Fundamenta Informaticae XX (2007) 1–11 1

IOS Press

Verifying a P system generating squares

Mario J. Pérez-Jiménez

Fernando Sancho-Caparrini

Dpto. Ciencias de la Computación e Inteligencia Artificial

Universidad de Sevilla, España

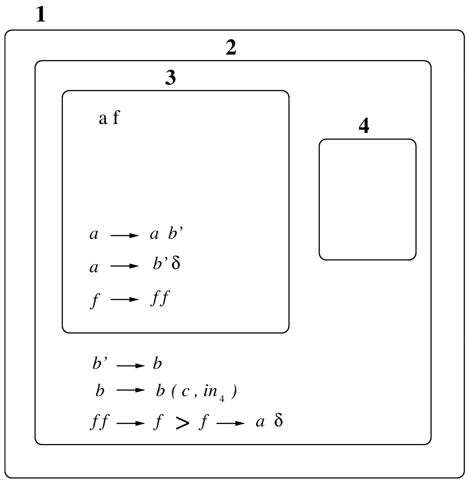
{*Mario.Perez,Fernando.Sancho*}*@cs.us.es*

Abstract. In [1], an example of a P system generating exactly all the squares of natural numbers greater than or equal to 1 is given. Nevertheless, only an informal proof of this result is presented. In this paper we study a similar P system (only one evolution rule is modified). A formalization of the syntax of the P system following [3] is given, and we perform the verification of this P system through soundness and completeness: (a) every successful computation generates a square greater than or equal to 1 (*soundness*); (b) every natural number greater or equal to 1 is the output of a successful computation of the system (*completeness*). Then we establish the formal verification through the study of the *critical points* of the computations of the P system that give to us important information to characterize the successful computations.

1. Introduction

In October 1998, Gheorghe Păun ([1]) introduced a new computability model, of a distributed parallel type, based on the notion of *membrane structure*. This model, called *transition P system*, start 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 sum of multiplicities of objects in the output membrane is the generated number by a computation.

In [1], the P system from Figure 1 is considered, where membrane 4 is the output one. Also, it is said that the set of natural numbers generated by the above P system is $N(\Pi) = \{n^2 : \ n \geq 1\}$.



P sistema Π

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

A *cell* (or *super-cell*) over an alphabet, A, is a pair (μ, M), where $\mu = (V(\mu), E(\mu))$ is a membrane structure (we consider $E^*(\mu)$ 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 $r = (\vec{d_r}, \vec{v}_r, \delta_r)$, where (i) $\vec{d_r}$ is a multiset over A, (ii) $\vec{v}_r$ is a function with the domain $V(\mu) \cup \{here, out\}$ and the 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 the domain $V(\mu)$ 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 the domain $V(\mu)$, 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, $\vec{d_r} \leq M(x)$; and (d) nodes where the rule tries to send objects (by means of in_y) are children of x, that is, $\forall\, y \in V(\mu)(\vec{v}_r(y) \neq \vec{0} \rightarrow (x, y) \in E^*(\mu))$.

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

We say that $P : V(\mu_0) \longrightarrow \mathbf{N}^{\mathbf{N}}$ is an *applicability matrix* over C, denoted $P \in \mathbf{M}_{\mathbf{Ap}}(C)$, 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 follows:

$$Don(x, P, C) = \begin{cases} \emptyset, & \text{if } x \in \Delta(P, C), \\ \{y \in V(\mu) : y \in \Delta(P, C)\ \wedge\ x \rightsquigarrow_\mu y\ \wedge \\ \quad \wedge\ \forall\, z \in V(\mu)(x \rightsquigarrow_\mu z \rightsquigarrow_\mu y \rightarrow z \in \Delta(P, C))\}, & \text{if } x \notin \Delta(P, C). \end{cases}$$

We define the *execution* of P over C, denoted $P(C)$, as the configuration $C' = (\mu', M')$ of Π, where:

