Formal verification of a transition P system generating the set $\{2^n + n^2 + n : \ n \geq 1\}$

Antonio Pérez-Jiménez; Mario J. Pérez-Jiménez and Fernando Sancho-Caparrini

Dpto. Ciencias de la Computación e Inteligencia Artificial
E.T.S. Ingeniería Informática. Universidad de Sevilla, Spain
Avda. Reina Mercedes s/n. 41012 - Sevilla. España

{pjimenez,marper,fsancho}@us.es

Abstract. In the foundational paper of membrane computing [1], an example of a P system generating exactly all the squares of natural numbers greater than or equal to 1 is given. Also, in the same paper it is remarked that a slightly modification of that P system can generate the set $\{2^n + n^2 + n : \ n \geq 1\}$. A formalization of the syntax of this modification following [3] is presented, aa well as the verification of the designed P system by studying the *critical points* of its computations.

1 Introduction

In October 1998, Gheorghe Păun ([1]) introduced a new computability model, of a distributed and parallel type, based on the notion of *membrane structure*. This model, called *transition P systems*, starts from the observation that the processes which take place in the complex structure of a living cell can be considered *computations*. Following [1], we can consider the P systems as devices which generate numbers: the total number of objects in the output membrane is the number *generated* by a computation.

In the cited paper, Gh. Păun suggests that the P system represented in Figure 1 (with membrane 1 as the output membrane) generates the set $\{2^n + n^2 + n : n \geq 1\}$ of natural numbers.

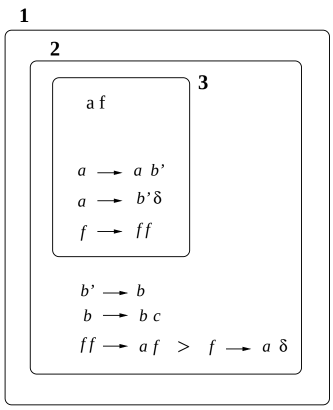


Fig. 1

This paper follows the line started in [4]. We will present a formal description of the syntax of such P system according to the formalization presented in [3]. Moreover, it is proven that the output of any successful configuration of Π encodes a natural number of the form $2^n + n^2 + n$, with $n \geq 1$, and, reciprocally, every natural number of this form is generated by some computation of Π.

This paper is structured in the following way. In Section 2 some preliminaries about formalization of transition P systems are presented, following [3]. In Section 3 the formal syntax of Π is given. In Section 4 characterizations of successful computations of the above P system are established. In Section 5 we show that the output of every successful configuration of Π encodes some $2^n + n^2 + n$, with n a natural number greater than or equal to 1 (*soundness* of the P system) and, also, that $2^n + n^2 + n$, with n a natural number greater than or equal to 1, is generated by some successful computation of Π (*completeness* of the P system).

2 Preliminaries About Transition P Systems

Following [3], a *membrane structure* is a rooted tree, where the nodes are called *membranes*, the root is called *skin*, and the leaves are called *elementary membranes*. Usually, we represent a rooted tree by an ordered pair such that the first component of the pair is the root of the tree and the second component is the adjacency list that consists of n lists, one for each vertex i. The list for vertex i contains just those vertices adjacent to i.

A *cell* over an alphabet, A, is a pair (μ, M), where $\mu = (V(\mu), E(\mu))$ is a membrane structure (from the tree associated to μ we can consider the relation $E^*(\mu)$ as follows: $(x, y) \in E^*(\mu) \iff y$ is a child of x in μ), and M is an application, $M : V(\mu) \longrightarrow \mathbf{M}(A)$ (the set of multisets over A; following [1] and [2], the multisets are represented by strings).

Let (μ, M) be a cell over an alphabet A. Let $x \in V(\mu)$. An *evolution rule* associated with x is a 3-tuple $r = (\boldsymbol{d}_r, \boldsymbol{v}_r, \delta_r)$, where (i) $\boldsymbol{d}_r$ is a multiset over A, (ii) $\boldsymbol{v}_r$ is a function with domain $V(\mu) \cup \{here, out\}$ and range contained in $\mathbf{M}(A)$, where $here, out \notin V(\mu)$ ($here \neq out$), and (iii) $\delta_r \in \{\neg\delta, \delta\}$, with $\neg\delta, \delta \notin A$ ($\neg\delta \neq \delta$).

