membrane codifies a truth assignment making true the first clause and that is ready to check the second clause of the formula. That is, from this moment, the membranes labelled with 2 not making true the first clause will not evolve.

In the j-th iteration ($2 \leq j \leq m$) of the aforementioned loop, the truth assignments making true the clause C_j of the formula are checked, taking into account that only the truth assignments containing the object E_j will be checked, since only these membranes make clauses $C_1, \dots, C_{j-1}$ of φ true. This is accomplished by applying the following rules from $\mathcal{R}_2$:

$$\left.\begin{array}{l}(E_j\, T_i, out; e_{i,j}, in)\\ (E_j\, F_i, out; \overline{e}_{i,j}, in)\end{array}\right\} 1 \leq i \leq n \wedge 1 \leq j \leq m$$

Then, the presence of an object $e_{i,j}$ or an object $\overline{e}_{i,j}$ in a membrane 2 of the configuration $\mathcal{C}_{(3n+1)+2\cdot(j-1)+1}$ indicates that this membrane codifies a truth assignment making clauses $C_1, \dots, C_j$ of φ true. Following this, those membranes will incorporate an object E_{j+1} coming from the skin by applying the following rules from $\mathcal{R}_2$:

$$\left.\begin{array}{l}(e_{i,j}, out, E_{j+1}\, T_i, in)\\ (\overline{e}_{i,j}, out, E_{j+1}\, F_i, in)\end{array}\right\} 1 \leq i \leq n \wedge 1 \leq j \leq m-1$$

If the input formula φ is satisfiable, then in some membrane labelled with 2 of the configuration $\mathcal{C}_{(3n+1)+2(m-1)+1}$ there will exist an object $e_{i,m}$ or an object $\overline{e}_{i,m}$. This indicates that the truth assignment that this membrane codifies makes true all the clauses from φ and, consequently, makes true the input formula. In this case, the checking phase ends up by applying a rule from $\mathcal{R}_2$ of the kind $(e_{i,m}\, E_0\,; out)$ or $(\overline{e}_{i,m}\, E_0\,; out)$ making an object E_0 go to the skin membrane of the configuration $\mathcal{C}_{(3n+1)+2(m-1)+2}$, where also the object f_{3n+2m} has been produced.

If the input formula φ is not satisfiable, the no membrane labelled with 2 of the configuration $\mathcal{C}_{(3n+1)+2(m-1)+1}$ contains an object $e_{i,m}$ neither an object $\overline{e}_{i,m}$. In this case, the checking phase ends up by applying the rule $(f'_{3n+2m}\,, out; f'_{3n+2m+1}\,, in) \in \mathcal{R}_3$ (in fact, this is the only rule applicable to the configuration $\mathcal{C}_{(3n+1)+2(m-1)+1}$).

The checking phase ends at step $(3n + 1) + 2(m - 1) + 2 = 3n + 2m + 1$.

Output phase

If the input formula φ is satisfiable, then objects E_0 and f_{3n+2m} will appear in the input membrane of the configuration $\mathcal{C}_{3n+2m+1}$. Then, by applying the rule $(E_0\, f_{3n+2m}\ \texttt{yes}\,; out)$ in the skin membrane, the object `yes` is released into the environment, providing and affirmative answer at computation step $(3n + 1) + 2m + 1 = 3n + 2m + 2$.

If the input formula φ is not satisfiable, then objects f_{3n+2m} and `yes` are present in the skin membrane of the configuration $\mathcal{C}_{(3n+1)+2(m-1)+1} = \mathcal{C}_{3n+2m}$, but not the object E_0. In this case, the only applicable rule in the system is $(f'_{3n+2m}, out; f'_{3n+2m+1}, in)$ in the membrane 3 and in the next computation step only the rule $(f'_{3n+2m+1}\,\texttt{no}\,; out) \in \mathcal{R}_3$ is applicable. Consequently, objects $f_{3n+2m}, \texttt{yes}, f'_{3n+2m+1}$ and no appear in the skin membrane of the configuration $\mathcal{C}_{3n+2m+2}$. Then, by applying the rule $(f_{3n+2m}\,\texttt{no}\,; out)$ in the skin membrane, an object no will be released into the environment, providing a negative answer in the step $3n + 2m + 3$.

Hence, the output phase takes 1 computation step in the case of an affirmative answer, and 2 computation steps in the case of a negative answer.

6 A Formal Verification

In this Section we show that the family $\mathbf{\Pi} = \{\Pi(\langle n, m\rangle) \mid n, m \in \mathbb{N}\}$ considered in the previous section provides a polynomial time solution to SAT problem according to the Definition 3.2. For that, we must prove that it is polynomially uniform by Turing machines and that there exists a polynomial encoding (cod, s) of SAT problem in the family $\mathbf{\Pi}$ such that $\mathbf{\Pi}$ is polynomially bounded, sound and complete with regards to $(\texttt{SAT}, cod, s)$.

6.1 Polynomial Uniformity of the Family

In this subsection, we shall show that the family $\mathbf{\Pi} = \{\Pi(\langle n, m\rangle) \mid n, m \in \mathbb{N}\}$ defined above is polynomially uniform by Turing machines. To this aim we prove that $\Pi(\langle n, m\rangle)$ is built in polynomial time with respect to the size parameter m and n of instances of SAT problem.

It is easy to check that the rules of a system $\Pi(\langle n, m\rangle)$ of the family are recursively defined through the values n (that represents the number of variables of the input formula) and m (that represents the number of clauses of the input formula). The amount of resources to construct $\Pi(\langle n, m\rangle)$ is of a polynomial order in the numbers n and m, as shown below:

1. The size of the working alphabet is of the order $\Theta(n^2 \cdot m)$.
2. The initial number of cells is $3 \in \Theta(1)$.
3. The initial number of objects in membranes is $9n + 3m + 17 \in \Theta(\max\{n, m\})$.
4. The total number of rules is of order $\Theta(n^2 \cdot m)$.
5. The maximum length of a rule is $3 \in \Theta(1)$.

Therefore, there exists a deterministic Turing machine that builds the system $\Pi(\langle n, m\rangle)$ in a polynomial time with respect to n and m.

6.2 Soundness and Completeness of the Family

In the first place, we are going to justify that in the skin membrane of the configuration $\mathcal{C}_n$ objects $e_{i,j}$ appear such that $x_{i,j} \in cod(\varphi)$ and objects $\overline{e}_{i,j}$ appear such that $\overline{x}_{i,j} \in cod(\varphi)$, each of them with multiplicity 2^{n-1}.

Theorem 1. *For each k $(1 \leq k \leq n-1)$, the membrane 1 of the configuration $\mathcal{C}_k$ contains the following multiset of objects:*

$$\begin{array}{l} \{d_{i,j,k}^{2^k} \mid x_{i,j} \in cod(\varphi) \ \wedge \ 1 \leq i \leq n \ \wedge \ 1 \leq j \leq m\} \cup \\ \{\overline{d}_{i,j,k}^{2^k} \mid \overline{x}_{i,j} \in cod(\varphi) \ \wedge \ 1 \leq i \leq n \ \wedge \ 1 \leq j \leq m\} \end{array}$$

Proof. By bounded induction on k. Let us start analyzing the base case $k = 1$. The membrane 1 is the input membrane of the system and, consequently, contains the multiset of objects:

$$\begin{array}{rl} cod(\varphi) = & \{x_{i,j} \mid x_i \in C_j \wedge 1 \leq i \leq n \wedge 1 \leq j \leq m\} \cup \\ & \{\overline{x}_{i,j} \mid \neg x_i \in C_j \wedge 1 \leq i \leq n \wedge 1 \leq j \leq m\} \end{array}$$

Then, by applying the rules from R_1 of the kind

$$\left.\begin{array}{l} (x_{i,j}\,, out; d_{i,j,1}^{2}\,; in) \\ (\overline{x}_{i,j}\,, out; \overline{d}_{i,j,1}^{2}\,; in) \end{array}\right\} 1 \leq i \leq n \wedge 1 \leq j \leq m$$

to the configuration $\mathcal{C}_0$, membrane 1 of $\mathcal{C}_1$ ends containing the multiset of objects:

$$\begin{array}{l} \{d_{i,j,1}^{2} \mid x_{i,j} \in cod(\varphi) \ \wedge \ 1 \leq i \leq n \ \wedge \ 1 \leq j \leq m\} \cup \\ \{\overline{d}_{i,j,1}^{2} \mid x_{i,j} \in cod(\varphi) \ \wedge \ 1 \leq i \leq n \ \wedge \ 1 \leq j \leq m\} \end{array}$$

Consequently, the result holds for $k = 1$.

By induction hypothesis, let us consider k such that $1 \leq k < n-1$ and let us suppose the result holds for k, that is, let us suppose that the membrane 1 of the configuration $\mathcal{C}_k$ contains the multiset of objects:

$$\begin{array}{l} \{d_{i,j,k}^{2^k} \mid x_{i,j} \in cod(\varphi) \ \wedge \ 1 \leq i \leq n \ \wedge \ 1 \leq j \leq m\} \cup \\ \{\overline{d}_{i,j,k}^{2^k} \mid x_{i,j} \in cod(\varphi) \ \wedge \ 1 \leq i \leq n \ \wedge \ 1 \leq j \leq m\} \end{array}$$

Let us see that the result also holds for $k + 1$.

By applying the rules of R_1 of the kind

$$\left.\begin{array}{l} (d_{i,j,k}\,, out; d_{i,j,k+1}^{2}\,, in) \\ (\overline{d}_{i,j,k}\,, out; \overline{d}_{i,j,k+1}^{2}\,, in) \end{array}\right\} 1 \leq i \leq n \wedge 1 \leq j \leq m \wedge 1 \leq k \leq n-2$$

to the configuration $\mathcal{C}_k$ results that the membrane 1 of $\mathcal{C}_{k+1}$ contains the multiset of objects:

$\{d_{i,j,k+1}^{2^{k+1}} \mid x_{i,j} \in cod(\varphi) \wedge 1 \leq i \leq n \wedge 1 \leq j \leq m\} \cup$
$\{\overline{d}_{i,j,k+1}^{2^{k+1}} \mid x_{i,j} \in cod(\varphi) \wedge 1 \leq i \leq n \wedge 1 \leq j \leq m\}$

Consequently, the result holds for $k+1$. This ends up with the proof of the theorem. □

Next, we are going to analyse the evolution of the system thurough every phase in the proposed solution.

Generation phase

We will denote by $(cod(\varphi))_e$ the multiset of $(cod(\varphi))$ where the objects $x_{i,j}$ and $\overline{x}_{i,j}$ are replaced by $e_{i,j}$ and $\overline{e}_{i,j}$, respectively. If in the multiset $(cod(\varphi))_e$ each object has multiplicity 2^k, then we will denote it by $(cod(\varphi))_e^{2^k}$.

Next, let us consider the formulas $\theta_1(p)$, $\theta_2(p)$ and $\theta_3(p)$, where $p = 0, 1, \ldots, n-1$. These formulas indicate the relevant contents of the configurations $\mathcal{C}_{3p+1}$, $\mathcal{C}_{3p+2}$ and $\mathcal{C}_{3p+3}$, respectively.

The formula $\theta_1(p)$ captures the contents of configuration $\mathcal{C}_{3p+1}$, corresponding to the **first step** of each loop iteration. The formula $\theta_1(p)$ is the following:

"*In configuration $\mathcal{C}_{3p+1}$ the following holds:*

- *In the membrane labelled with 1 we can find as relevant objects:*
  - $\rho_{i,3p+1}$ *and* $\tau_{i,3p+1}$ *(for* $1 \leq i \leq n$*), each of them with multiplicity 1 if* $r = 0$ *and multiplicity* 2^{p-1} *if* $p \geq 1$.
  - $\delta_{j,3p+1}$ *(for* $1 \leq j \leq m$*), each of them with multiplicity* 2^p.
  - $f_{3p+1}, \texttt{yes}, f'_0, \ldots, \widehat{f'_{3p+1}}, \ldots, f'_{3n+2m+1}$*, each of them with multiplicity 1.*
  - A_{p+2} , A'_{p+2} , B_{p+2} , B'_{p+2} , S*, each of them with multiplicity* 2^p.
  - $T_{p+1,p+1}$, $F'_{p+1,p+1}$*, each of them with multiplicity* 2^p.
  - *If* $p = 0$ *it contains* A_1 , B_1*, each of them with multiplicity* 1.
  - *If* $0 \leq p \leq n-2$ *then there also exist objects*
    - ⋆ $\alpha_{i,3p+1,k}$, $\alpha'_{i,3p+1,k}$ *(for* $1 \leq i \leq n-1$ *and* $0 \leq k \leq 1$*), each of them with multiplicity* 2^{p-1}*, if* $1 \leq i \leq n-2$*, and with multiplicity 1 if* $p = 0$.
    - ⋆ β_{3p+1}, β'_{3p+1}, β''_{3p+1}*, each of them with multiplicity* 2^p.
    - ⋆ γ_{3p+1}, γ'_{3p+1}, γ''_{3p+1}, γ'''_{3p+1}*, each of them with multiplicity* 2^p.
    - ⋆ y_{p+1} *with multiplicity* 2^{p+1}.
    - ⋆ $r_{1,p+1}, r_{2,p+1}, \ldots, r_{p+1,p+1}$ *with multiplicity* 2^p.