- μ' is the rooted tree obtained from μ by means of:
  - $V(\mu') = V(\mu) - \Delta(P, C)$.
  - If $x, y \in V(\mu')$, then:

$$\begin{aligned}(x, y) \in E^*(\mu') \Leftrightarrow\ & \exists\, x_0, \ldots, x_n \in V(\mu)(x_1, \ldots, x_{n-1} \in \Delta(P, C)\ \wedge\ x_0 = x\ \wedge \\ & x_n = y\ \wedge\ \forall\, i\ (0 \leq i < n \rightarrow (x_i, x_{i+1}) \in E^*(\mu)))\end{aligned}$$

- $M'(x) = \begin{cases} M''(x) \cup \displaystyle\bigcup_{y \in Don(x, P, C)} M''(y), & \text{if } x \notin \Delta(P, C), \\ \emptyset, & \text{if } x \in \Delta(P, C). \end{cases}$

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 labeled 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 labeled by configurations and applicability matrices, respectively, in such way that two labeled nodes C, C' are adjacent in $\mathbf{Comp}(\Pi)$, by means an edge labeled 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 $C_n = (\mu_n, M_n)$. Then we say that configuration C_n is *successful*, 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: $\mathbf{N}(\Pi) = \{O(\mathcal{C}) : \mathcal{C}$ is a successful computation of $\Pi\}$.

3. A Formalization of the Syntax of the P System Π

Next, we are going to formalize the syntax of the P system Π from Figure 1, following the definitions of the above section.

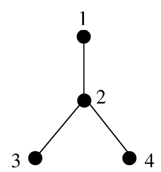
The P system we deal with is $\Pi = (A, C_0, \mathcal{R}, i_0)$, where:

(a) The base alphabet is $A = \{a, b, b', c, f\}$.

(b) The initial configuration, $C_0 = (\mu_0, M_0)$, is defined as follows:

$$\mu_0 = (1, ((1, 2), (2, 1, 3, 4), (3, 2), (4, 2)))$$

That is, μ_0 is the membrane structure given by means of the following rooted tree:



M_0 is the application from $\{1, 2, 3, 4\}$ to $\mathbf{M}(A)$ defined as: $M_0(1) = M_0(2) = M_0(4) = \emptyset$ and $M_0(3) = \{af\}$.

(c) $\mathcal{R} = (R, \rho)$, where:

- R is a collection of rules associated with C_0; that is, R is an application with the domain $\{1, 2, 3, 4\}$, defined as: $R(1) = R(4) = \emptyset$, $R(2) = \{r_1^2, r_2^2, r_3^2, r_4^2\}$ y $R(3) = \{r_1^3, r_2^3, r_3^3\}$, where:

- $r_1^2 = (d_{r_1^2}, v_{r_1^2}, \delta_{r_1^2})$, with $d_{r_1^2} = \{b'\}$, $v_{r_1^2} : \{1,2,3,4\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_1^2}(1) = v_{r_1^2}(2) = v_{r_1^2}(3) = v_{r_1^2}(4) = v_{r_1^2}(out) = \emptyset$; $v_{r_1^2}(here) = \{b\}$, and $\delta_{r_1^2} = -\delta$.
- $r_2^2 = (d_{r_2^2}, v_{r_2^2}, \delta_{r_2^2})$, with $d_{r_2^2} = \{b\}$, $v_{r_2^2} : \{1,2,3,4\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_2^2}(1) = v_{r_2^2}(2) = v_{r_2^2}(3) = v_{r_2^2}(out) = \emptyset$; $v_{r_2^2}(4) = \{c\}, v_{r_2^2}(here) = \{b\}$, and $\delta_{r_2^2} = -\delta$.
- $r_3^2 = (d_{r_3^2}, v_{r_3^2}, \delta_{r_3^2})$, with $d_{r_3^2} = \{ff\}$, $v_{r_3^2} : \{1,2,3,4\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_3^2}(1) = v_{r_3^2}(2) = v_{r_3^2}(3) = v_{r_3^2}(4) = v_{r_3^2}(out) = \emptyset$; $v_{r_3^2}(here) = \{f\}$, and $\delta_{r_3^2} = -\delta$.
- $r_4^2 = (d_{r_4^2}, v_{r_4^2}, \delta_{r_4^2})$, with $d_{r_4^2} = \{f\}$, $v_{r_4^2} : \{1,2,3,4\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_4^2}(1) = v_{r_4^2}(2) = v_{r_4^2}(3) = v_{r_4^2}(4) = v_{r_4^2}(out) = \emptyset$; $v_{r_4^2}(here) = \{a\}$, and $\delta_{r_4^2} = +\delta$.
- $r_1^3 = (d_{r_1^3}, v_{r_1^3}, \delta_{r_1^3})$, with $d_{r_1^3} = \{a\}$, $v_{r_1^3} : \{1,2,3,4\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_1^3}(1) = v_{r_1^3}(2) = v_{r_1^3}(3) = v_{r_1^3}(4) = v_{r_1^3}(out) = \emptyset$; $v_{r_1^3}(here) = \{ab'\}$, and $\delta_{r_1^3} = -\delta$.
- $r_2^3 = (d_{r_2^3}, v_{r_2^3}, \delta_{r_2^3})$, with $d_{r_2^3} = \{a\}$, $v_{r_2^3} : \{1,2,3,4\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_2^3}(1) = v_{r_2^3}(2) = v_{r_2^3}(3) = v_{r_2^3}(4) = v_{r_2^3}(out) = \emptyset$; $v_{r_2^3}(here) = \{b'\}$, and $\delta_{r_2^3} = +\delta$.
- $r_3^3 = (d_{r_3^3}, v_{r_3^3}, \delta_{r_3^3})$, with $d_{r_3^3} = \{f\}$, $v_{r_3^3} : \{1,2,3,4\} \cup \{here, out\} \to \mathbf{M}(A)$ given as $v_{r_3^3}(1) = v_{r_3^3}(2) = v_{r_3^3}(3) = v_{r_3^3}(4) = v_{r_3^3}(out) = \emptyset$; $v_{r_3^3}(here) = \{ff\}$, and $\delta_{r_3^3} = -\delta$.

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

(d) The output membrane is $i_0 = 4$.

4. Characterizing Successful Configurations of Π

Let Π' be any P system designed to generate a set B of natural numbers. To establish the verification of Π' in relation to the set B, a predicate over configurations (that is, over $Comp(\Pi') \times \mathbf{N}$) is looked for, which is, in some way, an invariant of the process of computation in the P system Π'. That is, this predicate will be true for every computation, $\mathcal{C}$, of Π' and every natural number from B. Also, the truth of the predicate over all the configurations of Π' must extract important information to establish the soundness and completeness of Π related to the generation of the set B.

The process of verification of our P system Π is based on the analysis of the content of every membrane in every computation that can be obtained in Π. 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, $\mathcal{C}_k$ represents the configuration obtained after the execution of k steps in the computation $\mathcal{C}$. In a natural way, a partial function, $\mathbf{STEP} : Comp(\Pi) \times \mathbf{N} \times V(\mu_0) \to \mathbf{M}(A)$, can be defined to assign to every computation $\mathcal{C}$, of Π, every natural number k and every membrane i of the P system, the content of the membrane i after the execution of k steps in the computation $\mathcal{C}$. If, after the execution of the k-th step, the membrane i is dissolved, then $\mathbf{STEP}(\mathcal{C}, k, i)$ is not defined and in this case we denote $\mathbf{STEP}(\mathcal{C}, k, i) \uparrow$. Otherwise, we denote $\mathbf{STEP}(\mathcal{C}, k, i) \downarrow$. In general, we denote $\mathbf{STEP}(\mathcal{C}, k, i) = \mathcal{C}_k(i)$. We denote by $|\mathcal{C}|$ the length of the computation $\mathcal{C}$ (it can be infinite)

Definition 4.1. For every membrane i and every computation $\mathcal{C}$ of Π, we define $\delta(\mathcal{C}, i) = \min\{m : \mathcal{C}_m(i) \uparrow\}$.

Having in mind that no membrane is dissolved in the initial configuration of a P system, we have that $\delta(\mathcal{C}, i) \geq 1$, for every $\mathcal{C} \in Comp(\Pi)$ and every membrane i of Π.