A *collection R of evolution rules* associated with C is a function with domain in $V(\mu)$ such that for every membrane $x \in V(\mu)$, $R_x = \{r_1^x, \dots, r_{s_x}^x\}$ is a finite set (possibly empty) of (evolution) rules associated with x. A *priority relation over R* is a function ρ, with domain in $V(\mu)$, such that for every membrane $x \in V(\mu)$, ρ_x is a strict partial order over R_x (possibly empty).

A *transition P system* is a 4-tuple $\Pi = (A, C_0, \mathcal{R}, i_0)$, where A is a non-empty finite set (usually called base alphabet), $C_0 = (\mu_0, M_0)$ is a cell over A, $\mathcal{R}$ is an ordered pair (R, ρ) where R is a collection of (evolution) rules associated with C_0 and ρ is a priority relation over R, and i_0 is a node of μ_0 which specifies the output membrane of Π.

A *configuration*, C, of a P system $\Pi = (A, C_0, \mathcal{R}, i_0)$ with $C_0 = (\mu_0, M_0)$, is a cell $C = (\mu, M)$ over A, where $V(\mu) \subseteq V(\mu_0)$, and μ has the same root as μ_0. The configuration C_0 will be called the *initial configuration of Π*. Let $x \in V(\mu_0)$. We say that the (evolution) rule $r \in R_x$ is *semi-applicable* to C if: (a) the membrane associated to node x exists in C, that is, $x \in V(\mu)$; (b) dissolution is not allowed in the root node, that is, if x is the root node of μ, then $\delta_r = \neg\delta$; (c) the membrane associated with x has all the necessary objects to apply the rule, that is, $\boldsymbol{d}_r \leq M(x)$; and (d) nodes where the rule tries to send objects (by means of in_y) are children of x, that is, $\forall\, y \in V(\mu)(\boldsymbol{v}_r(y) \neq \mathbf{0} \rightarrow (x, y) \in E^*(\mu))$.

We say that the rule $r \in R_x$ is *applicable* to C, if it is semi-applicable to C and there is no semi-applicable rules in R_x with a higher priority. That is: $\neg\exists\, r'$ $(r' \in R_x \wedge \rho_x(r', r) \wedge r'$ semi-applicable to $C)$.

We say that $\boldsymbol{p} \in \mathbf{N}^{\mathbf{N}}$ is an *applicability vector* over $x \in V(\mu)$ for C, and we denote it as $\boldsymbol{p} \in \mathbf{Ap}(x, C)$, if: (a) the node is still alive, that is, $\boldsymbol{p} \neq \mathbf{0} \Rightarrow x \in V(\mu)$; (b) it has correct size, that is, $\forall\, j$ $(j > s_x \rightarrow \boldsymbol{p}(j) = 0)$ (where s_x is the number of rules associated with x); (c) every rule can be applied as many times as the vector $\boldsymbol{p}$ indicates; (d) all the rules can be applied simultaneously, that is, $\sum_{j=1}^{s_x} \boldsymbol{p}(j) \otimes \boldsymbol{d}_{r_j^x} \leq M(x)$; and (e) it is maximal, that is,

$$\neg\exists\, \boldsymbol{v} \in \mathbf{N}^{\mathbf{N}} \ (\boldsymbol{p} < \boldsymbol{v} \wedge \boldsymbol{v} \in \mathbf{Ap}(x, C))$$

We say that $P : V(\mu_0) \longrightarrow \mathbf{N}^{\mathbf{N}}$ is an *applicability matrix* over C, denoted $P \in \mathbf{M_{Ap}}(C)$, if for every $x \in V(\mu_0)$ we have that $P(x) \in \mathbf{Ap}(x, C)$. We define

$$\Delta(P, C) = \{x : \ x \in V(\mu) \wedge \exists\, j \ (1 \leq j \leq s_x \wedge P_x(j) \neq 0 \wedge \delta_{r_j^x} = \delta)\}.$$

If P is an applicability matrix over $C = (\mu, M)$, and $V(\mu) = \{i_1, \ldots, i_k\}$, then we denote $P = ((p_1^{i_1}, \ldots, p_{s_{i_1}}^{i_1}), \ldots, (p_1^{i_k}, \ldots, p_{s_{i_k}}^{i_k}))$.

For each node $x \in V(\mu)$, we define the *donors* of x for C in the application of P as the set of nodes that will be dissolved by the application of P and will give their contents to x.