- *If* $1 \leq p \leq n-2$, *it contains the objects* A_{p+1}, A'_{p+1}, B_{p+1}, B'_{p+1}, *each of them with multiplicity* 2^{p-1}.
- *If* $1 \leq p \leq n-1$, *it contains the objects* $T_{i,p+1}$, $T'_{i,p+1}$, $F_{i,p+1}$, $F'_{i,p+1}$, *for* $1 \leq i \leq p$, *each of them with multiplicity* 2^{p-1}.
- *If also* $3p+1 \geq n$, *then it contains* $(cod(\varphi))_e^{2^{n-1}}$.*"*

- *There exist* 2^p *membranes labelled with 2, each of them containing objects* $b_{p+1}, b'_{p+1}, c_{p+1}, c'_{p+1}$ *with multiplicity 1. If* $p \geq 1$, *then each membrane labelled with 2 has a p-tuple of objects* $(\pi_{1,p}, \ldots, \pi_{p,p})$ *such that* $\pi \in \{t, f\}$ *and the corresponding tuples are all different in the different membranes. Thus, for example, for* $p = 1$, *there exist* $2^1 = 2$ *membranes labelled with 2 such that both of them contain objects* b_2, b'_2, c_2, c'_2 *and, additionally, the first of them contains the object* $t_{1,1}$ *and the second one* $f_{1,1}$. *For* $p = 2$, *there exist* $2^2 = 4$ *membranes labelled with 2 such that all of them contain the objects* b_3, b'_3, c_3, c'_3. *In addition, one of them contains* $t_{1,2}, t_{2,2}$, *the second one contains* $f_{1,2}, t_{2,2}$, *the third one contains* $t_{1,2}, f_{2,2}$ *and the fourth one contains* $f_{1,2}, f_{2,2}$.
- *The membrane labelled with 3 contains the objects* f'_{3p+1} *and* `no` *with multiplicity 1.*

The formula $\theta_2(p)$ captures the content of the configuration $\mathcal{C}_{3p+2}$ corresponding to the **second step** of each iteration of the loop. The formula $\theta_2(p)$ is the following:

"*In configuration* $\mathcal{C}_{3p+2}$ *the following holds:*

- *In membrane 1 we can find as relevant objects:*
  - *Objects* $\rho_{i,3p+1}$ *and* $\tau_{i,3p+1}$ *(for* $1 \leq i \leq n$*), each of them with multiplicity* 2^p.
  - *Objects* $\delta_{j,3p+1}$ *(for* $1 \leq j \leq m$*), each of them with multiplicity* 2^{p+1}.
  - *Objects* $f_{3p+1}, \texttt{yes}, f'_0, \ldots, \widehat{f'_{3p+1}}, \ldots, f'_{3n+2m+1}$
  - *If* $0 \leq p \leq n-2$, *then it also contains objects*
    - $\star$ $\alpha_{i,3p+1,k}$, $\alpha'_{i,3p+1,k}$ *(for* $1 \leq i \leq n-1$ *y* $0 \leq k \leq 1$*), each of them with multiplicity* 2^{p-1} *if* $p \geq 1$, *and with multiplicity 1 if* $p = 0$.
    - $\star$ β_{3p+1}, β'_{3p+1}, β''_{3p+1}, *each of them with multiplicity* 2^p.
    - $\star$ γ_{3p+1}, γ'_{3p+1}, γ''_{3p+1}, γ'''_{3p+1}, *each of them with multiplicity* 2^p.
    - $\star$ w_{p+1} *and* a_{p+1} *with multiplicity* 2^{p+1}.
    - $\star$ $s_{1,p+1}$, ... , $s_{p+1,p+1}$, *each of them with multiplicity* 2^p.
  - *If* $0 \leq p \leq n-3$, *then it also contains:*

  - ⋆ *objects* z_{p+1} *with multiplicity* 2^{p+1} *and objects* $u_{1,p+1}, \ldots, u_{p+1,p+1}$, *each of them with multiplicity* 2^p.
- – *If* $3p+1 \geq n$, *then it also contains* $(cod(\varphi))_e^{2^{n-1}}$.*"*

- *There exist* 2^p *membranes labelled with 2, each of them containing objects* B_{p+2}, S, B'_{p+2}, *as well as objects* $T_{p+1,p+1}, A_{p+2}, F'_{p+1,p+1}, A'_{p+2}$, *all of them with multiplicity 1. Also, if* $p \geq 1$, *then each membrane labelled with 2 contains* $2p$*–tuples of objects* $(\pi_{1,p+1}, \pi'_{1,p+1}, \ldots, \pi_{p,p+1}, \pi'_{p,p+1})$, *with* $\pi \in \{T, F\}$, *in such a way that in the different membranes, the corresponding tuples are different with each other.*
- *The membrane labelled with 3 contains the objects* f'_{3p+1} *and* `no`.

The formula $\theta_3(p)$ captures the contents of the configuration $\mathcal{C}_{3p+3}$ corresponding to the **third step** of each loop iteration. The formula $\theta_3(p)$ is the following:

"In the configuration $\mathcal{C}_{3p+3}$ *the following holds:*

- *In the membrane 1 we can find as relevant objects:*
  - – *If* $0 \leq p \leq n-2$, *then there exist objects*
    - ⋆ $\alpha_{i,3p+3,k}$, $\alpha'_{i,3p+3,k}$ *(for* $1 \leq i \leq n-1$ *and* $0 \leq k \leq 1$*), each of them with multiplicity* 2^p.
    - ⋆ β_{3p+3}, β'_{3p+3}, β''_{3p+3}, *each of them with multiplicity* 2^{p+1}.
    - ⋆ γ_{3p+3}, γ'_{3p+3}, γ''_{3p+3}, γ'''_{3p+3}, *each of them with multiplicity* 2^{p+1}.
    - ⋆ $\rho_{i,3p+3}$ *and* $\tau_{i,3p+3}$ *(for* $1 \leq i \leq n$*), each of them with multiplicity* 2^p.
    - ⋆ $\delta_{j,3p+3}$ *(for* $1 \leq j \leq m$*), each of them with multiplicity* 2^{p+1}.
    - ⋆ $f_{3p+3}, \mathtt{yes}, f'_0, \ldots, \widehat{f'_{3p+3}}, \ldots, f'_{3n+2m+1}$, *each of them with multiplicity 1.*
    - ⋆ $b_{p+2}, b'_{p+2}, c_{p+2}, c'_{p+2}$, *each of them with multiplicity* 2^{p+1}.
    - ⋆ $t_{1,p+1}$, $t_{p+1,p+1}$, $f_{1,p+1}$, $f_{p+1,p+1}$ *and* $q_{1,p+2}$, $q_{p+2,p+2}$, *each of them with multiplicity* 2^p.
    - ⋆ w_{p+1} *and* a_{p+1} *with multiplicity* 2^{p+1}.
  - – *If* $0 \leq p \leq n-3$ *then it also contains objects*
    - ⋆ v_{p+2} *with multiplicity* 2^{p+1}.
  - – *If* $p = n-1$ *then it also contains objects*
    - ⋆ $\delta_{j,3p+3}$ *(for* $1 \leq j \leq m$*), each of them with multiplicity* 2^{p+1}.
    - ⋆ $f_{3p+3}, \mathtt{yes}, f'_0, \ldots, \widehat{f'_{3p+3}}, \ldots, f'_{3n+2m+1}$, *each of them with multiplicity 1.*
    - ⋆ T_i *(for* $1 \leq i \leq n$*), each of them with multiplicity* 2^p.

  - ⋆ E_1 *with multiplicity* 2^{p+1}.
- – *If, besides,* $3p+1 \geq n$*, then it also contains* $(cod(\varphi))_e^{2^{n-1}}$.

- *There exist* 2^{p+1} *membranes labelled with 2.* 2^p *of these membranes contain objects* A_{p+2} *and* B_{p+2}*, as well as* $(p+1)$*–tuples* $(\pi_{1,p+1},\dots,\pi_{p+1,p+1})$ *with* $\pi \in \{T,F\}$*, in such a way that* $\pi_{p+1,p+1} = T_{p+1,p+1}$ *and the corresponding tuples are all different in the different membranes.*

  The other 2^p *membranes labelled with 2 contain the objects* A'_{p+2} *and* B'_{p+2}*, as well as* $(p+1)$*–tuples* $(\pi'_{1,p+1},\dots,\pi'_{p+1,p+1})$ *with* $\pi \in \{T,F\}$*, in such a way that* $\pi'_{p+1,p+1} = F'_{p+1,p+1}$ *and the corresponding tuples are all different in the different membranes.*
- *The membrane labelled with 3 contains the objects* f'_{3p+3} *and* $\mathtt{no}$*".*

Next, we are going to prove that the formula $\theta(p) \equiv \theta_1(p) \wedge \theta_2(p) \wedge \theta_3(p)$ is an invariant of the loop associated to the generation phase.

Theorem 2. *For each* $p = 0,1,\dots,n-1$*, the formula* $\theta(p) \equiv \theta_1(p) \wedge \theta_2(p) \wedge \theta_3(p)$ *is true*

Proof. By bounded induction on p. Let us start analyzing the base case $p = 0$; that is, let us show that the formula $\theta(0)$ holds. For this, we have to prove that the formulas $\theta_1(0)$, $\theta_2(0)$ and $\theta_3(0)$ are true.

Let us recall that the initial configuration of the system, $\mathcal{C}_0$ is the following:

$$\begin{aligned}
\mathcal{C}_0(1) = {} & \{\alpha_{i,0,k}, \alpha'_{i,0,k} \mid 1 \leq i \leq n-1 \wedge 0 \leq k \leq 1\} \cup \{\rho_{i,0}, \tau_{i,0} \mid 1 \leq i \leq n\} \cup \\
& \{\beta_0, \beta'_0, \beta''_0, \gamma_0, \gamma'_0, \gamma''_0, \gamma'''_0, c_1, c'_1, b_1, b'_1, v_1, q_{1,1}, f_0, \mathtt{yes}\} \cup \\
& \{\delta_{j,0} \mid 1 \leq j \leq m+1\} \cup \{f'_p \mid 1 \leq p \leq 3n+2m+1\} \cup cod(\varphi) \\
\mathcal{C}_0(2) = {} & \{A_1, B_1\} \\
\mathcal{C}_0(3) = {} & \{f'_0, \mathtt{no}\}
\end{aligned}$$

Then, the following rules are applied to the membranes as stated:

- In membrane 1 the following rules from $\mathcal{R}_1$ are applied:

$$\left.\begin{array}{l}
(\alpha_{i,0,k}, out; \alpha_{i,1,k}, in) \;:\; 1 \le i \le n-1 \wedge 0 \le k \le 1 \\
(\alpha'_{i,0,k}, out; \alpha'_{i,1,k}, in) \;:\; 1 \le i \le n-1 \wedge 0 \le k \le 1 \\
(\beta_0\,, out; \beta_1\; B_2\,, in) \\
(\beta'_0\,, out; \beta'_1\; B'_2\,, in) \\
(\beta''_0\,, out; \beta''_1\; S\,, in) \\
(\gamma_0\,, out; \gamma_1\; T_{1,1}\,, in) \\
(\gamma'_0\,, out; \gamma'_1\; F'_{1,1}\,, in) \\
(\gamma''_0\,, out; \gamma''_1\; A_2\,, in) \\
(\gamma'''_0\,, out; \gamma'''_1\; A'_2\,, in) \\
(\tau_{i,0}\,, out; \tau_{i,1}\,, in) \;:\; 1 \le i \le n \\
(\rho_{i,0}\,, out; \rho_{i,1}\,, in) \;:\; 1 \le i \le n \\
(\delta_{j,0}\,, out; \delta_{j,1}\,, in) \;:\; 1 \le j \le m \\
(f_0\,, out; f_1\,, in) \\
(v_1\,, out; y_1^2\,, in) \\
(q_{1,1}\,, out; r_{1,1}\,, in) \\
(x_{i,j}\,, out; d^2_{i,j,1}\,; in) \;:\; 1 \le i \le n \wedge 1 \le j \le m \wedge 0 \le k \le 1 \\
(\overline{x}_{i,j}\,, out; \overline{d}^2_{i,j,1}\,; in) \;:\; 1 \le i \le n \wedge 1 \le j \le m \wedge 0 \le k \le 1
\end{array}\right\}$$

- In membrane 2, the following rules from $\mathcal{R}_2$ are applied:

$$\left.\begin{array}{l}
(A_1, out; c_1\, c'_1, in) \\
(B_1, out; b_1\, b'_1, in)
\end{array}\right\}$$

- In membrane 3, the following rules from $\mathcal{R}_3$ are applied: $(f'_0, out; f'_1, in)$

  By applying the aforementioned rules, the configuration $\mathcal{C}_1$ holds the following:

- In the membrane labelled with 1 we have the objects:
  - B_2, S, B'_2, $T_{1,1}$, A_2, $F'_{1,1}$, A'_2, A_1, B_1, each one with multiplicity 1.
  - Objects $f_1, \texttt{yes}, f'_0, \widehat{f}'_1, f'_2, \ldots, f'_{3n+2m+1}$
  - Objects $\rho_{i,1}$ and $\tau_{i,1}$ (for $1 \le i \le n$), each one with multiplicity 1.
  - Objects $\delta_{j,1}$ (for $1 \le j \le m$), each one with multiplicity 1.
  - $\alpha_{i,1,k}$, $\alpha'_{i,1,k}$ (for $1 \le i \le n-1$ y $0 \le k \le 1$), each one with multiplicity 1.
  - β_1, β'_1, β''_1, each one with multiplicity 1.
  - γ_1, γ'_1, γ''_1, γ'''_1, each one with multiplicity 1.
  - y_1 with multiplicity 2^1.
  - $r_{1,1}$ with multiplicity 1.
  - For each $1 \le i \le n \wedge 1 \le j \le m$, objects $d^2_{i,j,1}$ such that $x_{i,j} \in cod(\varphi)$ and objects $\overline{d}^2_{i,j,1}$ such that $\overline{x}_{i,j} \in cod(\varphi)$.

- There exists 1 membrane labelled with 2 containing objects b_1, b'_1, c_1, c'_1 with multiplicity 1.
- The membrane labelled with 3 contains the objects f'_1 and `no`.

Hence, the formula $\theta_1(0)$ is true.

At configuration $\mathcal{C}_1$, the following rules are applied to the stated membranes:

- In membrane 1, the following rules from $\mathcal{R}_1$ are applied:

$$\left.\begin{array}{l}
(\alpha_{i,1,k}, out; \alpha_{i,2,k}, in) \ : \ 1 \leq i \leq n-1 \wedge 0 \leq k \leq 1 \\
(\alpha'_{i,1,k}, out; \alpha'_{i,2,k}, in) \ : \ 1 \leq i \leq n-1 \wedge 0 \leq k \leq 1 \\
(\beta_1\,, out; \beta_2\,, in) \\
(\beta'_1\,, out; \beta'_2\,, in) \\
(\beta''_1\,, out; \beta''_2\,, in) \\
(\gamma_1\,, out; \gamma_2\,, in) \\
(\gamma'_1\,, out; \gamma'_2\,, in) \\
(\gamma''_1\,, out; \gamma''_2\,, in) \\
(\gamma'''_1\,, out; \gamma'''_2\,, in) \\
(\tau_{i,1}\,, out; \tau_{i,2}\,, in) \ : \ 1 \leq i \leq n \\
(\rho_{i,1}\,, out; \rho_{i,2}\,, in) \ : \ 1 \leq i \leq n \\
(\delta_{j,1}\,, out; \delta^2_{j,2}\,, in) \ : \ 1 \leq j \leq m \\
(f_1\,, out; f_2\,, in) \\
(y_1\,, out; z_1\, w_1\,, in) \\
(r_{1,1}\,, out; s_{1,1}\, u_{1,1}\,, in) \\
(d_{i,j,1}\,, out; d^2_{i,j,2}\,; in) \ : \ 1 \leq i \leq n \wedge 0 \leq k \leq 1 \\
(\overline{d}_{i,j,1}\,, out; \overline{d}^2_{i,j,2}\,; in) \ : \ 1 \leq i \leq n \wedge 1 \leq j \leq m \wedge 0 \leq k \leq 1
\end{array}\right\}$$

- In membrane 2, the following rules from $\mathcal{R}_2$ are applied:

$$\left.\begin{array}{l}
(b_1, out; B_2\, S, in) \\
(b'_1, out; B'_2, in) \\
(c_1, out; T_{1,1}\, A_2, in) \\
(c'_1, out; F'_{1,1}\, A'_2, in)
\end{array}\right\}$$

- In membrane 3, the following rule from $\mathcal{R}_3$ is applied: $(f'_1, out; f'_2, in)$.

As a result of this, configuration $\mathcal{C}_2$ verifies the following:

- Membrane 1 contains the following objects:
  - f_1, `yes`, $f'_0, \widehat{f'_1}, f'_2, \ldots, f'_{3n+2m+1}$, with multiplicity 1.
  - $\rho_{i,1}$ and $\tau_{i,1}$ (for $1 \leq i \leq n$), each one with multiplicity 1.

  - $\delta_{j,1}$ (for $1 \le j \le m$), each one with multiplicity 1.
  - If $0 \le n-2$, then it also contains the objects
    - ⋆ $\alpha_{i,1,k}$, $\alpha'_{i,1,k}$ (for $1 \le i \le n-1$ and $0 \le k \le 1$), each one with multiplicity 1.
    - ⋆ β_1, β'_1, β''_1, each one with multiplicity 1.
    - ⋆ γ_1, γ'_1, γ''_1, γ'''_1, each one with multiplicity 2^1.
    - ⋆ w_1 and a_1 with multiplicity 2^1.
  - If $0 \le n-3$, then it also contains:
    - ⋆ object z_1, with multiplicity 2^1, and objects $s_{1,1}$, $u_{1,1}$, each with multiplicity 2^1.
- There exists only one membrane labelled with 2 containing objects B_2, S, B'_2, as well as objects $T_{1,1}, A_2, F'_{1,1}, A'_2$.
- The membrane labelled with 3 contains the objects f'_1 and `no`.

Hence, the formula $\theta_2(0)$ is true.

At configuration $\mathcal{C}_2$, the following rules are applied to the stated membranes:

- In membrane 1 the following rules from $\mathcal{R}_1$ are applied:

$$\left.\begin{array}{l}
(\alpha_{i,2,k}, out; \alpha_{i,3,k}, in) \ : \ 1 \le i \le n-1 \wedge 0 \le k \le 1 \\
(\alpha'_{i,2,k}, out; \alpha'_{i,3,k}, in) \ : \ 1 \le i \le n-1 \wedge 0 \le k \le 1 \\
(\beta_2\, , out; \beta_3^2\, , in) \\
(\beta'_2\, , out; \beta_3'^2\, , in) \\
(\beta''_2\, , out; \beta_3''^2\, , in) \\
(\gamma_2\, , out; \gamma_3^2\, , in) \\
(\gamma'_2\, , out; \gamma_3'^2\, , in) \\
(\gamma''_2\, , out; \gamma_3''^2\, , in) \\
(\gamma'''_2\, , out; \gamma_3'''^2\, , in) \\
(\tau_{i,2}\, , out; \tau_{i,3}\, , in) \ : \ 1 \le i \le n \\
(\rho_{i,2}\, , out; \rho_{i,3}\, , in) \ : \ 1 \le i \le n \\
(\delta_{j,2}\, , out; \delta_{j,3}\, , in) \ : \ 1 \le j \le m \\
(f_2\, , out; f_3\, , in) \\
(a_1\, , out; b_2\, b'_2\, , in) \\
(w_1\, , out; c_2\, c'_2\, , in) \\
(u_{1,1}\, , out; q_{1,2}\, q_{2,2}\, , in) \\
(d_{i,j,2}\, , out; d^2_{i,j,3}\, ; in) \ : \ 1 \le i \le n \wedge 0 \le k \le 1 \\
(\overline{d}_{i,j,2}\, , out; \overline{d}^2_{i,j,3}\, ; in) \ : \ 1 \le i \le n \wedge 1 \le j \le m \wedge 0 \le k \le 1
\end{array}\right\}$$

- In membrane 2 the following separation rule from $\mathcal{R}_2$ is applied: $[\ S\ ]_2 \to [\ \Gamma_0\ ]_2\ [\ \Gamma_1\ ]_2$

- In membrane 3 the following rule from $\mathcal{R}_3$ is applied: $(f'_2, out; f'_3, in)$.

Hence, configuration $\mathcal{C}_3$ verifies the following:

- In membrane 1, we can find the following relevant objects (the non-relevant objects are a_1, a'_1, b_1, b'_1, which cannot trigger any rule of the system at that instant):
  - $\alpha_{i,3,k}$, $\alpha'_{i,3,k}$ (for $1 \leq i \leq n-1$ and $0 \leq k \leq 1$), each one with multiplicity 1.
  - β_3, β'_3, β''_3, each one with multiplicity 2^1.
  - γ_3, γ'_3, γ''_3, γ'''_3, each one with multiplicity 2^1.
  - Objects $\rho_{i,3}$ and $\tau_{i,3}$ (for $1 \leq i \leq n$), each one with multiplicity 1.
  - Objects $\delta_{j,3}$ (for $1 \leq j \leq m$), each one with multiplicity 2^1.
  - Objects $f_3, \mathtt{yes}, f'_0, \dots, \widehat{f'_3}, \dots, f'_{3n+2m+1}$
  - If $0 \leq n-2$, there also exist objects:
    - $\star$ $b_2, b'_2, c_2, c'_2, v_2$, each one with multiplicity 2^1.
    - $\star$ $t_{1,1}$, $f_{1,1}$ and $q_{1,2}$, $q_{2,2}$, each one with multiplicity 1.
- There exist 2 membranes labelled with 2. One of them contains objects A_2, B_2 and T_{11}. The other membrane contains objects A'_2, B'_2 and F'_{11}.
- The membrane labelled with 3 contains objects f'_3 and $\mathtt{no}$.

Hence, the formula $\theta_3(0)$ is true and, consequently, the formula $\theta(0)$ is true; that is, the result holds for $p = 0$.

By induction hypothesis, let p be such that $0 \leq p < n-1$, and let us suppose the result holds for p; that is, the formulas $\theta_1(p), \theta_2(p)$ and $\theta_3(p)$ are true. Let us see that the result also holds for $p+1$; that is, the formulas $\theta_1(p+1), \theta_2(p+1)$ and $\theta_3(p+1)$ are also true.