Given a P system Π and a membrane i of Π, we can define in a natural way a partial function $D_i : Comp(\Pi) \to \mathbf{N} - \{0\}$ by $D_i(\mathcal{C}) = \delta(\mathcal{C}, i)$. That is, D_i assigns to every computation $\mathcal{C}$ of Π a natural number representing the instant where the membrane i of Π is dissolved (if any).

To establish that the considered P system Π generates the set $\{n^2 : n \geq 1\}$, we will try to characterize the successful computations of Π.

For that, first we will give a predicate over the configurations of Π to be an invariant along the execution of the P system Π. Let us consider the formula
$\theta(\mathcal{C}, n) \equiv (n < \delta(\mathcal{C}, 3) \to \mathcal{C}_n = (\mu_0, (\emptyset, \emptyset, ab'^n f^{2^n}, \emptyset))) \wedge (n = \delta(\mathcal{C}, 3) \to \mathcal{C}$ success. $\wedge\ O(\mathcal{C}) = n^2)$

To make easier the proofs and following Section 2, the applicability vector will be expressed with a finite number of components (as many as rules the membrane has). We denote by $\mathbf{0}$ the vector with all null components,irrespectively which is its size.

If $C = (\mu, M)$ is a cell, where $V(\mu) = \{a_1, \ldots, a_n\} \subset \mathbf{N}$ with $a_1 < \cdots < a_n$, we note $M = (M(a_1), \ldots, M(a_n))$. For simplicity of notation, we represent the multisets by means of the associated word, and $\emptyset$ will be the empty multiset.

First, we are going to determine every configuration of the P system before membrane 3 is dissolved.

Proposition 4.1. For every computation $\mathcal{C}$ of Π we have:

$$\forall\, n\ (n < \delta(\mathcal{C}, 3) \to \mathcal{C}_n = (\mu_0, (\emptyset, \emptyset, ab'^n f^{2^n}, \emptyset))).$$

Proof:
Let $\mathcal{C}$ be a computation of Π. Let us prove the result by induction on n. For the base case, $n = 0$, it is enough to consider that $\delta(\mathcal{C}, 3) \geq 1$ and $\mathcal{C}_0 = (\mu_0, (\emptyset, \emptyset, af, \emptyset))$.

Let $n \in \mathbf{N}$ such that $(n < \delta(\mathcal{C}, 3) \to \mathcal{C}_n = (\mu_0, (\emptyset, \emptyset, ab'^n f^{2^n}, \emptyset))$. If $n + 1 < \delta(\mathcal{C}, 3)$, then $n < \delta(\mathcal{C}, 3)$ and, hence, $\mathcal{C}_n = (\mu_0, (\emptyset, \emptyset, ab'^n f^{2^n}, \emptyset))$. As $\mathcal{C}_{n+1}(3) \downarrow$, we deduce that the configuration $\mathcal{C}_{n+1}$ is obtained from $\mathcal{C}_n$ by applying the matrix $\vec{p} = (\mathbf{0}, \mathbf{0}, (1, 0, 2^n), \mathbf{0})$ (the applicability matrix over $\mathcal{C}_n$), since no dissolution is applied over membrane 3. Then, we have that $\mathcal{C}_{n+1} = \vec{p}(\mathcal{C}_n) = (\mu_0, (\emptyset, \emptyset, ab'^{(n+1)} f^{2^{n+1}}, \emptyset))$. □

Next, we will prove that a *critical point* of the computations of the P system Π appears in the moment when membrane 3 is dissolved. That is, we will justify that knowing when membrane 3 is dissolved is important in order to characterize the successful computations of Π.

Proposition 4.2. For every computation $\mathcal{C}$ of the P system Π such that $n = \delta(\mathcal{C}, 3) < \infty$, we have:

1. $\mathcal{C}_n = (\mu', (\emptyset, b'^n f^{2^n}, \emptyset))$, where $\mu' = (1, ((1, 2), (2, 1, 4), (4, 2)))$.
2. For every k such that $0 \leq k \leq n - 1$, we have $\mathcal{C}_{n+1+k} = (\mu', (\emptyset, b^n f^{2^{n-k-1}}, c^{kn}))$, where μ' is as above.
3. $\mathcal{C}_{2n+1} = (\mu'', (ab^n, c^{n^2}))$, where $\mu'' = (1, ((1, 4), (4, 1)))$.