We define the *execution* of P over C, denoted $P(C)$, as the configuration $C' = (\mu', M')$ of Π, where μ' is the membrane structure obtained after the application of the rules indicated by P, and M' is the new contents in the membranes of μ'.

We say that a configuration C_1 of a P system Π yields a configuration C_2 by a *transition in one step* of Π, denoted $C_1 \Rightarrow_\Pi C_2$, if there exists a non–zero applicability matrix over C_1, P, such that $P(C_1) = C_2$.

The *computation tree of a P system Π*, denoted $\mathbf{Comp}(\Pi)$, is a rooted labelled maximal tree defined as follows: The root of the tree is the initial configuration, C_0, of Π. The children of a node are the configurations that follow in one step of transition. Nodes and edges are labelled by configurations and applicability matrices, respectively, in such a way that two labelled nodes C, C' are adjacent in $\mathbf{Comp}(\Pi)$, by means an edge labelled with P, if and only if $P \in \mathbf{M_{Ap}}(C) - \{\mathbf{0}\} \wedge C' = P(C)$. The maximal branches of $\mathbf{Comp}(\Pi)$ will be called *computations* of Π. We will say that a computation of Π *halts* if it is a finite branch. The configurations verifying $\mathbf{M_{Ap}}(C) = \{\mathbf{0}\}$ will be called *halting configurations*.

We say that a computation $\mathcal{C} \equiv C_0 \Rightarrow_\Pi C_1 \Rightarrow_\Pi \ldots \Rightarrow_\Pi C_n$ of a P system $\Pi = (A, C_0, \mathcal{R}, i_0)$ is *successful* if it halts and i_0 is a leaf of the rooted tree μ_n, where $C_n = (\mu_n, M_n)$. We will say that the configuration C_n is *successful* too, and n is the *length* of $\mathcal{C}$. The *numerical output* of a successful computation, $\mathcal{C}$, is $O(\mathcal{C}) = |M_{C_n}(i_0)|$ where C_n is the successful configuration of $\mathcal{C}$. The output of a P system Π is $O(\Pi) = \{O(\mathcal{C}) : \ \mathcal{C}$ is a successful computation of $\Pi\}$.

Let $\Pi = (A, C_0, \mathcal{R}, i_0)$ be a P system. The set of natural numbers generated by Π, denoted $\mathbf{N}(\Pi)$, is defined as follows:

$$\mathbf{N}(\Pi) = \{O(\mathcal{C}) : \ \mathcal{C} \text{ is a successful computation of } \Pi\}$$

3 A Formalization of the Syntax of the P System Π

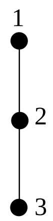
Now we give a formalization of the syntax of the transition P system Π from Figure 1, following the definitions of the above section.

The transition P system we deal with is $\Pi = (A, C_0, \mathcal{R}, i_0)$, where:

(a) The base alphabet is $A = \{a, b, b', c, f\}$.
(b) The initial configuration, $C_0 = (\mu_0, M_0)$, is defined as:

$$\mu_0 = (1, ((1, 2), (2, 3)))$$

That is, μ_0 is the membrane structure given by means of the following rooted tree (with membranes labelled by natural numbers):



M_0 is the function from $\{1, 2, 3\}$ to $\mathbf{M}(A)$ defined as: $M_0(1) = M_0(2) = \emptyset$, y $M_0(3) = \{af\}$.
(c) $\mathcal{R} = (R, \rho)$, where:
   - R is a collection of rules associated with C_0; that is, R is a function with the domain $\{1, 2, 3\}$, defined as: $R(1) = \emptyset$, $R(2) = \{r_2^1, r_2^2, r_2^3, r_2^4\}$ and $R(3) = \{r_3^1, r_3^2, r_3^3\}$, where:
     - $r_2^1 = (d_{r_2^1}, v_{r_2^1}, \delta_{r_2^1})$, with $d_{r_2^1} = \{b'\}$, $v_{r_2^1} : \{1, 2, 3\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_2^1}(1) = v_{r_2^1}(2) = v_{r_2^1}(3) = v_{r_2^1}(out) = \emptyset$; $v_{r_2^1}(here) = \{b\}$, and $\delta_{r_2^1} = -\delta$.
     - $r_2^2 = (d_{r_2^2}, v_{r_2^2}, \delta_{r_2^2})$, con $d_{r_2^2} = \{b\}$, $v_{r_2^2} : \{1, 2, 3\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_2^2}(1) = v_{r_2^2}(2) = v_{r_2^2}(3) = v_{r_2^2}(out) = \emptyset$; $v_{r_2^2}(here) = \{bc\}$, and $\delta_{r_2^2} = -\delta$.
     - $r_2^3 = (d_{r_2^3}, v_{r_2^3}, \delta_{r_2^3})$, con $d_{r_2^3} = \{ff\}$, $v_{r_2^3} : \{1, 2, 3\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_2^3}(1) = v_{r_2^3}(2) = v_{r_2^3}(3) = v_{r_2^3}(out) = \emptyset$; $v_{r_2^2}(here) = \{af\}$, and $\delta_{r_2^3} = -\delta$.
     - $r_2^4 = (d_{r_2^4}, v_{r_2^4}, \delta_{r_2^4})$, con $d_{r_2^4} = \{f\}$, $v_{r_2^4} : \{1, 2, 3\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_2^4}(1) = v_{r_2^4}(2) = v_{r_2^4}(3) = v_{r_2^4}(out) = \emptyset$; $v_{r_2^4}(here) = \{a\}$, and $\delta_{r_2^4} = +\delta$.

