Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. III (2008), No. 3, pp. 295-303

Tissue P Systems with Cell Division

Gheorghe Păun, Mario J. Pérez-Jiménez, Agustín Riscos-Núñez

Abstract: In tissue P systems several cells (elementary membranes) communicate through symport/antiport rules, thus carrying out a computation. We add to such systems the basic feature of (cell–like) P systems with active membranes – the possibility to divide cells. As expected (as it is the case for P systems with active membranes), in this way we get the possibility to solve computationally hard problems in polynomial time; we illustrate this possibility with SAT problem.
Keywords: Tissue-like P systems, cell division rule, SAT problem, NP-complete problem

1 Introduction

In membrane computing, there are two main classes of P systems: with the membranes arranged hierarchically, inspired from the structure of the cell, and with the membranes placed in the nodes of a graph, all of them at the same level, inspired from the cell inter-communication in tissues. A particularly interesting sub-class of the first class are the systems with active membranes, where the membrane division can be used in order to solve hard problems, e.g., **NP**-complete problems, in polynomial or even linear time, by a space-time trade-off. In the tissue P systems, the communication among cells is performed by means of symport/antiport rules, well-known in biology. Details can be found in [8], [10], as well as in the comprehensive page from the web address `http://ppage.psystems.eu`).

In this paper we combine the two definitions, and consider tissue P systems (with the communication done through symport/antiport rules) with cell division rules of the same form as in P systems with active membranes, but without using polarizations. The rules are used in the non-deterministic maximally parallel way, with the restriction that if a division rule is used for dividing a cell, then this cell does not participate in any other rule, for division or communication (the intuition is that when dividing, the interaction of the cell with other cells or with the environment is blocked); the cells obtained by division have the same labels as the mother cell, hence the rules to be used for evolving them or their objects are inherited (the label precisely identifies the available rules).

This natural extension of tissue P systems provides the possibility of solving SAT problem in polynomial time (with respect to the number of variables and of clauses), in a confluent way: at precise times, one of the objects `yes` or `no` is sent to the environment, giving the answer to the question whether the input propositional formula is satisfiable. The construction is uniform: in a polynomial time, a family of recognizing tissue P systems with cell division is constructed, which, receiving as inputs encodings of instances of SAT, tells us whether or not these instances are satisfiable.

2 Tissue P Systems with Cell Division

We assume the reader to be familiar with basic elements of membrane computing and we directly define the class of P systems which is investigated in this paper.

A *tissue P system with cell division* of degree $m \geq 1$ is a construct

$$\Pi = (O, E, w_1, \ldots, w_m, R, i_o),$$

where:

1. $m \geq 1$ (the initial degree of the system; the system contains m cells, labeled with $1,2,\ldots,m$; we will use 0 to refer to the environment);
2. O is the alphabet of *objects*;
3. $w_1,\ldots,w_m$ are strings over O, describing the *multisets of objects* placed in the m cells of the system at the beginning of the computation;
4. $E \subseteq O$ is the set of objects present in the environment in arbitrarily many copies each;
5. R is a finite set of *evolution rules*, of the following forms:
   (a) $(i,x/y,j)$, for $i,j \in \{0,1,2,\ldots,m\}, i \neq j$, and $x,y \in O^*$;
   communication rules; $1,2,\ldots,m$ identify the cells of the system, 0 is the environment; when applying a rule $(i,x/y,j)$, the objects of the multiset represented by x are sent from region i to region j and simultaneously the objects of the multiset y are sent from region j to region i;
   (b) $[\,a\,]_i \rightarrow [\,b\,]_i[\,c\,]_i$, where $i \in \{1,2,\ldots,m\}$ and $a,b,c \in O$;
   division rules; under the influence of object a, the cell with label i is divided in two cells with the same label; in the first copy the object a is replaced by b, in the second copy the object a is replaced by c; all other objects are replicated and copies of them are placed in the two new cells.
6. $i_o \in \{1,2,\ldots,m\}$ is the output cell.

Therefore, we use antiport rules for communication (for a rule $(i,x/y,j)$ we say that the maximum of the lengths of x and y is the *weight* of the rule), and division rules as in P systems with active membranes.