4. The computation $\mathcal{C}$ is successful, its length is $|\mathcal{C}| = 2n + 1$, and, also, the numerical output of this computation is $O(\mathcal{C}) = n^2$.

Proof:

1. If $n = \delta(\mathcal{C}, 3) < \infty$, then $0 \leq n - 1 < \delta(\mathcal{C}, 3)$. From Proposition 4.1, we deduce that $\mathcal{C}_{n-1} = (\mu_0, (\emptyset, \emptyset, ab'^{(n-1)} f^{2^{n-1}}, \emptyset))$. Having in mind that $\delta(\mathcal{C}, 3) = n$, we obtain that the configuration $\mathcal{C}_n$ is obtained from $\mathcal{C}_{n-1}$ by executing the applicability matrix $\vec{p} = (\mathbf{0}, \mathbf{0}, (0, 1, 2^{n-1}), \mathbf{0})$ over $\mathcal{C}_{n-1}$. Hence, $\mathcal{C}_n = \vec{p}(\mathcal{C}_{n-1}) = (\mu', (\emptyset, b'^n f^{2^n}, \emptyset))$, where $\mu' = (1, ((1, 2), (2, 1, 4), (4, 2)))$.

2. We provethis assertion by induction on k. For the base case, $k = 0$, let us observe that from (1) we obtain that $\mathcal{C}_n = (\mu', (\emptyset, b'^n f^{2^n}, \emptyset))$. In this situation, since $n \geq 1$, it is possible to apply the rule r_3^2 to the membrane 2 and then, by the way in which the priority is interpreted, the rule r_4^2 cannot be applied to $\mathcal{C}_n$ (this rule would dissolve membrane 2). Hence, the only matrix applicability over $\mathcal{C}_n$ will be $\vec{p} = (\mathbf{0}, (n, 0, 2^{n-1}, 0), \mathbf{0})$. In consequence, $\mathcal{C}_{n+1} = \vec{p}(\mathcal{C}_n) = (\mu', (\emptyset, b^n f^{2^{n-1}}, \emptyset))$.

   Let k be such that $0 \leq k < n - 1$, and let us suppose that $\mathcal{C}_{n+1+k} = (\mu', (\emptyset, b^n f^{2^{n-k-1}}, c^{kn}))$. Since $n - k - 1 > 0$, we deduce that it is possible to apply the rule r_3^2 to membrane 2 and then, the only applicability matrix over $\mathcal{C}_{n+1+k}$ is $\vec{p} = (\mathbf{0}, (0, n, 2^{n-k-2}, 0), \mathbf{0})$. Hence, we have that

   $$\mathcal{C}_{n+1+k+1} = (\mu', (\emptyset, b^n f^{2^{n-k-2}}, c^{(k+1)n})).$$

3. By applying (2) to the case $k = n - 1$, we obtain that $\mathcal{C}_{2n} = (\mu', (\emptyset, b^n f, c^{(n-1)n}))$.

   Then, the only applicability matrix over $\mathcal{C}_{2n}$ is $\vec{p} = (\mathbf{0}, (0, n, 0, 1), \mathbf{0})$. Hence, we have that the configuration $\mathcal{C}_{2n+1}$ is $(\mu'', (ab^n, c^{n^2}))$, where $\mu'' = (1, ((1, 4), (4, 1)))$.