- $r_3^1 = (d_{r_3^1}, v_{r_3^1}, \delta_{r_3^1})$, con $d_{r_3^1} = \{a\}$, $v_{r_3^1} : \{1,2,3\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_3^1}(1) = v_{r_3^1}(2) = v_{r_3^1}(3) = v_{r_3^1}(out) = \emptyset$; $v_{r_3^1}(here) = \{ab'\}$, and $\delta_{r_3^1} = -\delta$.
- $r_3^2 = (d_{r_3^2}, v_{r_3^2}, \delta_{r_3^2})$, con $d_{r_3^2} = \{a\}$, $v_{r_3^2} : \{1,2,3\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_3^2}(1) = v_{r_3^2}(2) = v_{r_3^2}(3) = v_{r_3^2}(out) = \emptyset$; $v_{r_3^2}(here) = \{b'\}$, and $\delta_{r_3^2} = +\delta$.
- $r_3^3 = (d_{r_3^3}, v_{r_3^3}, \delta_{r_3^3})$, con $d_{r_3^3} = \{f\}$, $v_{r_3^3} : \{1,2,3\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_3^3}(1) = v_{r_3^3}(2) = v_{r_3^3}(3) = v_{r_3^3}(out) = \emptyset$; $v_{r_3^3}(here) = \{ff\}$, and $\delta_{r_3^3} = -\delta$.

– ρ is the function with domain $\{1,2,3\}$ defined as: $\rho(1) = \rho(3) = \emptyset$ and $\rho(2) = \{(r_2^3, r_2^4)\}$.

(d) The output membrane is $i_0 = 1$.

4 Characterizing the halting configurations of Π

The process of verifying the P system Π is based on the analysis of the content of every membrane in all possible computations of Π. For this, we will consider a function, denoted by **STEP**, that will associate to every computation, $\mathcal{C}$, of Π, every membrane, i, and every natural number, k, the content of the i–th membrane after the execution of k steps of the computation $\mathcal{C}$. If after the execution of the k–th step, the i–th membrane is dissolved, then the value of $\mathbf{STEP}(\mathcal{C}, i, k)$ will not be defined. In this last case, we will denote $\mathbf{STEP}(\mathcal{C}, i, k) \uparrow$, otherwise we will denote $\mathbf{STEP}(\mathcal{C}, i, k) \downarrow$.

Given a computation $\mathcal{C}$ of Π we will denote $\mathcal{C}_0 \Rightarrow_\Pi \mathcal{C}_1 \Rightarrow_\Pi \ldots \Rightarrow_\Pi \mathcal{C}_k \Rightarrow_\Pi \ldots$. That is, $\mathcal{C}_k$ represent the configuration obtained after the execution of k steps of the computation $\mathcal{C}$. Usually, we will denote $\mathbf{STEP}(\mathcal{C}, i, k) = \mathcal{C}_k(i)$. Also, we will denote by $|\mathcal{C}|$ the length of the computation $\mathcal{C}$ that, eventually, can be infinite.

In order to characterize the successful computations of the P system Π, we are going to study what happens in the instant when membrane 3 is dissolved. For this, firstly we are going to determine the content of this membrane in every instant of the computation when it has still not dissolved.