The rules of a system as above are used in the non-deterministic maximally parallel manner as customary in membrane computing. In each step, all cells which can evolve must evolve in a maximally parallel way (in each step we apply a multiset of rules which is maximal, no further rule can be added), with the following important mentioning: if a cell is divided, then the division rule is the only one which is applied for that cell in that step, its objects do not evolve by means of communication rules. This is like saying that a cell which divides first cuts all its communication channels with the other cells and with the environment; the dotter cells will participate to the interaction with other cells or with the environment only in the next step – providing that they are not divided once again. Their label precisely identify the rules which can be applied to them.

The computation starts from the initial configuration and proceeds as defined above; only halting computations give a result, and the result is the number of objects present in the halting configuration in cell i_o; the set of numbers computed in this way by the various halting computations in Π is denoted by $N(\Pi)$.

In the present paper we are not interested in the computing power of systems as above – already systems without membrane division are known to be Turing complete (see [8], [6], etc.), but in their computing efficiency. That is why we introduce a variant of tissue P systems with membrane division, namely *recognizing systems with input* following the definitions of complexity classes in terms of membrane computing (see [9]). Such a system has the form $\Pi = (O,\Sigma,E,w_1,\ldots,w_m,R,i_{in})$, where:

- $(O,E,w_1,\ldots,w_m,R,0)$ is a tissue P system with cell division of initial degree $m \geq 1$ (as defined in the previous section, but with the environment, indicated by taking $i_o = 0$, used for reading the output of a computation), and $w_1,\ldots,w_m$ are strings over $O - \Sigma$.
- The working alphabet O has two distinguished objects `yes` and `no`, present in at least one copy in some initial multisets $w_1, \ldots, w_m$, but not present in E.
- Σ is an (input) alphabet strictly contained in O.
- $i_{in} \in \{1,\ldots,m\}$ is the input cell.

- All computations halt.
- If $\mathcal{C}$ is a computation of Π, then either the object `yes` or the object `no` (but not both) must have been released into the environment, and only in the last step of the computation.

The computations of the system Π with input $w \in \Sigma^*$ start from a configuration of the form $(w_1, w_2, \ldots, w_{i_{in}} w, \ldots, w_m; E)$, that is, after adding the multiset w to the contents of the input cell i_{in}. We say that the multiset w is *recognized* by Π if and only if the object `yes` is sent to the environment, in the last step of the corresponding computation. We say that $\mathcal{C}$ is an accepting computation (respectively, rejecting computation) if the object `yes` (respectively, `no`) appears in the environment associated with the corresponding halting configuration of $\mathcal{C}$.

Definition 1. We say that a decision problem $X = (I_X, \theta_X)$ is solvable in polynomial time by a family $\Pi = \{\Pi(n) \mid n \in \mathbf{N}\}$ of recognizer tissue-like P systems with cell division if the following holds:

- The family Π 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 $n \in \mathbf{N}$.
- There exists a pair (cod, s) of polynomial-time computable functions over I_X (called a polynomial encoding of I_X in Π) 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 $\Pi(s(u))$;
  - the family Π 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 $\Pi(s(u))$ with input $cod(u)$ is halting and, moreover, it performs at most $p(|u|)$ steps;
  - the family Π is *sound* with regard to (X, cod, s), that is, for each $u \in I_X$, if there exists an accepting computation of $\Pi(s(u))$ with input $cod(u)$, then $\theta_X(u) = 1$;
  - the family Π 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 $\Pi(s(u))$ with input $cod(u)$ is an accepting one.

We denote by $\mathbf{PMC}_{TD}$ the set of all decision problems which can be solved by means of recognizer tissue-like P systems with cell division in polynomial time. This class is closed under polynomial–time reduction and under complement.

We close this section with an important remark about the previous way of solving decision problems. Specifically, we have said nothing about the way the computations proceed; in particular, they can be non-deterministic, as standard in membrane computing. It is important however that the systems always stop and always they send out an object which is the correct answer to the input problem. From the soundness and completeness conditions above we deduce that every P system $\Pi(n)$ is *confluent*, in the following sense: every computation of a system with the *same* input multiset must always give the *same* answer.

3 Solving `SAT` in Polynomial Time

As expected, the possibility to divide cells means the possibility to create an exponential space in a linear time, and this space can be used in order to obtain fast solutions to computationally hard problems.