Let us notice that the configuration $\mathcal{C}_{3(p+1)+1}$ is obtained from the configuration $\mathcal{C}_{3(p+1)}$ by applying the following rules:

- In membrane 1, the following rules from $\mathcal{R}_1$ are applied:

$$\left.\begin{array}{l}
(\alpha_{i,3(p+1),k}, out; \alpha_{i,3(p+1)+1,k}\, \Delta^k_{i,p+1}, in) \;:\; 1 \le i \le p \wedge 0 \le k \le 1 \\
(\alpha'_{i,3(p+1),k}, out; \alpha'_{i,3(p+1)+1,k}\, \Delta'^k_{i,p+1}, in) \;:\; 1 \le i \le p \wedge 0 \le k \le 1 \\
(\alpha_{i,3(p+1),k}, out; \alpha_{i,3(p+1)+1,k}, in) \;:\; p+1 \le i \le n-1 \wedge 0 \le k \le 1 \\
(\alpha'_{i,3(p+1),k}, out; \alpha'_{i,3(p+1)+1,k}, in) \;:\; p+1 \le i \le n-1 \wedge 0 \le k \le 1 \\
(\beta_{3(p+1)}\,, out; \beta_{3(p+1)+1}\; B_{p+2}\,, in) \\
(\beta'_{3(p+1)}\,, out; \beta'_{3(p+1)+1}\; B'_{p+2}\,, in) \\
(\beta''_{3(p+1)}\,, out; \beta''_{3(p+1)+1}\; S\,, in) \\
(\gamma_{3(p+1)}\,, out; \gamma_{3(p+1)+1}\; T_{1,1}\,, in) \\
(\gamma'_{3(p+1)}\,, out; \gamma'_{3(p+1)+1}\; F'_{1,1}\,, in) \\
(\gamma''_{3(p+1)}\,, out; \gamma''_{3(p+1)+1}\; A_2\,, in) \\
(\gamma'''_{3(p+1)}\,, out; \gamma'''_{3(p+1)+1}\; A'_2\,, in) \\
(\tau_{i,3(p+1)}\,, out; \tau_{i,3(p+1)+1}\,, in) \;:\; 1 \le i \le n \\
(\rho_{i,3(p+1)}\,, out; \rho_{i,3(p+1)+1}\,, in) \;:\; 1 \le i \le n \\
(\delta_{j,3(p+1)}\,, out; \delta_{j,3(p+1)+1}\,, in) \;:\; 1 \le j \le m \\
(f_{3(p+1)}\,, out; f_{3(p+1)+1}\,, in) \\
(v_{3(p+1)+1}\,, out; y^2_{3(p+1)+1}\,, in) \\
(q_{1,1}\,, out; r_{1,1}\,, in) \;:\; \text{ if } \; p = 0 \\
(q_{i,j}\,, out; r^2_{i,j}\,, in) \;:\; \text{ if } \; p \ge 1 \wedge 1 \le i \le j \le p+1
\end{array}\right\}$$

- In membrane 2, the following rules from $\mathcal{R}_2$ are applied:

$$\left.\begin{array}{l}
(T_{i,p+1}, out; t_{i,p+1}, in) \;:\; 1 \le i \le n \\
(T'_{i,p+1}, out; t_{i,p+1}, in) \;:\; 1 \le i \le n \\
(F_{i,p+1}, out; f_{i,p+1}, in) \;:\; 1 \le i \le n \\
(F'_{i,p+1}, out; f_{i,p+1}, in) \;:\; 1 \le i \le n \\
(A_{p+2}, out; c_{p+2}\, c'_{p+2}, in) \\
(A'_{p+2}, out; c_{p+2}\, c'_{p+2}, in) \\
(B_{p+2}, out; b_{p+2}\, b'_{p+2}, in) \\
(B'_{p+2}, out; b_{p+2}\, b'_{p+2}, in)
\end{array}\right\}$$

- In membrane 3, the following rule from $\mathcal{R}_3$ is applied:

$$(f'_{3(p+1)}, out; f'_{3(p+1)+1}, in)$$

Hence, configuration $\mathcal{C}_{3(p+1)+1}$ verifies the following:

- In the membrane labelled with 1 we can find as relevant objects:
  - $\rho_{i,3(p+1)+1}$ and $\tau_{i,3(p+1)+1}$ (for $1 \le i \le n$), each one with multiplicity 2^p.
  - $\delta_{j,3(p+1)+1}$ (for $1 \le j \le m$), each one with multiplicity 2^{p+1}.
  - $f_{3(p+1)+1}, \texttt{yes}, f'_0, \ldots, \widehat{f'}_{3(p+1)+1}, \ldots, f'_{3n+2m+1}$, each one with multiplicity 1.
  - $A_{p+3}\,,\, A'_{p+3}\,,\, B_{p+3}\,,\, B'_{p+3}\,,\, S$, each one with multiplicity 2^{p+1}.

- $T_{p+2,p+2}$, $F'_{p+2,p+2}$, each one with multiplicity 2^{p+1}.
- $\beta_{3(p+1)+1}$, $\beta'_{3(p+1)+1}$, $\beta''_{3(p+1)+1}$, each one with multiplicity 2^{p+1}.
- $\gamma_{3(p+1)+1}$, $\gamma'_{3(p+1)+1}$, $\gamma''_{3(p+1)+1}$, $\gamma'''_{3(p+1)+1}$, each one with multiplicity 2^{p+1}.
- y_{p+2} each one multiplicity 2^{p+2}.
- $r_{1,p+2}, r_{2,p+2}, \ldots, r_{p+2,p+2}$, with multiplicity 2^{p+1}.
- A_{p+2}, A'_{p+2}, B_{p+2}, B'_{p+2}, each one with multiplicity 2^p.
- $T_{i,p+2}$, $T'_{i,p+2}$, $F_{i,p+2}$, $F'_{i,p+2}$, for $1 \leq i \leq p+1$, each one with multiplicity 2^p.
- In the case $1 \leq p \leq n-3$, it also contains objects $\alpha_{i,3(p+1)+1,k}$, $\alpha'_{i,3(p+1)+1,k}$ (for $1 \leq i \leq n-1$ and $0 \leq k \leq 1$), each one with multiplicity 2^p.
- If $3(p+1)+1 \geq n$, then it also contains $(cod(\varphi))_e^{2^{n-1}}$.

- There exist 2^{p+1} membranes labelled with 2, each of them containing objects $b_{p+2}, b'_{p+2}, c_{p+2}, c'_{p+2}$ with multiplicity 1. Besides, each one of them contains a $(p+1)$-tuple of objects $(\pi_{1,p+1}, \ldots \pi_{p+1,p+1})$ such that $\pi \in \{t, f\}$ and the tuples are all different in the different membranes.
- The membrane labelled with 3 contains the objects $f'_{3(p+1)+1}$ and `no` with multiplicity 1.

Hence, the formula $\theta_1(p+1)$ is true. To prove that the formula $\theta_2(p+1)$ is true, we notice that configuration $\mathcal{C}_{3(p+1)+2}$ is obtained from configuration $\mathcal{C}_{3(p+1)+1}$ by applying the following rules to the stated membranes:

- In membrane 1, the following rules from $\mathcal{R}_1$ are applied:

$(\alpha_{i,3(p+1)+1,k}, out; \alpha_{i,3(p+1)+2,k}, in)\ :\ 1 \le i \le n-1 \wedge 0 \le k \le 1$
$(\alpha'_{i,3(p+1)+1,k}, out; \alpha'_{i,3(p+1)+2,k}, in)\ :\ 1 \le i \le n-1 \wedge 0 \le k \le 1$
$(\beta_{3(p+1)+1}\,, out; \beta_{3(p+1)+2}\,, in)$
$(\beta'_{3(p+1)+1}\,, out; \beta'_{3(p+1)+2}\,, in)$
$(\beta''_{3(p+1)+1}\,, out; \beta''_{3(p+1)+2}\,, in)$
$(\gamma_{3(p+1)+1}\,, out; \gamma_{3(p+1)+2}\,, in)$
$(\gamma'_{3(p+1)+1}\,, out; \gamma'_{3(p+1)+2}\,, in)$
$(\gamma''_{3(p+1)+1}\,, out; \gamma''_{3(p+1)+2}\,, in)$
$(\gamma'''_{3(p+1)+1}\,, out; \gamma'''_{3(p+1)+2}\,, in)$
$(\tau_{i,3(p+1)+1}\,, out; \tau_{i,3(p+1)+2}\,, in)\ :\ 1 \le i \le n$
$(\rho_{i,3(p+1)+1}\,, out; \rho_{i,3(p+1)+2}\,, in)\ :\ 1 \le i \le n$
$(\delta_{j,3(p+1)+1}\,, out; \delta^2_{j,3(p+1)+2}\,, in)\ :\ 1 \le j \le m$
$(f_{3(p+1)+1}\,, out; f_{3(p+1)+2}\,, in)$
$(y_1\,, out; z_1\, w_1\,, in)$
$(r_{1,3(p+1)+1}\,, out; s_{1,3(p+1)+1}\, u_{1,3(p+1)+1}\,, in)\ :\ p+1 \le n-2$
$(r_{1,3(p+1)+1}\,, out; s_{1,3(p+1)+1}\, in)\ :\ p+1 = n-1$
$(d_{i,j,3(p+1)+1}\,, out; d^2_{i,j,3(p+1)+2}\,; in)\ :\ 3(p+1)+1 \le n-1 \wedge 1 \le i \le n \wedge 0 \le k \le 1$
$(\overline{d}_{i,j,1}\,, out; \overline{d}^2_{i,j,2}\,; in)\ :\ 3(p+1)+1 \le n-1 \wedge 1 \le i \le n \wedge 1 \le j \le m \wedge 0 \le k \le 1$
$(d_{i,j,3(p+1)+1}\,, out; e_{i,j}\,; in)\ :\ 3(p+1)+1 = n \wedge 1 \le i \le n \wedge 0 \le k \le 1$
$(\overline{d}_{i,j,3(p+1)+1}\,, out; \overline{e}_{i,j}\,; in)\ :\ 3(p+1)+1 = n \wedge 1 \le i \le n \wedge 1 \le j \le m \wedge 0 \le k \le 1$

- In membrane 2, the following rules from $\mathcal{R}_2$ are applied:

$$\left.\begin{array}{l}(t_{i,p+1}, out; T_{i,p+2}\, {T'}_{i,p+2}\,, in)\\(f_{i,p+1}, out; F_{i,p+2}\, F'_{i,p+2}\,, in)\\(b_{p+2}, out; B_{p+3}\, S, in)\\(b_{p+2}, out; B'_{p+3}, in)\\(c_{p+2}, out; T_{p+2,p+2}\, A_{p+3}\,, in)\\(c'_{p+2}, out; F'_{p+2,p+2}\, A'_{p+3}\,, in)\end{array}\right\}$$

- In membrane 3, the following rule from $\mathcal{R}_3$ is applied:
  $(f'_{3(p+1)+1}, out; f'_{3(p+1)+2}, in)$.

Hence, configuration $\mathcal{C}_{3(p+1)+2}$ verifies the following:

- In membrane 1, we can find as relevant objects:
  - Objects $\rho_{i,3(p+1)+1}$ and $\tau_{i,3(p+1)+1}$ (for $1 \le i \le n$), each of them with multiplicity 2^{p+1}.
  - Objects $\delta_{j,3(p+1)+1}$ (for $1 \le j \le m$), each of them with multiplicity 2^{p+2}.
  - Objects $f_{3(p+1)+1}, \texttt{yes}, f'_0, \ldots, \widehat{f'}_{3(p+1)+1}, \ldots, f'_{3n+2m+1}$
  - If $p+1 \le n-2$ then it also contains objects
    - ⋆ $\alpha_{i,3(p+1)+1,k}$, $\alpha'_{i,3(p+1)+1,k}$ (for $1 \le i \le n-1$ and $0 \le k \le 1$), each of them with multiplicity 2^p.

- ⋆ $\beta_{3(p+1)+1}$, $\beta'_{3(p+1)+1}$, $\beta''_{3(p+1)+1}$, each of them with multiplicity 2^{p+1}.
- ⋆ $\gamma_{3(p+1)+1}$, $\gamma'_{3(p+1)+1}$, $\gamma''_{3(p+1)+1}$, $\gamma'''_{3(p+1)+1}$, each of them with multiplicity 2^p.
- ⋆ w_{p+2} and a_{p+2} with multiplicity 2^{p+1}.
- ⋆ $s_{1,p+2}$, ... , $s_{p+2,p+2}$, each of them with multiplicity 2^{p+1}.

– If $p+1 \leq n-3$ then it also contains objects
  - ⋆ z_{p+2} with multiplicity 2^{p+2} and objects $u_{1,p+2}$, ... , $u_{p+2,p+2}$, each with multiplicity 2^{p+1}.

– If, besides, $3(p+1)+1 \geq n$, then it contains $(cod(\varphi))_e^{2^{n-1}}$.