4. From (3) we deduce that $\mathcal{C}_{2n+1} = (\mu'', (ab^n, c^{n^2}))$. Having in mind that $V(\mu'') = \{1, 4\}$ and $R_1 = R_4 = \emptyset$ we deduce that $\mathbf{M_{Ap}}(\mathcal{C}_{2n+1}) = \{(\mathbf{0}, \mathbf{0})\}$. Then the configuration $\mathcal{C}_{2n+1}$ is a halting one. Also, since $4 \in V(\mu'')$ and 4 is a leaf of μ'' it follows that the configuration $\mathcal{C}_{2n+1}$ is successful. Hence, the computation $\mathcal{C}$ is successful, its length is $2n + 1$, and its numerical output is $O(\mathcal{C}) = |\mathcal{C}_{2n+1}(4)| = n^2$.

□

As a first consequence of this proposition, let us see that after the moment when membrane 3 is dissolved, the P system evolves in a “deterministic” way.

Corollary 4.1. For every $n \geq 1$ and every $\mathcal{C}, \mathcal{C}' \in Comp(\Pi)$ such that $n = \delta(\mathcal{C}, 3) = \delta(\mathcal{C}', 3)$ we have that $\forall k \ (n \leq k \leq 2n + 1 \rightarrow \mathcal{C}_k = \mathcal{C}'_k)$.

Proof:
The case $k = n$ follows from (1) in the previous proposition, the case $n < k \leq 2n$ follows from (2), and the case $k = 2n + 1$ follows from (3). □

Next, let us see that if two computations have the same moment of dissolution of membrane 3, then these computations are equal.

Corollary 4.2. For every $n \geq 1$ and every $\mathcal{C}, \mathcal{C}' \in Comp(\Pi)$ such that $n = \delta(\mathcal{C}, 3) = \delta(\mathcal{C}', 3)$ we have that $\mathcal{C} = \mathcal{C}'$.

Proof:
Let $n \geq 1$ and $\mathcal{C}, \mathcal{C}' \in Comp(\Pi)$ such that $n = \delta(\mathcal{C}, 3) = \delta(\mathcal{C}', 3)$. By applying (4) in Proposition 4.2, and the prevoous corollary, it is sufficient to prove that $\forall\, k\ (0 \leq k \leq n-1 \rightarrow \mathcal{C}_k = \mathcal{C}'_k)$. But, this last relation follows directly from Proposition 4.1. □

Corollary 4.3. There exists at most one computation of Π which is not successful.

Proof:
Let $\mathcal{C}$ be a computation of Π which is not successful. From Proposition 4.2 we deduce that $\forall k\ (k < \delta(\mathcal{C}, 3))$. Hence, from Proposition 4.1 it follows that $\mathcal{C}_k = (\mu_0, (\emptyset, \emptyset, ab'^k f^{2^k}, \emptyset)))$. Then, $\mathcal{C}$ is unique. □

Next, let us see that the formula $\theta(\mathcal{C}, n)$ is true for every configuration $\mathcal{C}_n$ of the P system Π.

Corollary 4.4. The formula $\theta(\mathcal{C}, n)$ is an invariant of the P system Π. That is, $\forall \mathcal{C} \in Comp(\Pi)\ \forall n \in \mathbf{N}\ (\theta(\mathcal{C}, n))$.

Proof:
It follows directly from Proposition 4.1 and (4) in Proposition 4.2. □

Next, we are going to characterize the successful computations of Π by means of the moment when membrane 3 is dissolved.

Corollary 4.5. Let $\mathcal{C}$ be a computation of Π. The following assertions are equivalents:

(a) $\mathcal{C}$ is a successful computation.

(b) $\delta(\mathcal{C}, 3) < \infty$.

(c) $\delta(\mathcal{C}, 3) < \infty$ and $|\mathcal{C}| = 2 \cdot \delta(\mathcal{C}, 3) + 1$.

Proof:
Let $\mathcal{C}$ be a successful computation. Let $k = |\mathcal{C}|$. Then $1 \leq k < \infty$. Let us prove that $\delta(\mathcal{C}, 3) \leq k$. If this is not the case, then from Proposition 1 we have that $\mathcal{C}_k = (\mu_0, (\emptyset, \emptyset, ab'^k f^{2^k}, \emptyset))$, and this contradicts the equality $k = |\mathcal{C}|$, since from the existence of no null applicability matrix over $\mathcal{C}_k$ (for example, $\vec{p} = (\mathbf{0}, \mathbf{0}, (1, 0, 2^k), \mathbf{0}))$ we would have that $\mathcal{C}_k$ is not a halting configuration.