Theorem 1. *Tissue P systems with active membranes can solve* `SAT` *in polynomial time. (Otherwise stated,* $\texttt{SAT} \in \mathbf{PMC}_{TD}$*.)*

Proof. Let us consider a propositional formula $\gamma = C_1 \wedge \cdots \wedge C_m$, consisting of m clauses $C_j = y_{j,1} \vee \cdots \vee y_{j,k_j}$, $1 \leq j \leq m$, where $y_{j,i} \in \{x_l, \neg x_l \mid 1 \leq l \leq n\}$, $1 \leq i \leq k_j$ (there are used n variables). Without loss of generality, we may assume that no clause contains two occurrences of some x_i or two occurrences of some $\neg x_i$ (the formula is not redundant at the level of clauses), or both x_i and $\neg x_i$ (otherwise such a clause is trivially satisfiable, hence can be removed).

We codify γ, which is an instance of SAT with size parameters n and m, by the multiset

$$\begin{aligned} cod(\gamma) &= \{s_{i,j} \mid y_{j,r} = x_i,\ 1 \leq i \leq n, 1 \leq j \leq m, 1 \leq r \leq k_j\} \\ &\cup \{s'_{i,j} \mid y_{j,r} = \neg x_i,\ 1 \leq i \leq n, 1 \leq j \leq m, 1 \leq r \leq k_j\}. \end{aligned}$$

(We replace each variable x_i from each clause C_j with $s_{i,j}$ and each negated variable $\neg x_i$ from each clause C_j with $s'_{i,j}$, then we remove all parentheses and connectives. In this way we pass from γ to $cod(\gamma)$ in a number of steps which is linear with respect to $n \cdot m$.)

The instance γ will be processed by the tissue P system $\Pi(s(\gamma))$ with input $cod(\gamma)$, where $s(\gamma) = \langle n, m \rangle = \frac{(n+m)\cdot(n+m+1)}{2} + n$. We construct the recognizing tissue P system (of degree 2) with input

$$\Pi(\langle n, m \rangle) = (O, \Sigma, E, w_1, w_2, R, 2),$$

with the following components:

$$\begin{aligned} O &= \Sigma \cup \{a_i, t_i, f_i \mid 1 \leq i \leq n\} \cup \{r_i \mid 1 \leq i \leq m\} \\ &\cup \{T_i, F_i \mid 1 \leq i \leq n\} \cup \{T_{i,j}, F_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq m+1\} \\ &\cup \{b_i \mid 1 \leq i \leq 3n+m+1\} \cup \{c_i \mid 1 \leq i \leq n+1\} \\ &\cup \{d_i \mid 1 \leq i \leq 3n+nm+m+2\} \cup \{e_i \mid 1 \leq i \leq 3n+nm+m+4\} \\ &\cup \{f, g, \texttt{yes}, \texttt{no}\}, \\ \Sigma &= \{s_{i,j}, s'_{ij} \mid 1 \leq i \leq n,\ 1 \leq j \leq m\}, \\ E &= O - \{\texttt{yes}, \texttt{no}\}, \\ w_1 &= \texttt{yes no } b_1\ c_1\ d_1\ e_1, \\ w_2 &= f\ g\ a_1\ a_2\ \ldots\ a_n, \end{aligned}$$

and the following rules.

1. **Division rules:**

   $[\, a_i \,]_2 \rightarrow [\, T_i \,]_2 [\, F_i \,]_2$, for all $i = 1, 2, \ldots, n$.

   (Membrane 2 is repeatedly divided, each time expanding one object a_i, corresponding to a variable x_i, into T_i and F_i, corresponding to the values *true* and *false* which this variable may assume. In this way, in n steps, we get 2^n cells with label 2, each one containing one of the 2^n truth-assignments possible for the n variables. The objects f, g are duplicated, hence a copy of each of them will appear in each cell.)

2. **Communication rules:**

   $(1, b_i / b^2_{i+1}, 0)$, for all $i = 1, 2, \ldots, n+1$,
   $(1, c_i / c^2_{i+1}, 0)$, for all $i = 1, 2, \ldots, n+1$,
   $(1, d_i / d^2_{i+1}, 0)$, for all $i = 1, 2, \ldots, n+1$,
   $(1, e_i / e_{i+1}, 0)$, for all $i = 1, 2, \ldots, 3n+nm+m+3$.