Proposition 1. *For each $m \in \mathbf{N}$ and each computation $\mathcal{C}$ of Π such that $\mathcal{C}_m(3) \downarrow$ it is verified that $\mathcal{C}_m(3) = \{a\ b'^m\ f^{2^m}\}$ and $\mathcal{C}_m(1) = \mathcal{C}_m(2) = \emptyset$.*

Proof. The proof is made by induction on m. In the case $m = 0$, it is enough to having in mind that, according to the initial configuration of Π, we have $\mathcal{C}_0(3) = \{a\ f\}$ and $\mathcal{C}_0(1) = \mathcal{C}_0(2) = \emptyset$.

Let $m \in \mathbf{N}$ and let us suppose that the result is true for m. Let $\mathcal{C}$ be a computation of Π such that $\mathcal{C}_{m+1} \downarrow$. Having in mind that the semantic of basic P systems does not consider *membrane creation*, it has to verify that $\mathcal{C}_m(3) \downarrow$. Hence, from induction hypothesis we deduce that $\mathcal{C}_m(3) = \{a\ b'^m\ f^{2^m}\}$ and $\mathcal{C}_m(1) = \mathcal{C}_m(2) = \emptyset$. Since $\mathcal{C}_{m+1}(3) \downarrow$, it results that in the $(m+1)$–th step of the computation of $\mathcal{C}$ the rules $r_3^1 \equiv a \to ab'$ and $r_3^3 \equiv f \to ff$ have to be applied (in a maximal manner). Consequently, $\mathcal{C}_{m+1}(3) = \{a\ b'^{(m+1)}\ f^{2^{m+1}}\}$. Moreover, $\mathcal{C}_m(1) = \emptyset \Rightarrow \mathcal{C}_{m+1}(1) = \emptyset$ and $\mathcal{C}_m(2) = \emptyset \Rightarrow \mathcal{C}_{m+1}(2) = \emptyset$. This proves that the result is true for $m+1$.

□

Following we prove that the critical points in the computations of the P system Π happen in the instant when membrane 3 is dissolved.

Proposition 2. *For each $m \in \mathbf{N}$ and each computation $\mathcal{C}$ de Π such that $\mathcal{C}_m(3) \downarrow$ and $\mathcal{C}_{m+1}(3) \uparrow$ it is verified that:*

1. *$\mathcal{C}_{m+1}(1) = \emptyset$ and $\mathcal{C}_{m+1}(2) = \{b'^{(m+1)}\ f^{2^{m+1}}\}$.*
2. *For every k $(0 \leq k \leq m)$:*
   $$\mathcal{C}_{m+2+k}(1) = \emptyset$$
   $$\mathcal{C}_{m+2+k}(2) = \{b^{m+1}\ c^{k(m+1)}\ f^{2^{m-k}} a^{2^m + \cdots + 2^{m-k}}\}$$
3. *$\mathcal{C}_{2m+3}(1) = \{b^{m+1}\ c^{(m+1)^2}\ a^{2^{m+1}}\}$ and $\mathcal{C}_{2m+3}(2) \uparrow$.*
4. *The computation $\mathcal{C}$ is successful, its length is $|\mathcal{C}| = 2m+3$, and its output is $2^{(m+1)^2} + (m+1)^2 + (m+1)$.*

Proof. (1) By induction on m. In the case $m = 0$ we consider a computation, $\mathcal{C}$, of Π such that $\mathcal{C}_0(3) \downarrow$ and $\mathcal{C}_1(3) \uparrow$. In this case, the computation $\mathcal{C}$ will apply the rules $r_3^2 \equiv a \rightarrow b'\delta$ and $r_3^3 \equiv f \rightarrow ff$ (in a maximal manner). Hence $\mathcal{C}_1(1) = \mathcal{C}_0(1) = \emptyset$ and $\mathcal{C}_1(2) = \{b'\ f^2\}$.