- There exist 2^{p+1} membranes labelled with 2 each of them containing objects B_{p+3}, S, B'_{p+3}, as well as the objects $T_{p+2,p+2}, A_{p+3}, F'_{p+2,p+2}, A'_{p+3}$ all of them with multiplicity 1. Besides, each membrane labelled with 2 contains $2(p+1)$–tuples of objects $(\pi_{1,p+2}, \pi'_{1,p+2}, \ldots, \pi_{p+1,p+2}, \pi'_{p+1,p+2})$ with $\pi \in \{T, F\}$, in such a way that and the tuples are all different in the different membranes.
- The membrane labelled with 3 contains the objects $f'_{3(p+1)+1}$ and `no`.

Hence, the formula $\theta_2(p+1)$ is true. To prove that the formula $\theta_3(p+1)$ is true, we notice that the configuration $\mathcal{C}_{3(p+1)+3}$ is obtained from the configuration $\mathcal{C}_{3(p+1)+2}$ by applying the following rules to the stated membranes:

- In membrane 1, the following rules from $\mathcal{R}_1$ are applied:

$(\alpha_{i,3(p+1)+2,k}, out; \alpha^2_{i,3(p+1)+3,k}, in)$: $p+1 \leq n-2 \wedge 1 \leq i \leq n-1 \wedge 0 \leq k \leq 1$
$(\alpha'_{i,3(p+1)+2,k}, out; \alpha'^2_{i,3(p+1)+3,k}, in)$: $p+1 \leq n-2 \wedge 1 \leq i \leq n-1 \wedge 0 \leq k \leq 1$
$(\beta_{3(p+1)+2}, out; \beta^2_{3(p+1)+3}, in)$: $p+1 \leq n-2$
$(\beta'_{3(p+1)+2}, out; \beta'^2_{3(p+1)+3}, in)$: $p+1 \leq n-2$
$(\beta''_{3(p+1)+2}, out; \beta''^2_{3(p+1)+3}, in)$: $p+1 \leq n-2$
$(\gamma_{3(p+1)+2}, out; \gamma^2_{3(p+1)+3}, in)$: $p+1 \leq n-2$
$(\gamma'_{3(p+1)+2}, out; \gamma'^2_{3(p+1)+3}, in)$: $p+1 \leq n-2$
$(\gamma''_{3(p+1)+2}, out; \gamma''^2_{3(p+1)+3}, in)$: $p+1 \leq n-2$
$(\gamma'''_{3(p+1)+2}, out; \gamma'''^2_{3(p+1)+3}, in)$: $p+1 \leq n-2$
$(\tau_{i,3(p+1)+2}, out; \tau_{i,3(p+1)+3}, in)$: $p+1 \leq n-2 \wedge 1 \leq i \leq n$
$(\rho_{i,3(p+1)+2}, out; \rho_{i,3(p+1)+3}, in)$: $p+1 \leq n-2 \wedge 1 \leq i \leq n$
$(\delta_{j,3(p+1)+2}, out; \delta_{j,3(p+1)+3}, in)$: $p+1 \leq n-2 \wedge 1 \leq j \leq m$
$(\tau_{i,3(p+1)+2}, out; T_i, in)$: $p+1 = n-1 \wedge 1 \leq i \leq n$
$(\rho_{i,3(p+1)+2}, out; F_i, in)$: $p+1 = n-1 \wedge 1 \leq i \leq n$
$(\delta_{1,3(p+1)+2}, out; E_1, in)$: $p+1 = n-1$
$(\delta_{j,3(p+1)+2}, out; \delta_{j,3(p+1)+3}, in)$: $p+1 = n-1 \wedge 2 \leq j \leq m$
$(f_{3(p+1)+2}, out; f_{3(p+1)+3}, in)$
$(a_{p+1}, out; b_{p+2}\, b'_{p+2}, in)$: $p+1 \leq n-1$
$(w_{p+1}, out; c_{p+2}\, c'_{p+2}, in)$: $p+1 \leq n-1$
$(z_{p+1}, out; v_{p+2}, in)$: $p+1 \leq n-2$
$(u_{1,p+1}, out; q_{1,p+2}\, q_{2,p+2}, in)$: $p+1 \leq n-3$
$(u_{i,p+1}, out; q_{i,p+2}\, q_{2,2}, in)$: $p+1 \leq n-3 \wedge 1 \leq i \leq n-1 \wedge 1 \leq i \leq p+1$
$(s_{i,p+1}, out; t_{1,p+2}\, f_{1,p+2}, in)$: $p+1 \leq n-2 \wedge 1 \leq i \leq p+1$

- In membrane 2, the following separation rule from $\mathcal{R}_2$ is applied: $[\ S\ ]_2 \rightarrow [\ \Gamma_0\ ]_2\ [\ \Gamma_1\ ]_2$
- In membrane 3, the following rule from $\mathcal{R}_3$ is applied:
  $(f'_{3(p+1)+2}, out; f'_{3(p+1)+3}, in)$

Hence, the configuration $\mathcal{C}_{3(p+1)+3}$ verifies the following:

- In membrane 1, we can find as relevant objects:
  – If $p+1 \leq n-2$, then there exist objects:
    ⋆ $\alpha_{i,3(p+1)+3,k}$, $\alpha'_{i,3(p+1)+3,k}$ (for $1 \leq i \leq n-1$ and $0 \leq k \leq 1$), each of them with multiplicity 2^{p+1}.
    ⋆ $\beta_{3(p+1)+3}$, $\beta'_{3(p+1)+3}$, $\beta''_{3(p+1)+3}$, each of them with multiplicity 2^{p+2}.
    ⋆ $\gamma_{3(p+1)+3}$, $\gamma'_{3(p+1)+3}$, $\gamma''_{3(p+1)+3}$, $\gamma'''_{3(p+1)+3}$, each of them with multiplicity 2^{p+2}.
    ⋆ $\rho_{i,3(p+1)+3}$ and $\tau_{i,3(p+1)+3}$ (for $1 \leq i \leq n$), each of them with multiplicity 2^{p+1}.
    ⋆ $\delta_{j,3(p+1)+3}$ (for $1 \leq j \leq m$), each of them with multiplicity 2^{p+2}.

- ⋆ $f_{3(p+1)+3}, \texttt{yes}, f'_0, \ldots, \widehat{f'}_{3(p+1)+3}, \ldots, f'_{3n+2m+1}$, each of them with multiplicity 1.
- ⋆ $b_{p+3}, b'_{p+3}, c_{p+3}, c'_{p+3}$, each of them with multiplicity 2^{p+2}.
- ⋆ $t_{1,p+2}$, $t_{p+2,p+2}$, $f_{1,p+2}$, $f_{p+2,p+2}$ and $q_{1,p+3}$, $q_{p+3,p+3}$, each of them with multiplicity 2^{p+1}.
- ⋆ w_{p+2} and a_{p+2}, with multiplicity 2^{p+2}.

– If $p+1 \leq n-3$, then it also contains objects:
  - ⋆ v_{p+3}, with multiplicity 2^{p+2}.
  - ⋆ $q_{1,p+3}$, $q_{p+3,p+3}$, each of them with multiplicity 2^{p+1}.

– If $p+1 = n-1$, then it also contains objects:
  - ⋆ $\delta_{j,3(p+1)+3}$ (for $1 \leq j \leq m$), each of them with multiplicity 2^{p+2}.
  - ⋆ $f_{3(p+1)+3}, \texttt{yes}, f'_0, \ldots, \widehat{f'}_{3(p+1)+3}, \ldots, f'_{3n+2m+1}$, each of them with multiplicity 1.
  - ⋆ T_i (for $1 \leq i \leq n$), each of them with multiplicity 2^{p+1}.
  - ⋆ E_1 with multiplicity 2^{p+2}.

– If, besides, $3p+1 \geq n$, then it contains $(cod(\varphi))_e^{2^{n-1}}$.

- There exist 2^{p+2} membranes labelled by 2. 2^{p+1} of them contain objects A_{p+3} and B_{p+3}, as well as $(p+2)$–tuples $(\pi_{1,p+2}, \ldots, \pi_{p+2,p+2})$ with $\pi \in \{T, F\}$, in such a way that $\pi_{p+2,p+2} = T_{p+2,p+2}$ and all tuples are different.

  The other 2^{p+1} membranes labelled by 2 contain the objects A'_{p+3} and B'_{p+3}, as well as $(p+2)$–tuples $(\pi'_{1,p+2}, \ldots, \pi'_{p+2,p+2})$ with $\pi \in \{T, F\}$, in such a way that $\pi'_{p+2,p+2} = F'_{p+2,p+2}$ and all tuples are different.
- The membrane labelled by 3 contains objects $f'_{3(p+1)+3}$ and $\texttt{no}$.

Hence, formula $\theta_3(p+1)$ is true and, consequently, formula $\theta(p+1)$ is true. This completes the proof of the theorem. □

Thus, when completing the aforementioned loop that corresponds to the generation phase, the formula $\theta_3(n-1)$ is true. Consequently, at configuration $\mathcal{C}_{3(n-1)+3} = \mathcal{C}_{3n}$, we have:

- In membrane 1, we can find as relevant objects:
  – $f_{3n}, \texttt{yes}, f'_0, \ldots, \widehat{f'_{3n}}, \ldots, f'_{3n+2m+1}$, each of them with multiplicity 1.
  – $\delta_{j,3n}$ (for $2 \leq j \leq m$), each of them with multiplicity 2^n.

- E_1, with multiplicity 2^n.
- T_i, F_i $(1 \leq i \leq n)$, each of them with multiplicity 2^{n-1}.
- $(cod(\varphi))_e^{2^{n-1}}$.

- There exist 2^n membranes labelled by 2. Half of these membranes contain objects A_{n+1} and B_{n+1}, as well as n–tuples $(\pi_{1,n}, \ldots, \pi_{n,n})$ with $\pi \in \{T, F\}$, in such a way that $\pi_{n,n} = T_{n,n}$ and all tuples are different.

  The other 2^{n-1} membranes labelled by 2 contain the objects A'_{n+1} and B'_{n+1}, as well as n–tuples $(\pi'_{1,n}, \ldots, \pi'_{n,n})$ with $\pi \in \{T, F\}$, in such a way that $\pi'_{n,n} = F'_{n,n}$ and all tuples are different.
- The membrane labelled by 3 contains the objects f'_{3n} and `no`.

The generation phase ends with an additional computation step that allows going from configuration $\mathcal{C}_{3n}$ to configuration $\mathcal{C}_{3n+1}$, whose content is described in the following theorem.

Theorem 3. *At configuration $\mathcal{C}_{3n+1}$ we have:*

- *In the membrane labelled by 1 we can find as relevant objects:*
  - $f_{3n+1}, \mathtt{yes}, f'_0, \ldots, \widehat{f'_{3n+1}}, \ldots, f'_{3n+2m+1}$, *each of them with multiplicity 1.*
  - E_j *(for* $2 \leq j \leq m$*), each of them with multiplicity* 2^n*.*
  - B_{n+1}, B'_{n+1}*, with multiplicity* 2^{n-1}*.*
  - $(cod(\varphi))_e^{2^{n-1}}$*.*
- *There exist* 2^n *membranes labelled by 2 each of them containing objects* A_{n+1} *and* E_1*, as well as n–tuples* $(\pi_1, \ldots, \pi_n)$ *with* $\pi \in \{T, F\}$*, in such a way that in the different membranes, the corresponding tuples are different with each other.*
- *The membrane labelled by 3 contains the objects* f'_{3n+1} *and* `no`*.*

Proof. It is enough to take into account that configuration $\mathcal{C}_{3n+1}$ is obtained from configuration $\mathcal{C}_{3n}$ by applying the following rules to the stated membranes:

- In membrane 1, the following rules from $\mathcal{R}_1$ are applied:

$$\left.\begin{array}{l}(\delta_{j,3n}\,, out; E_j\,, in) \;:\; 2 \leq j \leq m \\ (f_{3n}\,, out; f_{3n+1}\,, in)\end{array}\right\}$$

- In membrane 2, the following rules from $\mathcal{R}_2$ are applied:

$$\left.\begin{array}{l}(B_{n+1}\,, out; E_1\,, in)\\(B'_{n+1}\,, out; E_1\,, in)\\(T_{i,n}\,, out; T_i\,, in)\ :\ 1 \le i \le n\\(T'_{i,n}\,, out; T_i\,, in)\ :\ 1 \le i \le n\\(F_{i,n}\,, out; F_i\,, in)\ :\ 1 \le i \le n\\(F'_{i,n}\,, out; F_i\,, in)\ :\ 1 \le i \le n\end{array}\right\}$$

- In membrane 3, the following rule from $\mathcal{R}_3$ is applied: $(f'_{3n}, out; f'_{3n+1}, in)$.