   (In parallel with the operation of dividing cell 2, the counters b_i, c_i, d_i, e_i from cell 1 grow their subscripts. In each step, the number of copies of objects of the first three types is doubled, hence

after n steps we get 2^n copies of b_{n+1}, c_{n+1}, and d_{n+1}. Objects b_i will check which clauses are satisfied by a given truth-assignment, objects c_i are used in order to multiply the number of copies of t_i, f_i as we will see immediately, d_i are used to check whether there is at least one truth-assignment which satisfies all clauses, and e_i will be used in order to produce the object `no`, if this will be the case, in the end of the computation.)

$(1, b_{n+1}c_{n+1}/f, 2)$,
$(1, d_{n+1}/g, 2)$.

(In step $n+1$, the counters $b_{n+1}, c_{n+1}, d_{n+1}$ are brought in cells with label 2, in exchange of f and g. Because we have 2^n copies of each object of these types and 2^n cells 2, each one containing exactly one copy of f and one of g, due to the maximality of the parallelism of using the rules, each cell 2 gets precisely one copy of each of $b_{n+1}, c_{n+1}, d_{n+1}$. Note that cells 2 cannot divide any more, because the objects a_i were exhausted.)

$(2, c_{n+1}T_i/c_{n+1}T_{i,1}, 0)$,
$(2, c_{n+1}F_i/c_{n+1}F_{i,1}, 0)$, for each $i = 1,2,\ldots,n$,
$(2, T_{i,j}/t_iT_{i,j+1}, 0)$,
$(2, F_{i,j}/f_iF_{i,j+1}, 0)$, for each $i = 1,2,\ldots,n$ and $j = 1,2,\ldots,m$.

(In the presence of c_{n+1}, the objects T_i, F_i introduce the objects $T_{i,1}$ and $F_{i,1}$, respectively, which initiates the possibility of introducing m copies of each t_i and f_i in each cell 2. The idea is that because we have m clauses, in order to check their values for a given truth-assignment of variables, it is possible to need one value for each variable for each clause. Note that this phase needs $2n$ steps for introducing the double-subscripted objects $T_{i,1}, F_{i,1}$ – for each one we need one step, because we have only one copy of c_{n+1} available – then further m steps are necessary for each $T_{i,1}, F_{i,1}$ to grow its second subscript; all these steps are done in parallel, but for the last introduced $T_{i,1}, F_{i,1}$ we have to continue m steps after the $2n$ necessary for priming. In total, we perform $2n+m$ steps.)

$(2, b_i/b_{i+1}, 0)$,
$(2, d_i/d_{i+1}, 0)$, for all $i = n+1,\ldots,(n+1)+(2n+m)-1$.

(In parallel with the previous operations, the counters b_i and d_i increase their subscripts, until reaching the value $3n+m+1$. This is done in all cells 2 at the same time. Simultaneously, e_i increases its subscript in cell 1.)

$(2, b_{3n+m+1}t_is_{i,j}/b_{3n+m+1}r_j, 0)$,
$(2, b_{3n+m+1}f_is'_{i,j}/b_{3n+m+1}r_j, 0)$, for all $1 \leq i \leq n$ and $1 \leq j \leq m$,
$(2, d_i/d_{i+1}, 0)$, for all $i = 3n+m+1,\ldots,(3n+m+1)+nm-1$.

(In the presence of b_{3n+m+1} – and not before – we check the values assumed by clauses for the truth-assignments from each cell 2. We have only one copy of b_{3n+m+1} in each cell, hence we need at most nm steps for this: each clause contains at most n literals, and we have m clauses. In parallel, d increases the subscript, until reaching the value $3n+nm+m+1$.)

$(2, d_{3n+nm+m+i}r_i/d_{3n+nm+m+i+1}, 0)$, for all $i = 1,2,\ldots,m$.

(In each cell with label 2 we check whether or not all clauses are satisfied by the corresponding truth-assignment. For each clause which is satisfied, we increase by one the subscript of d, hence the subscript reaches the value $3n+nm+2m+1$ if and only if all clauses are satisfied.)

$(2, d_{3n+nm+2m+1}/f\ \texttt{yes}, 1)$.

(If one of the truth-assignments from a cell 2 has satisfied all clauses, then we reach $d_{3n+nm+2m+1}$, which is sent to cell 1 in exchange of the objects `yes` and f.)

$(2, \texttt{yes}/\lambda, 0)$.