If $\delta(\mathcal{C}, 3) < \infty$ then, from (4) in Proposition 4.2, it follows that $|\mathcal{C}| = 2n + 1$. Finally, $(c) \Rightarrow (a)$ it follows directly from (4) in Proposition 4.2. □

5. Soundness and Completeness of the P System Π

To establish that the set of natural numbers generated by Π is $N(\Pi) = \{n^2 : n \geq 1\}$ we must prove two results:

- The numerical output of any successful computation of the P system Π encodes the square of a natural number greater than or equal to 1 (the *soundness* of the P system).
- For every $n \geq 1$ there exists at least one successful computation, $\mathcal{C}$, of the P system Π with the numerical output $O(\mathcal{C}) = n^2$ (the *completeness* of the P system).

Theorem 5.1. (**Soundness**) If $\mathcal{C}$ is a successful computation of the P system Π, then there exists $n \geq 1$ such that the output of $\mathcal{C}$ is $O(\mathcal{C}) = n^2$.

Proof:
Let $\mathcal{C}$ be a successful computation of Π. If $n = \delta(\mathcal{C}, 3)$, then from Corollary 4.2 it follows that $1 \leq n < \infty$. Since the formula $\theta(\mathcal{C}, n)$ is true and $n = \delta(\mathcal{C}, 3)$, we infer that the computation $\mathcal{C}$ is successful and, also, $O(\mathcal{C}) = n^2$. □

To establish the completeness of Π to generate the set $\{n^2 : n \geq 1\}$, we consider the formula $\varphi(n) \equiv \exists \mathcal{C} \in Comp(\Pi)\ (n = \delta(\mathcal{C}, 3))$. Let us see that this formula is true for every natural number greater than or equal to 1.

Proposition 5.1. For every natural number $n \geq 1$ there exists a unique computation, $\mathcal{C}$, of Π such that $\delta(\mathcal{C}, 3) = n$.

Proof:
We prove the existence by induction on n. For the base case, $n = 1$, the configuration $\mathcal{C}_1$, obtained from the initial configuration, $\mathcal{C}_0$, by applying the matrix $\vec{p} = (\mathbf{0}, \mathbf{0}, (0, 1, 1), \mathbf{0})$ (applicability matrix over $\mathcal{C}_0$) is considered. Since $r_2^3 \equiv a \to b'\delta$, we obtain that $\delta(\mathcal{C}, 3) = 1$.

Let $n \geq 1$ and let us suppose the result is true for n. Let $\mathcal{C}$ be a computation of Π such that $\delta(\mathcal{C}, 3) = n$. From Proposition 4.1, we deduce that $\mathcal{C}_{n-1} = (\mu_0, (\emptyset, \emptyset, ab'^{(n-1)} f^{2^{n-1}}, \emptyset)))$.

The set of applicability matrices over $\mathcal{C}_{n-1}$ is $\mathbf{M_{Ap}}(\mathcal{C}_{n-1}) = \{\vec{p}_1, \vec{p}_2\}$, where

$$\vec{p}_1 = (\mathbf{0}, \mathbf{0}, (0, 1, 2^{n-1}), \mathbf{0}), \vec{p}_2 = (\mathbf{0}, \mathbf{0}, (1, 0, 2^{n-1}), \mathbf{0})$$

Let $\mathcal{C}'_n = \vec{p}_2(\mathcal{C}_{n-1})$. Then $\mathcal{C}'_n = (\mu_0, (\emptyset, \emptyset, ab'^n f^{2^n}, \emptyset))$. Let $\mathcal{C}'_{n+1} = \vec{p}_3(\mathcal{C}'_n)$, where $\vec{p}_3 = (\mathbf{0}, \mathbf{0}, (0, 1, 2^n), \mathbf{0})$, and in this step membrane 3 is dissolved. We have $\mathcal{C}'_{n+1} = (\mu', (\emptyset, b'^{(n+1)} f^{2^{n+1}}, \emptyset))$, where the membrane structure is $\mu' = (1, ((1, 2), (2, 1, 4), (4, 2)))$. Hence, the computation $\mathcal{C}' \equiv \mathcal{C}_0 \Rightarrow_\Pi \mathcal{C}_1 \Rightarrow_\Pi \ldots \Rightarrow_\Pi \mathcal{C}_{n-1} \Rightarrow_\Pi \mathcal{C}'_n \Rightarrow_\Pi \mathcal{C}'_{n+1} \Rightarrow_\Pi \ldots$, verifies that $\delta(\mathcal{C}', 3) = n + 1$.