$\square$

Let us notice that in configuration $\mathcal{C}_{3n+1}$ each of the 2^n membranes labelled by 2 codifies a truth assignment associated with the variables $\{x_1, \ldots, x_n\}$. Thus, if one of these membranes contains object T_i (respectively, F_i), then the membrane codifies a truth assignment that associates the `true` value (resp., the `false` value) to the variable x_i.

Checking phase

As we explained in the computation overview, the checking phase starts at computation step $3n+2$, and consists of a loop with m iterations, taking 2 steps each. Hence, the checking phase takes $2m$ steps. It is worth pointing out that at step p of this loop, clause C_{p+1} is checked.

In this phase no separation rule is applied at any computation step, so all the following configurations have exactly 2^n membranes labelled by 2.

Let us consider the formula $\nu_1(p)$, for $0 \le p \le m-2$, defined as follows:

"*At configuration $\mathcal{C}_{(3n+1)+2p+1}$ we have:*

- *In membrane labelled by 1 we can find as relevant objects:*
  - $f_{(3n+1)+2p+1}, \texttt{yes}, f'_0, \ldots, \widehat{f'}_{(3n+1)+2p+1}, \ldots, f'_{3n+2m+1}$, *each of them with multiplicity 1.*
  - E_j $(p+2 \le j \le m)$*, each of them with multiplicity* 2^n*.*
  - $e_{i,j}$*, such that* $x_{i,j} \in cod(\varphi)$ *and* $j \ge p+2$*, as well as objects* $\overline{e}_{i,j}$*, such that* $\overline{x}_{i,j} \in cod(\varphi)$ *and* $j \ge p+2$*. All these objects appear with multiplicity* 2^{n-1}*.*
- *Each of the* 2^n *membranes labelled by 2 contains an object* E_0*.*
- *A membrane labelled by 2 encodes a truth assignment* σ *making true clauses* $C_1, \ldots, C_p, C_{p+1}$ *of* φ*, if and only if it contains a (single) object* $e_{i,p+1}$ *or an object* $\overline{e}_{i,p+1}$*, for a given* i*,* $1 \le i \le n$*. Besides, in that membrane,* σ *keeps all its values,* T *and* F*, except for the* i*-th, which has been replaced by* $e_{i,p+1}$ *(if the object in its place in the previous configuration was* T_i*) or by* $\overline{e}_{i,p+1}$ *(if the object in its place in the previous configuration was* F_i*).*

- *The membrane labelled by 3 contains the objects $f'_{(3n+1)+2p+1}$ and* no*, both with multiplicity 1."*

Let us consider the formula $\nu_2(p)$, for $0 \leq p \leq m-2$, defined as follows:

"At configuration $\mathcal{C}_{(3n+1)+2p+2}$ we have:

- *In membrane 1, we can find as relevant objects:*
  - $f_{(3n+1)+2p+2}$, yes, $f'_0, \ldots, \widehat{f'}_{(3n+1)+2p+2}, \ldots, f'_{3n+2m+1}$*, each with multiplicity 1.*
  - E_j $(p+3 \leq j \leq m)$*, each with multiplicity* 2^n*.*
  - $e_{i,j}$ *such that* $x_{i,j} \in cod(\varphi)$ *and* $j \geq p+2$*, as well as objects* $\overline{e}_{i,j}$ *such that* $\overline{x}_{i,j} \in cod(\varphi)$ *and* $j \geq p+2$*. All these objects appear with multiplicity* 2^{n-1}*.*
- *Each of the* 2^n *membranes labelled by 2 contains an object* E_0*.*
- *A membrane labelled by 2 encodes a truth assignment σ making true clauses $C_1, \ldots, C_p, C_{p+1}$ of φ, if and only if it contains object E_{p+2}. Besides, in that membrane, σ keeps all its values, T and F.*
- *Membrane 3 contains objects $f'_{(3n+1)+2p+2}$ and* no*, both with multiplicity 1."*

Next, we are going to prove that the formula $\nu(p) \equiv \nu_1(p) \wedge \nu_2(p)$ is an invariant of the loop associated to the checking phase.

Theorem 4. *For each $p = 0, \ldots, m-2$, formula $\nu(p) \equiv \nu_1(p) \wedge \nu_2(p)$ holds.*

Proof. By bounded induction on p. Let us start analyzing the base case $p = 0$; that is, let us show that the formula $\nu(0)$ is true. For this, we have to prove that the formulas $\nu_1(0)$ and $\nu_2(0)$ are true.

First of all, we notice that configuration $\mathcal{C}_{(3n+1)+1}$ is obtained from configuration $\mathcal{C}_{(3n+1)}$ by applying the following rules to the stated membranes:

- In membrane 1, rule $(f_{3n+1}\,, out; f_{(3n+1)+1}\,, in)$ from $\mathcal{R}_1$ is applied.
- In membranes labelled by 2, the following rules from $\mathcal{R}_2$ can be applied:

$$\left.\begin{array}{l} (E_1\,T_i\,, out; e_{i,1}\,, in) \\ (E_1\,F_i\,, out; \overline{e}_{i,1}\,, in) \\ (A_{n+1}\,, out; E_0\,, in) \\ (A'_{n+1}\,, out; E_0\,, in) \end{array}\right\}$$

A membrane labelled by 2 encodes a truth assignment σ making true clause C_1 if and only if there exists a literal $l_{i_0,1}$ in clause C_1 that is true by σ. If $l_{i_0,1} = x_k$, then rule $(E_1\,T_k\,, out; e_{k,1}\,, in)$ is applied; if $l_{i_0,1} = \overline{x}_k$, then rule

$(E_1\, F_k\,, out; \overline{e}_{k,1}\,, in)$ is applied. To sum up, either object $e_{i,1}$ or object $\overline{e}_{i,1}$ appears in a membrane 2 if and only if the truth assignment associated to such membrane makes true clause C_1 of φ.

- In membrane 3, rule $(f'_{3n+1}\,, out; f'_{(3n+1)+1}\,, in)$ from $\mathcal{R}_1$ is applied.

Hence, configuration $\mathcal{C}_{(3n+1)+1}$ verifies the following:

- In membrane 1, we can find as relevant objects:
  - $f_{(3n+1)+1}, \mathtt{yes}, f'_0, \ldots, \widehat{f'}_{(3n+1)+1}, \ldots, f'_{3n+2m+1}$, each of them with multiplicity 1.
  - E_j $(2 \leq j \leq m)$, each of them with multiplicity 2^n.
  - $e_{i,j}$, such that $x_{i,j} \in cod(\varphi)$ and $j \geq 1$, as well as objects $\overline{e}_{i,j}$, such that $\overline{x}_{i,j} \in cod(\varphi)$ and $j \geq 1$. All these objects appear with multiplicity 2^{n-1}.
- Each of the 2^n membranes labelled by 2 contains an object E_0.
- A membrane labelled by 2 encodes a truth assignment σ making true clause C_1 of φ if and only if it contains a (single) object $e_{i,1}$ or an object $\overline{e}_{i,1}$, for a given i, $1 \leq i \leq n$. Besides, in that membrane, σ keeps all its values T and F, except for the i-th. There are two possible cases: (a) if the i-th object was T_i in the previous configuration, then it has been released to the skin membrane in this step, and replaced by $e_{i,1}$; and (b) if the i-th object was F_i in the previous configuration, then it has been released to the skin membrane in this step, and replaced by $\overline{e}_{i,1}$.
- Membrane labelled by 3 contains objects $f'_{(3n+1)+1}$ and $\mathtt{no}$, both with multiplicity 1.

Hence, formula $\nu_1(0)$ holds.

Next, let us show that formula $\nu_2(0)$ also holds. For this purpose, let us notice that configuration $\mathcal{C}_{(3n+1)+2}$ is obtained from configuration $\mathcal{C}_{(3n+1)+1}$ by applying the following rules to the stated membranes:

- In membrane 1, the rule $(f_{(3n+1)+1}\,, out; f_{(3n+1)+2}\,, in)$ from $\mathcal{R}_1$ is applied.
- In membrane 2, rules from $\mathcal{R}_2$ of the following kind are applied:

$$\left.\begin{array}{l}(e_{i,1}\,, out; T_i\, E_{1+1}\,, in)\\(\overline{e}_{i,1}\,, out; F_i\, E_{1+1}\,, in)\end{array}\right\}$$

  Obviously, these rule will be only applied to those membranes labelled by 2 containing an object $e_{i,1}$ or an object $\overline{e}_{i,1}$; that is, to membranes codifying a truth assignment making the first clause true. In this case, by using the previous rule, (a) truth assignment value associated to such membrane is restored; and

(b) object E_2 is incorporated in order for the checking process of the second clause to start. Only membranes labelled by 2 and codifying a truth assignment making the first clause true will carry out this checking.

- In membrane 3, rule $(f'_{(3n+1)+1}\,, out; f'_{(3n+1)+2}\,, in)$ from $\mathcal{R}_1$ is applied

Hence, configuration $\mathcal{C}_{(3n+1)+2}$ verifies the following:

- In membrane 1, we can find as relevant objects:
  - Objects $f_{(3n+1)+2}, \texttt{yes}, f'_0, \ldots, \widehat{f'}_{(3n+1)+2}, \ldots, f'_{3n+2m+1}$, each of them with multiplicity 1.
  - Objects E_j $(3 \le j \le m)$, each of them with multiplicity 2^n.
  - Objects $e_{i,j}$, such that $x_{i,j} \in cod(\varphi)$ and $j \ge 2$, as well as objects $\overline{e}_{i,j}$, such that $\overline{x}_{i,j} \in cod(\varphi)$ and $j \ge 2$. All these objects appear with multiplicity 2^{n-1}.
- Each of the 2^n membranes labelled by 2 contains an object E_0.
- A membrane labelled by 2 encodes a truth assignment σ making true clause C_1 of φ, if and only if it contains an object E_2. Besides, in that membrane, σ keeps all its values, T and F.
- Membrane 3 contains the objects $f'_{(3n+1)+2}$ and $\texttt{no}$, both with multiplicity 1.

Thus, the formula $\nu_2(0)$ holds and, consequently, also the formula $\nu(0)$ does; that is, the result holds for the base case $p = 1$.

By induction hypothesis, let p be such that $0 \le p < m-1$ and let us suppose that the result holds for p; that is, formulas $\nu_1(p)$ and $\nu_2(p)$ hold. Let us see that the result also holds for $p+1$; that is, the formulas $\nu_1(p+1)$ and $\nu_2(p+1)$ are also true.

In order to prove that the result holds for $p+1$,let us notice that configuration $\mathcal{C}_{(3n+1)+2(p+1)+1}$ is obtained the configuration $\mathcal{C}_{(3n+1)+2(p+1)} = \mathcal{C}_{(3n+1)+2p+2}$ (let us recall that the content of this configuration is known because we are assuming that formula $\nu_2(p)$ holds) by applying the following rules:

- In membrane 1, the following rule $(f_{(3n+1)+2(p+1)}\,, out; f_{(3n+1)+2(p+1)+1}\,, in)$ from $\mathcal{R}_1$ is applied.
- In membrane 2, the following rules from $\mathcal{R}_2$ are applied:

$$\left.\begin{array}{l} (E_{p+2}\, T_i\,, out; e_{i,p+2}\,, in) \;:\; 1 \le i \le n \\ (E_{p+2}\, F_i\,, out; \overline{e}_{i,p+2}\,, in) \;:\; 1 \le i \le n \end{array}\right\}$$

  By using these rules, if a membrane 2 codifies a truth assignment σ making true clauses $C_1, \ldots, C_p, C_{p+1}$, then that membrane contains an object E_{p+2} at configuration $\mathcal{C}_{(3n+1)+2(p+1)}$. Thus, if there exists a literal l_{i_0} of clause C_{p+2} that is true by the truth assignment σ, then the following holds: if $l_{i_0} = x_k$, then the rule $(E_{p+2}\, T_k\, , out; e_{k,p+2}\, , in)$ would be applicable, and if $l_{i_0} = \overline{x}_k$, then the rule $(E_{p+2}\, F_k\, , out; \overline{e}_{k,p+2}\, , in)$ would be applicable.