Let $m \in \mathbf{N}$ and let us suppose that the result is true for m. Let $\mathcal{C}$ a computation of Π such that $\mathcal{C}_{m+1}(3) \downarrow$ and $\mathcal{C}_{m+2}(3) \uparrow$. Having in mind that $\mathcal{C}_{m+1}(3) \downarrow$, from proposition 1 it is deduced that $\mathcal{C}_{m+1}(3) = \{a\ b'^{(m+1)}\ f^{2^{m+1}}\}$ and $\mathcal{C}_{m+1}(1) = \mathcal{C}_{m+1}(2) = \emptyset$. Having in mind that $\mathcal{C}_{m+2}(3) \uparrow$, it results that in the $(m+1)$–th step of the computation $\mathcal{C}$, membrane 2 has to be dissolved; that is, in this step the rules $r_3^2 \equiv a \rightarrow b'\delta$ and $r_3^3 \equiv f \rightarrow ff$ have to be applied (in a maximal manner). Hence $\mathcal{C}_{m+2}(1) = \mathcal{C}_{m+1}(1) = \emptyset$ and $\mathcal{C}_{m+2}(2) = \{b'^{(m+2)}\ f^{2^{m+2}}\}$.
(2) Let $m \in \mathbf{N}$. Let $\mathcal{C}$ be a computation of Π such that $\mathcal{C}_m(3) \downarrow$ and $\mathcal{C}_{m+1}(3) \uparrow$. Let us see that for every k $(0 \leq k \leq m)$ it is verified that $\mathcal{C}_{m+2+k}(1) = \emptyset$ and $\mathcal{C}_{m+2+k}(2) = \{b^{m+1}\ c^{k(m+1)}\ f^{2^{m-k}}\ a^{2^m + \cdots + 2^{m-k}}\}$.

By bounded induction on k. In the case $k = 0$, let us observe that, from item (1), it results that $\mathcal{C}_{m+1}(1) = \emptyset$ and $\mathcal{C}_{m+1}(2) = \{b'^{(m+1)}\ f^{2^{m+1}}\}$. In this situation, in the $(m+2)$–th step of the computation $\mathcal{C}$, the rules $r_2^1 \equiv b' \rightarrow b$ and $r_2^3 \equiv ff \rightarrow af$ will be applied (in a maximal manner). Hence $\mathcal{C}_{m+2}(1) = \mathcal{C}_{m+1}(1) = \emptyset$ and $\mathcal{C}_{m+2}(2) = \{b^{m+1}\ f^{2^m}\ a^{2^m}\}$.

Let $k < m$ and let us suppose that the result is true for k. From induction hypothesis it is deduced that the configuration $\mathcal{C}_{m+2+k}(1) = \emptyset$ and the configuration $\mathcal{C}_{m+2+k}(2) = \{b^{m+1}\ c^{k(m+1)}\ f^{2^{m-k}} a^{2^m + \cdots + 2^{m-k}}\}$. In this situation, in the $(m+2+k+1)$–th step of the computation $\mathcal{C}$, the rules $r_2^2 \equiv b \rightarrow bc$ and $r_2^3 \equiv ff \rightarrow f$ will be applied (in a maximal manner). Hence

$$\begin{cases} \mathcal{C}_{m+2+k+1}(1) = \mathcal{C}_{m+2+k}(1) = \emptyset \\ \mathcal{C}_{m+2+k+1}(2) = \{b^{m+1}\ c^{m+1}\ c^{k(m+1)}\ f^{2^{m-k-1}} a^{2^{m-k-1}} a^{2^m + \cdots + 2^{m-k}}\} \\ \qquad\qquad\qquad\quad = \{b^{m+1}\ c^{(k+1)(m+1)}\ f^{2^{m-(k+1)}}\ a^{2^m + \cdots + 2^{m-k} + 2^{m-(k+1)}}\} \end{cases}$$

(3) Let $m \in \mathbf{N}$. Let $\mathcal{C}$ be a computation of Π such that $\mathcal{C}_m(3) \downarrow$ and $\mathcal{C}_{m+1}(3) \uparrow$. From item (3) it is deduced that the configuration $\mathcal{C}_{2m+2}(1) = \emptyset$ and the configuration $\mathcal{C}_{2m+2}(2) = \{b^{m+1}\ c^{m(m+1)}\ f\ a^{2^m+\cdots+2^0}\}$. In this situation, in the $(2m+3)$–th step of the computation $\mathcal{C}$, the rules $r_2^2 \equiv b \rightarrow bc$ and $r_2^4 \equiv f \rightarrow a\delta$ will be applied (in a maximal manner). Hence $\mathcal{C}_{2m+3}(2) \uparrow$ and $\mathcal{C}_{2m+3}(1) = \{b^{m+1}\ c^{m+1}\ c^{m(m+1)}\ a\ a^{2^m+\cdots+2^0}\} = \{b^{m+1}\ c^{(m+1)^2}\ a^{2^{m+1}}\}$.
(4) Let $m \in \mathbf{N}$. Let $\mathcal{C}$ be a computation of Π such that $\mathcal{C}_m(3) \downarrow$ and $\mathcal{C}_{m+1}(3) \uparrow$. From item (3) it is deduced that $\mathcal{C}$ is successful, because only the membrane 1 is not dissolved and it has no rules. Moreover, $\mathcal{C}_{2m+3}$ is a halting configuration, and the output is

$$|\mathcal{C}_{2m+3}(1)| = |\{b^{m+1}\ c^{(m+1)^2}\ a^{2^{m+1}}\}| = 2^{m+1} + (m+1) + (m+1)^2$$