Given $n \geq 1$, the uniqueness of the computation $\mathcal{C}$ verifying $\delta(\mathcal{C}, 3) = n$ follows directly from Corollary 4.2. □

Proposition 5.2. There exists an unique computation, $\mathcal{C}$, of Π which is not successful.

Proof:
First, let us prove that such a computation exists. For every $k \in \mathbf{N}$, let us consider the configuration $\mathcal{C}_k = (\mu_0, (\emptyset, \emptyset, ab'^k f^{2^k}, \emptyset)))$. If we take the matrix $\vec{p} = (\mathbf{0}, \mathbf{0}, (1, 0, 2^k), \mathbf{0})$, then we have that $\forall\, k\ (\mathcal{C}_{k+1} = \vec{p}(\mathcal{C}_k))$. Then $\mathcal{C} \equiv \mathcal{C}_0 \Rightarrow_\Pi \mathcal{C}_1 \Rightarrow_\Pi \ldots \mathcal{C}_k \Rightarrow_\Pi \mathcal{C}_{k+1} \Rightarrow_\Pi \ldots$ is a computation of Π. Also, from the construction, it is obvious that $\mathcal{C}$ is not a halting computation.

The uniqueness of such a computation follows from Corollary 4.3. □

Corollary 5.1. For every $n \geq 1$ the formula $\varphi(n)$ is true.

Corollary 5.2. The partial function $D_3 : Comp(\Pi) \to \mathbf{N} - \{0\}$, defined by $D_3(\mathcal{C}) = \delta(\mathcal{C}, 3)$ is a bijection from the set $S(\Pi)$ of successful configurations of Π to the set $\mathbf{N} - \{0\}$.

Note: $D_1 = D_4 = \emptyset$, and D_2 is not bijective.

Theorem 5.2. (**Completeness**) For every natural number $n \geq 1$ there exists a successful computation, $\mathcal{C}$, of the P system Π, such that its numerical output is $O(\mathcal{C}) = n^2$.

Proof:
Let $n \in \mathbf{N}$ such that $n \geq 1$. Since the formula $\varphi(n)$ is true, there exists a computation $\mathcal{C}$ of Π such that $\delta(\mathcal{C}, 3) = n$. Having in mind that the formula $\theta(\mathcal{C}, n)$ is true, we conclude that the computation $\mathcal{C}$ is successful, and, also, $O(\mathcal{C}) = n^2$. □

6. Conclusions

The formal verification of a computing model is usually a hard task. If the procedures in the model are not defined through an imperative language, then this task is harder. This is the case of P systems, that, basically, is a procedural computing model.

The formal verification of a P system is based on the characterization of its successful computations, and 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 truth of such formula in every configuration must give important information to characterize the successful computations.

In this paper the formal verification of a P system given by Păun ([1]) to generate squares of natural numbers greater than or equal to 1 has been obtained. The process of verification is based on the analysis of a *critical point* appearing in every halting configuration: the moment when a relevant membrane is dissolved. Moreover, in this work a detailed study of *every* computations of the P system is given, and a classification of these computations is obtained. The formalization and study of the verification of P systems may represent an important step to the treatment of them through reasoning systems.

References

[1] Gh. Păun, 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] Gh. Păun, G. Rozenberg, A guide to membrane computing, *Theoretical Computer Science*, to appear.

[3] M.J. Pérez–Jiménez, F. Sancho-Caparrini. A formalization of transition P systems. *Fundamenta Informaticae*. to appear.