- In membrane 3, the rule $(f'_{(3n+1)+2(p+1)}\, , out; f'_{(3n+1)+2(p+1)+1}\, , in)$ from $\mathcal{R}_3$ is applied.

As a result of this, at configuration $\mathcal{C}_{(3n+1)+2(p+1)+1}$ we have:

- In membrane 1, we can find as relevant objects:
  – Objects $f_{(3n+1)+2(p+1)+1}, \texttt{yes}, f'_0, \dots, \widehat{f'}_{(3n+1)+2(p+1)+1}, \dots, f'_{3n+2m+1}$, each of them with multiplicity 1.
  – Objects E_j $(p+3 \le j \le m)$, each of them with multiplicity 2^n.
  – Objects $e_{i,j}$, such that $x_{i,j} \in cod(\varphi)$ and $j \ge p+3$, as well as objects $\overline{e}_{i,j}$, such that $\overline{x}_{i,j} \in cod(\varphi)$ and $j \ge p+3$. All these objects appear with multiplicity 2^{n-1}.
- Each of the 2^n membranes labelled by 2 contains an object E_0.
- A membrane labelled by 2 encodes a truth assignment σ making true clauses $C_1, \dots, C_p, C_{p+1}, C_{p+2}$ of φ if and only if it contains a (single) object $e_{i,p+2}$ or an object $\overline{e}_{i,p+2}$, for a given i, $1 \le i \le n$. Besides, in that membrane, σ keeps all its values, T and F, excepting the i-th. There are two possible cases: (a) if the i-th object was T_i in the previous configuration, then it has been released to the skin membrane in this step, and replaced by $e_{i,p+2}$; and (b) if the i-th object was F_i in the previous configuration, then it has been released to the skin membrane in this step, and replaced by $\overline{e}_{i,p+2}$.
- The membrane labelled by 3 contains objects $f'_{(3n+1)+2(p+1)+1}$ and $\texttt{no}$ with multiplicity 1.

Hence, the formula $\nu_1(p+1)$ holds.

Next, let us show that the formula $\nu_2(p+1)$ is also true. For this purpose, let us notice that the configuration $\mathcal{C}_{(3n+1)+2(p+1)+2}$ is obtained from the configuration $\mathcal{C}_{(3n+1)+2(p+1)+1}$ by applying the following rules to the stated membranes:

- In membrane 1 the rule $(f_{3n+2(p+1)+1}\, , out; f_{3n+2(p+1)+2}\, , in)$ from $\mathcal{R}_1$ is applied.
- In membrane 2, the following rules $\mathcal{R}_2$ are applied:

$$\left.\begin{array}{l}(e_{i,p+2}\, , out; T_i\, E_{p+3}\, , in)\\ (\overline{e}_{i,p+2}\, , out; F_i\, E_{p+3}\, , in)\end{array}\right\}$$

In configuration $\mathcal{C}_{(3n+1)+2(p+1)+1}$, the truth assignment σ encoded by a membrane 2 makes clauses $C_1, \ldots, C_p, C_{p+1}, C_{p+2}$ true if and only if that membrane contains an object $e_{i,p+2}$ or an object $\overline{e}_{i,p+2}$. In this case, a rule of the type $(e_{i,p+2}\,, out; T_i\, E_{p+3}\,, in)$ or of the type $(\overline{e}_{i,p+2}\,, out; F_i\, E_{p+3}\,, in)$. can be applied. Besides, if neither object $e_{i,j}$ nor $\overline{e}_{i,j}$ appears in a membrane 2 of $\mathcal{C}_{(3n+1)+2(p+1)+1}$, then that membrane will not evolve any more.

- In membrane 3, rule $(f'_{3n+2(p+1)+1}\,, out; f'_{3n+2(p+1)+2}\,, in)$ from $\mathcal{R}_3$ is applied.

Hence, configuration $\mathcal{C}_{(3n+1)+2(p+1)+2}$ verifies the following:

- In membrane 1, we can find as relevant objects:
  - Objects $f_{(3n+1)+2(p+1)+2}, \texttt{yes}, f'_0, \ldots, \widehat{f'}_{(3n+1)+2(p+1)+2}, \ldots, f'_{3n+2m+1}$, each of them with multiplicity 1.
  - Objects E_j $(p+4 \le j \le m)$, each of them with multiplicity 2^n.
  - Objects $e_{i,j}$, such that $x_{i,j} \in cod(\varphi)$ and $j \ge p+3$, as well as objects $\overline{e}_{i,j}$ such that, $\overline{x}_{i,j} \in cod(\varphi)$ and $j \ge p+3$. All these objects appear with multiplicity 2^{n-1}."
- Each of the 2^n membranes labelled by 2 contains an object E_0.
- A membrane labelled by 2 encodes a truth assignment σ making true clauses $C_1, \ldots, C_p, C_{p+1}, C_{p+2}$ of φ, if and only if it contains an object E_{p+3}. Besides, in that membrane, σ keeps all its values, T and F.
- Membrane 3 contains objects $f'_{(3n+1)+2(p+1)+2}$ and $\texttt{no}$ with multiplicity 1.

Thus, the formula $\nu_2(p+1)$ holds, consequently, it also formula $\nu(p+1)$ does; that is, the result holds for $p+1$. This completes the proof of the theorem. □

From the Theorem 4 we deduce that the formula $\nu(m-2)$ holds, and in particular, also formula $\nu_2(m-2)$ does. That is, at configuration $\mathcal{C}_{(3n+1)+2(m-2)+2} = \mathcal{C}_{(3n+1)+2(m-1)}$ we have the following:

- In membrane, we can find as relevant objects:
  - $f_{(3n+1)+2m}, \texttt{yes}, f'_0, \ldots, \widehat{f'}_{(3n+1)+2m}, \ldots, f'_{3n+2m+1}$, each of them with multiplicity 1.
  - $e_{i,m}$ such that $x_{i,m} \in cod(\varphi)$ and $\overline{e}_{i,m}$ such that $\overline{x}_{i,m} \in cod(\varphi)$.
- Each of the 2^n membranes labelled by 2 contains object E_0.
- A membrane labelled by 2 encodes a truth assignment σ making true clauses $C_1, \ldots, C_{m-2}, C_{m-1}$ of φ, if and only if it contains the object $E_{(m-2)+2} = E_m$. Besides, in that membrane σ, keeps all its values, T and F.

- The membrane labelled by 3 contains the objects $f'_{(3n+1)+2m}$ and $\texttt{no}$ with multiplicity 1.

Then, configuration $\mathcal{C}_{(3n+1)+2(m-1)+1}$ is obtained from $\mathcal{C}_{(3n+1)+2(m-1)}$ by applying the following rules to the stated membranes:

- In membrane 1 rule $(f_{(3n+1)+2(m-1)}\,, out; f_{(3n+1)+2(m-1)+1}\,, in)$ from $\mathcal{R}_1$ is applied.
- In membranes 2, the following rules from $\mathcal{R}_2$ can be applied:

$$\left.\begin{array}{l}(E_m\,T_i\,, out;\, e_{i,m}\,, in)\\ (E_m\,F_i\,, out;\, \overline{e}_{i,m}\,, in)\end{array}\right\}$$

- In membrane 3, rule $(f'_{(3n+1)+2(m-1)}\,, out; f'_{(3n+1)+2(m-1)+1}\,, in)$ from $\mathcal{R}_3$ is applied.

Hence, at configuration $\mathcal{C}_{(3n+1)+2(m-1)+1}$ we have the following:

- In the membrane, 1 we can find as relevant objects:
  - $f_{(3n+1)+2(m-1)+1} = f_{3n+2m}, \texttt{yes}, f'_0, \ldots, \widehat{f'}_{3n+2m}, f'_{3n+2m+1}$, each of them with multiplicity 1.
- Each of the 2^n membranes labelled by 2 contains an object E_0.
- A membrane labelled by 2 encodes a truth assignment σ making true the clauses $C_1, \ldots, C_{m-1}, C_m$ of φ if and only if it contains an object $e_{i,m}$ or an object $\overline{e}_{i,m}$; that is, the input formula φ is satisfiable if and only there exists a membrane labelled by 2 that contains an object $e_{i,m}$ or an object $\overline{e}_{i,m}$.
- The membrane labelled by 3 contains the objects $f'_{3n+2m}, \texttt{no}$, each of them with multiplicity 1.

Then, configuration $\mathcal{C}_{(3n+1)+2(m-1)+2}$ is obtained from $\mathcal{C}_{(3n+1)+2(m-1)+1}$ by applying the following rules to the stated membranes:

- In a membrane 2, a rule of the type $(e_{i,m)}\,E_0\,, out)$ or of the type $(\overline{e}_{i,m)}\,E_0\,, out)$ will be applied if an only if the truth assignment σ encoded by that membrane makes the formula φ true.
- In the membranes 3, rule $(f'_{3n+2m}\,, out;\; f'_{3n+2m+1}\,, in)$ from $\mathcal{R}_3$ will be applied.

  Consequently, at configuration $\mathcal{C}_{(3n+1)+2(m-1)+1}$ we have the following:

- Membrane 1 contains an object E_0 if and only if the input formula φ is satisfiable.
- Membrane 3 contains objects $f'_{3n+2m+1}, \texttt{no}$, each of them with multiplicity 1.

Then, the checking phase has finished.

Output phase

Case 1: **Affirmative output**.

Let us assume that input formula φ is satisfiable. In this case, at configuration $\mathcal{C}_{3n+2m+1}$, the skin membrane contains some object E_0 and object f_{3n+2m}, while membrane 3 contains objects $f'_{3n+2m+1}$ and $\texttt{no}$.

Hence, in the next computation step (leading to configuration $\mathcal{C}_{3n+2m+2}$), rule $(E_0\, f_{3n+2m}\ \texttt{yes}\,; out) \in \mathcal{R}_1$ will be applied sending object $\texttt{yes}$ to the environment, and providing an affirmative answer. At the same time rule $(f'_{3n+2m+1}\ \texttt{no}\,; out) \in \mathcal{R}_3$ will be applied sending to the skin object $\texttt{no}$. In this case, configuration $\mathcal{C}_{3n+2m+2}$ is halting since object f_{3n+2m} has been sent to the environment and, consequently rule $(f_{3n+2m}\ \texttt{no}\,; out) \in \mathcal{R}_1$ cannot be applied.

To sum up, the affirmative answer is provided in the computation step $(3n + 1) + 2m + 1 = 3n + 2m + 2$.

Case 2: **Negative output**.

If the input formula φ is not satisfiable, then in the skin membrane of configuration $\mathcal{C}_{3n+2m+1}$ objects f_{3n+2m} and $\texttt{yes}$ will appear, but not the object E_0. In this case, rule $(E_0\, f_{3n+2m}\ \texttt{yes}\,; out) \in \mathcal{R}_1$ will not be applicable to the configuration $\mathcal{C}_{3n+2m+1}$ and, consequently, the only applicable rule to this configuration being $(f'_{3n+2m+1}\ \texttt{no}\,; out) \in \mathcal{R}_3$. Therefore, in the skin membrane of configuration $\mathcal{C}_{3n+2m+2}$ objects f_{3n+2m}, $\texttt{yes}$, $f'_{3n+2m+1}$ and $\texttt{no}$ appear, but not object E_0. In this case, the rule $(E_0\, f_{3n+2m}\ \texttt{yes}\,; out) \in \mathcal{R}_1$ will not be applicable, being rule $(f_{3n+2m}\ \texttt{no}\,; out) \in \mathcal{R}_1$ the only applicable to the system. Execution of this rule will send object $\texttt{no}$ to the environment, providing a negative answer at computation step $3n + 2m + 3$.

Hence, the output phase takes 1 step in the case of an affirmative answer, and 2 steps in the case of a negative answer.

Corollary 1. $\texttt{SAT} \in \mathbf{PMC}_{\mathbf{CSC(3)}}$.

7 P-lingua simulator as a checker of the solution

The formal verification of a solution given in the framework of a computing model is a necessary, and usually very complex to implement, task. In order to assist researchers in designing P system families to efficiently solve hard problems and verifying them, simulation tools are indispensable.