□

Following, we characterize the successful computation of Π through the instant when membrane 3 is dissolved.

Proposition 3. *Let $\mathcal{C}$ be a computation of Π. The following conditions are equivalent:*

(a) $\mathcal{C}$ is a successful computation.
(b) $\exists m\ (|\mathcal{C}| = 2m+3\ \wedge\ \mathcal{C}_m(3) \downarrow\ \wedge \mathcal{C}_{m+1}(3) \uparrow)$.

Proof. Let $\mathcal{C}$ be a successful computation of Π. Let $k = |\mathcal{C}|$. Firstly, we note that $\mathcal{C}_k(3) \uparrow$. Otherwise, from proposition 1, it would result that $\mathcal{C}_k(3) = \{a\ b'^k\ f^{2^k}\}$ and $\mathcal{C}_k(1) = \mathcal{C}_k(2) = \emptyset$. Then, the computation $\mathcal{C}$ would not be a halting computation, because there would be rules to be applied in membrane 3 (for example, the rules $r_3^1 \equiv a \rightarrow ab'$ and $r_3^3 \equiv f \rightarrow ff$).

Hence, $\mathcal{C}_k(3) \uparrow$. Having in mind that $\mathcal{C}_0(3) \downarrow$, an unique $m \in \mathbf{N}$ such that $\mathcal{C}_m(3) \downarrow$ and $\mathcal{C}_{m+1}(3) \uparrow$ (with $m+1 \leq k$) exists. Consequently, from (4) of proposition 2 it is deduced that $|\mathcal{C}| = 2m+3$.

Reciprocally, let us suppose that there exists a natural number m such that $|\mathcal{C}| = 2m+3\ \wedge\ \mathcal{C}_m(3) \downarrow\ \wedge \mathcal{C}_{m+1}(3) \uparrow$. From (4) of proposition 2 it is deduced that the computation $\mathcal{C}$ is successful.

□

5 Completeness and soundness of the P system Π

In order to give a formal verification of the fact that the P system Π generates all the searched numbers, and no more, we must to prove the following two results:

– The output of every computation of the P system Π encodes a natural number of the searched form (*soundness* of the P system).
– For every number in the searched form, there exists, at least, one computation of the P system Π encoding this number in its output (*completeness* of the P system).

Theorem 1. (**Soundness**) *If $\mathcal{C}$ is a successful computation of the P system Π, then there exists $m \in \mathbf{N}$ such that the output of $\mathcal{C}$ encodes the number $2^{m+1} + (m+1)^2 + (m+1)$.*

Proof. Let $\mathcal{C}$ be a successful computation of the P system Π. From proposition 3 it is deduced that there exists $m \in \mathbf{N}$ such that $|\mathcal{C}| = 2m+3$ and $\mathcal{C}_m(3) \downarrow \wedge\ \mathcal{C}_{m+1}(3) \uparrow$. Then, from (4) of proposition 2 it is deduced that the output of the computation $\mathcal{C}$ is $|\mathcal{C}_{2m+3}(1)| = 2^{(m+1)^2} + (m+1)^2 + (m+1)$.

$\square$

Proposition 4. *For every $m \in \mathbf{N}$ there exists a computation, $\mathcal{C}$, of Π such that $\mathcal{C}_m(3) \downarrow\ \wedge\ \mathcal{C}_{m+1}(3) \uparrow$.*

Proof. By induction on m. In the case $m = 0$, it is enough to consider the configuration $\mathcal{C}_1$ obtained from the initial one, $\mathcal{C}_0$, by applying the rules $r_3^2 \equiv a \to b'\delta$ and $r_3^3 \equiv f \to ff$ (in a maximal manner). Hence, $\mathcal{C}_0(3) = \{af\}$ and $\mathcal{C}_1(3) =\uparrow$.