The solution to SAT problem by means of a family from $\mathbf{CSC(3)}$ presented in Section 4 has been extraordinarily complex. It is structured into several modules, each of them performing a specific task. Modules have been designed and checked separately and subsequently incorporated into the general solution. The different modules have been checked (in several relevant instances) with the help of the P-Lingua simulator for the model **CSC** developed in [4]. Regarding the formal verification, the simulator was used to check that the identified invariants were corroborated in the corresponding configurations.

The `P-Lingua` source code that defines a cell-like P system belonging to the family specified above and the corresponding `MeCoSim` custom application source files can be found at [16].

7.1 Results of simulation

We have simulated several P systems of the defined family solving relevant instances to `SAT` problem. Simulation results are shown in Table 1.

Table 1. Formula satisfiability and simulation time

Formula	**n**	**m**	**SAT**	**Time (s)**
$(\bar{x}_1+\bar{x}_2)\cdot x_1\cdot x_2$	2	3	F	0,233
$(\bar{x}_1+\bar{x}_2)\cdot x_2\cdot(\bar{x}_1+x_2)$	2	3	T	0,224
$(x_1+x_2)\cdot(x_1+x_2+\bar{x}_3)\cdot\bar{x}_1\cdot\bar{x}_2$	3	4	F	0,491
$(\bar{x}_1+x_2)\cdot\bar{x}_1\cdot x_3\cdot(\bar{x}_1+x_3)$	3	4	T	0,487
$(x_1+x_4)\cdot(x_1+\bar{x}_4)\cdot x_3\cdot(x_2+\bar{x}_3+x_4)\cdot\bar{x}_1$	4	5	F	0,827
$(x_3+\bar{x}_4)\cdot(\bar{x}_1+x_2+\bar{x}_3+x_4)\cdot(x_1+x_2)\cdot(\bar{x}_1+x_2+x_3+x_4)\cdot(\bar{x}_1+x_3)$	4	5	T	0,981
$(x_1+\bar{x}_2+x_3+x_5)\cdot(\bar{x}_1+x_4)\cdot(\bar{x}_2+\bar{x}_4)\cdot x_4\cdot x_2\cdot(\bar{x}_1+x_2+\bar{x}_3+x_4)$	5	6	F	2,369
$(x_3+x_4)\cdot(x_4+\bar{x}_5)\cdot(\bar{x}_1+x_2+\bar{x}_3+\bar{x}_4)\cdot(x_1+\bar{x}_2+x_4)\cdot(x_1+\bar{x}_3+x_4)\cdot(x_3+x_5)$	5	6	T	2,312
$(x_3+x_5+x_6)\cdot(x_3+\bar{x}_4+x_5+\bar{x}_6)\cdot\bar{x}_3\cdot\bar{x}_6\cdot(x_1+\bar{x}_2+\bar{x}_3+x_5+x_6)\cdot(x_1+x_4+x_5)\cdot(\bar{x}_5+x_6)$	6	7	F	4,877
$(\bar{x}_1+\bar{x}_2+x_5)\cdot(x_2+x_3)\cdot(x_3+\bar{x}_5+\bar{x}_6)\cdot(\bar{x}_1+x_2+\bar{x}_3+x_4+x_5+x_6)\cdot(\bar{x}_2+\bar{x}_3)\cdot(x_2+x_3+x_6)\cdot(x_1+\bar{x}_2+x_3+x_4+x_5+x_6)$	6	7	T	4,195
$(\bar{x}_5+\bar{x}_6+\bar{x}_7)\cdot(x_3+\bar{x}_4+x_7)\cdot(\bar{x}_1+x_3+x_5+x_6+\bar{x}_7)\cdot(x_1+x_3+\bar{x}_5+x_6+x_7)\cdot(x_2+x_6)\cdot(x_2+\bar{x}_6)\cdot\bar{x}_2\cdot(x_2+x_3+x_4+\bar{x}_5+x_7)$	7	8	F	10,320
$(\bar{x}_2+x_5+x_6+x_7)\cdot(x_2+\bar{x}_4+\bar{x}_5+\bar{x}_7)\cdot(x_1+x_2+\bar{x}_3+\bar{x}_6+x_7)\cdot(x_1+x_2+x_3+\bar{x}_5+x_6+\bar{x}_7)\cdot(\bar{x}_3+\bar{x}_5+x_6+\bar{x}_7)\cdot(x_1+x_2+\bar{x}_3+\bar{x}_7)\cdot(\bar{x}_1+x_2+\bar{x}_4+\bar{x}_6)\cdot(x_3+x_5+x_6+\bar{x}_7)$	7	8	T	8,862
$(x_3+x_4+\bar{x}_6+\bar{x}_8)\cdot(x_6+\bar{x}_7)\cdot(\bar{x}_2+x_3+\bar{x}_4+x_5+x_8)\cdot x_7\cdot(x_1+\bar{x}_2+x_5+\bar{x}_7+\bar{x}_8)\cdot(x_2+x_7+x_8)\cdot(\bar{x}_6+\bar{x}_7)\cdot(x_1+x_5+\bar{x}_8)\cdot(x_1+\bar{x}_4+x_5+\bar{x}_6+x_7)$	8	9	F	16,364
$(x_1+\bar{x}_5+\bar{x}_6+\bar{x}_7+\bar{x}_8)\cdot(x_2+x_3+x_4+\bar{x}_6+\bar{x}_7+x_8)\cdot(x_3+x_4+\bar{x}_5+\bar{x}_6+\bar{x}_7+\bar{x}_8)\cdot(\bar{x}_1+\bar{x}_3+\bar{x}_4+\bar{x}_5+x_6+\bar{x}_7+x_8)\cdot(\bar{x}_3+\bar{x}_7)\cdot(x_4+x_5+\bar{x}_7)\cdot(x_1+x_3+\bar{x}_4)\cdot(x_1+\bar{x}_2+\bar{x}_3+\bar{x}_4+\bar{x}_5+\bar{x}_6+\bar{x}_7)\cdot(x_4+\bar{x}_5+\bar{x}_6+x_7+\bar{x}_8)$	8	9	T	18,856

$(\bar{x}_2 + \bar{x}_3 + x_5 + x_7) \cdot (x_2 + x_5 + x_6 + x_7 + x_9) \cdot (\bar{x}_3 + x_5 + x_7 + x_8) \cdot (x_1 + \bar{x}_4 + \bar{x}_5 + x_6 + x_8) \cdot (\bar{x}_2 + x_3 + x_5 + x_7 + x_8 + \bar{x}_9) \cdot (\bar{x}_2 + \bar{x}_4 + x_7 + x_9) \cdot (\bar{x}_2 + x_4 + x_6 + x_9) \cdot x_1 \cdot x_5 \cdot (\bar{x}_1 + \bar{x}_5)$	9	10	F	34,669
$(x_3 + x_8) \cdot (x_1 + \bar{x}_2 + x_5 + \bar{x}_6 + x_9) \cdot (x_3 + x_6 + x_9) \cdot (x_3 + x_5 + \bar{x}_6 + \bar{x}_8) \cdot (x_1 + x_2 + \bar{x}_5 + x_7 + \bar{x}_8 + \bar{x}_9) \cdot (\bar{x}_1 + x_2 + \bar{x}_4 + x_5 + \bar{x}_6 + \bar{x}_7 + x_9) \cdot (x_1 + x_2 + x_4 + \bar{x}_6 + x_8 + \bar{x}_9) \cdot (\bar{x}_1 + x_2 + \bar{x}_3 + \bar{x}_4 + x_7 + \bar{x}_8) \cdot (\bar{x}_1 + x_2 + x_3 + x_5 + \bar{x}_6 + x_8 + \bar{x}_9) \cdot (x_2 + \bar{x}_3 + x_4 + \bar{x}_6 + \bar{x}_7 + \bar{x}_9)$	9	10	T	36,450

Let us recall that in P systems of $\mathbf{CSC}(3)$, there is no replication of objects, but a distribution of them. Consequently, in order to generate an exponential amount of some objects, it is necessary to use the skin membrane, interacting with the environment by using antiport rules with length 3 (in a computation step, an object is released into the environment and, simultaneously, two objects enter the system).

8 Conclusions

In this paper we have studied the computational efficiency of cell-like P systems with symport/antiport rules and membrane separation. A uniform polynomial time solution to `SAT` problem by a family of such P systems which uses communication rules involving at most three objects is given, and the formal verification is shown.

Bearing in mind that $\mathbf{PMC_{CSC(2)}} = \mathbf{P}$(that is, only tractable problems are efficiently solved by families of P systems with symport/antiport rules and membrane separation which uses communication rules with length at most two) an optimal frontier of the efficiency has been obtained with respect to the length of such rules. Specifically, we have shown that, in the framework of P systems with symport/antiport rules and membrane separation, passing from 2 to 3 amounts to passing from non–efficiency to efficiency, assuming that $\mathbf{P} \neq \mathbf{NP}$.

Acknowledgements

The work of L. Valencia-Cabrera, L.F. Macías-Ramos, A. Riscos-Núñez and M.J. Pérez-Jiménez was supported by Project TIN2012-37434 of the Ministerio de Economía y Competitividad of Spain. The work of B. Song and L. Pan was supported by National Natural Science Foundation of China (61033003, 91130034 and 61320106005).

References

1. A. Alhazov, T.O. Ishdorj. Membrane operations in P systems with active membranes. In Gh.Păun, A. Riscos-Núñez, A. Romero-Jiménez, F. Sancho-Caparrini (eds.) *Proceedings of the Second Brainstorming Week on Membrane Computing*, Sevilla, 2-7

February 2004, Research Group on Natural Computing, TR 01/2004, University of Seville, 37-44.
2. T.H. Cormen, C.E. Leiserson, R.L. Rivest. *An Introduction to Algorithms.* The MIT Press, Cambridge, Massachussets, 1994.
3. M.R. Garey, D.S. Johnson. *Computers and Intractability A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, (1979).
4. L.F. Macías-Ramos, L. Valencia-Cabrera, B. Song, T. Song, L. Pan, M.J. Pérez-Jiménez. P-Lingua based software for cell-like P systems with symport/antiport rules. *Fundamenta Informaticae*, 2015, in press.
5. L.F. Macías-Ramos, L. Valencia-Cabrera, B. Song, T. Song, L. Pan, M.J. Pérez-Jiménez. Membrane Fission: A Computational Complexity Perspective. *Complexity*, 2015, in press.
6. S. Morlot, A. Roux. Mechanics of dynamic-mediated membrane fission. *Annhual Review of Biophysics*, **42**, (2013), 629–649.
7. L. Pan, T.-O. Ishdorj. P systems with active membranes and separation rules. *Journal of Universal Computer Science*, **10**, 5 (2004), 630–649.
8. L. Pan, M.J. Pérez-Jiménez. Computational complexity of tissue–like P systems. *Journal of Complexity*, **26**, 3 (2010), 296–315.
9. A. Păun, Gh. Păun, G. Rozenberg. Computing by communication in networks of membranes, *International Journal of Foundations of Computer Science*, **13**, 6 (2002), 779–798
10. A. Păun, Gh. Păun. The power of communication: P systems with symport/antiport, *New Generation Computing*, **20**, 3 (2002), 295–305.
11. Gh. Păun, G. Rozenberg, A. Salomaa (eds.). *The Oxford Handbook of Membrane Computing*, Oxford University Press, Oxford, 2010.
12. Gh. Păun. Attacking **NP**-complete problems. In *Unconventional Models of Computation, UMC'2K* (I. Antoniou, C. Calude, M. J. Dinneen, eds.), Springer-Verlag, 2000, 94-115.
13. M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini, F. Complexity classes in models of cellular computing with membranes. *Natural Computing*, **2**, 3 (2003), 265–285.
14. M.J. Pérez-Jiménez, A. Romero-Jiménez, F. Sancho-Caparrini, A polynomial complexity class in P systems using membrane division, *Journal of Automata, Languages and Combinatorics*, **11**, 4 (2006) 423–434.
15. M.J. Pérez-Jiménez, P. Sosík. An optimal frontier of the efficiency of tissue P systems with cell separation. *Fundamenta Informaticae*, in press, 2015.
16. The MeCoSim Web Site: `http://www.p-lingua.org/mecosim/`.