Let $m \in \mathbf{N}$ and let us suppose that the result is true for m. Then, there exists a computation, $\mathcal{C}$, of Π such that $\mathcal{C}_m(3) \downarrow\ \wedge\ \mathcal{C}_{m+1}(3) \uparrow$. Let $\mathcal{C}'_{m+1}$ the configuration obtained from $\mathcal{C}_m$ by applying the rules $r_3^1 \equiv a \to ab'$ and $r_3^3 \equiv f \to ff$ (in a maximal manner). Let $\mathcal{C}'_{m+2}$ be the configuration obtained from $\mathcal{C}'_{m+1}$ by applying the rules $r_3^2 \equiv a \to b'\delta$ and $r_3^3 \equiv f \to ff$ (in a maximal manner). Let $\mathcal{C}'$ be the computation $\mathcal{C}_0 \Rightarrow_\Pi \mathcal{C}_1 \Rightarrow_\Pi \ldots \Rightarrow_\Pi \mathcal{C}_m \Rightarrow_\Pi \mathcal{C}'_{m+1} \Rightarrow_\Pi \mathcal{C}'_{m+2} \Rightarrow_\Pi \ldots$. This computation verifies that

- $\mathcal{C}'_{m+1}(3) \downarrow$, since from proposition 1 it is deduced that $\mathcal{C}_m(3) = \{ab'^m f^{2^m}\}$ and, therefore, $\mathcal{C}'_{m+1}(3) = \{ab'^{(m+1)} f^{2^{m+1}}\}$.
- $\mathcal{C}'_{m+2}(3) \uparrow$, because of the application of the rule r_3^2 in membrane 3.

$\square$

After an analysis of the previous proof, we can note that for a given $m \in \mathbf{N}$, there is *only one* computation, $\mathcal{C}$, of Π verifying above conditions.

Theorem 2. *(***Completeness***) For every $m \in \mathbf{N}$ there exists a computation, $\mathcal{C}$, of the P system Π such that $\mathcal{C}$ is successful and, also, its output encodes the number $2^{m+1} + (m+1)^2 + (m+1)$.*

Proof. Let $m \in \mathbf{N}$. From Proposition 4 we deduce that there exists a computation, $\mathcal{C}$, of Π such that $\mathcal{C}_m(3) \downarrow$ and $\mathcal{C}_{m+1}(3) \uparrow$. From (4) in Proposition 2 we have the computation $\mathcal{C}$ is successful and, also, the output is

$$|\mathcal{C}_{2m+3}(1)| = \{b^{m+1}\ c^{(m+1)^2}\ a^{2^{m+1}}\} = 2^{m+1} + (m+1)^2 + (m+1)$$

$\square$

6 Conclusions

The formal verification of a transition P system is based in the characterizations of its successful computations, for this, an analysis of the content of its membranes in every configuration is needed. The study of *critical points* of the computations can give formulas over the configurations that will be invariants of the whole process of evolution of the P system. Also, the veracity of such a formula in every configuration must provide relevant information to characterize the successful computations of the P system.

In this paper the formal verification of a transition P system generating the set of natural numbers $\{2^n + n^2 + n : \; n \geq 1\}$ has been obtained. The process of verification is based in the analysis of a *critical point* appearing in every halting configuration: the instant when a relevant membrane is dissolved. Moreover, in this work a detailed study of *every* computation of the P system is given, and a classification of this computations is obtained.

The formalization and study of the verification of P systems must represent an important step to the treatment of them through automated reasoning systems, as well as a way to improved their designs.

Acknowledgement

The authors wish to acknowledge the support of the project TIC2002-04220-C03-01 of the Ministerio de Ciencia y Tecnología of Spain, cofinanced by FEDER funds.

References

1. Păun Gh. Computing with membranes, *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143, and *Turku Center for Computer Science-TUCS Report* No 208, 1998 (www.tucs.fi).
2. Păun, Gh.; Rozenberg, G. A guide to membrane computing, *Theoretical Computer Science*, 287, 2002, 73–100.
3. Pérez–Jiménez, M.J.; F. Sancho-Caparrini, F. A formalization of basic P systems, *Fundamenta Informaticae*, 49, 2002, 261–272.
4. Pérez–Jiménez, M.J.; Sancho-Caparrini, F.. Verifying a Psystem Generating Squares. *Romanian Journal of Information Science and Technology*, 5, 1-2, 2002, 181–191.