(In the next step, the object `yes` leaves the system, signaling the fact that the formula is satisfiable. In cell 1, the counter e will increase one more step its subscript, but after that it will remain unchanged – it can leave cell 1 only in the presence of f, but this object was already moved to cell 2.)

$(1, e_{3n+nm+2m+2} f\ \texttt{no}/\lambda, 2)$,
$(2, \texttt{no}/\lambda, 0)$.

(If the counter e reaches the subscript $3n+nm+2m+2$ and the object f is still in cell 1, then the object `no` can be moved to a cell 2, randomly chosen, and from here it exits the system, signaling that the formula is not satisfiable.)

In order to show that the family $\Pi = \{\Pi(\langle n,m\rangle) \mid n,m \in \mathbf{N}\}$ is polynomially uniform by deterministic Turing machines we first note that the sets of rules associated with the system $\Pi(\langle n,m\rangle)$ are recursive. Hence, it is enough to note that the amount of necessary resources for defining each system is quadratic in $\max\{n,m\}$, and this is indeed the case, since those resources are the following:

1. Size of the alphabet: $6nm+17n+4m+12 \in \Theta(nm)$.
2. Initial number of cells: $2 \in \Theta(1)$.
3. Initial number of objects: $n+8 \in \Theta(n)$.
4. Number of rules: $4nm+10n+3m+16 \in \Theta(nm)$.
5. Upper bound for the length of the rules: $3 \in \Theta(1)$

From the previous explanations, one can see that, starting with the multiset $cod(\gamma)$ added to cell 2, which is the input cell, the system correctly answers the question whether or not γ is satisfiable. The duration of the computation is polynomial in terms of n and m: the answer `yes` is sent out in step $3n+nm+2m+2$, while the answer `no` is sent out in step $3n+nm+2m+4$. This concludes the proof. □

The antiport rules from the previous construction are of weight at most 3, but the weight can be reduced to two, at the expense of some slowdown of the system. For instance, instead of the rule $(1, e_{3n+nm+2m+2} f\ \texttt{no}/\lambda, 2)$ we can consider the rules $(1, e_{3n+nm+2m+2} f/h, 0)$, $(1, h\ \texttt{no}/\lambda, 2)$, where h is a new object. We can proceed in the same way with the rules $(2, b_{3n+m+1} t_i s_{i,j}/b_{3n+m+1} r_j, 0)$, $(2, b_{3n+m+1} f_i s'_{i,j}/b_{3n+m+1} r_j, 0)$, for $1 \leq i \leq n$ and $1 \leq j \leq m$, but in this way instead of at most nm steps for finding the satisfied clauses we will need at most $2nm$ steps. The details are left to the reader.

Taking into account that `SAT` is an **NP**–complete problem and the class $\mathbf{PMC}_{TD}$ is closed under polynomial–time reduction and under complement, we have:

Corollary 2. $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{TD}$

4 Final Remarks

We have proven that by adding the membrane division feature to tissue P systems (with the communication done by antiport rules of a small weight) we can solve **NP**-complete problems in polynomial time. We exemplify this possibility with `SAT` problem.

It remains as a research topic to consider the same extension for other types of systems, for instance, for cell P systems with symport/antiport rules, or for neural P systems (with states associated with cells and multiset rewriting rules for processing the objects. The difficulty in the case of cell P

systems with symport/antiport comes from the fact that only the skin membrane can communicate with the environment; on the other hand, the skin membrane cannot be divided, hence we need exponentially many objects for communication with inner membranes, and such objects should be brought in from the environment. In turn, neural P systems with the maximal use of rules and replicated communication are already known to be able to solve **NP**-complete problems in polynomial time; the challenge now is not to use replication). In spite of these difficulties, we expect results similar to the above one also in these cases.

Another problem which remains open is to consider tissue P systems with the communication using only symport rules.

A previous version of the present paper was circulated in the volume of the Second Brainstorming Week on Membrane Computing, held in Sevilla, in February 2004, and in the meantime several papers have considered tissue-like P systems with cell division as a framework for devising polynomial solutions to **NP**-complete problems. For instance, [2] deals with the Subset Sum problem, [3] deals with the Partition problem, [5] deals with the Vertex Cover problem, and [4] considers the 3–coloring problem. What is not yet investigated is the possibility to also solve **PSPACE** problems, as it is the case, for instance, for cell-like P systems with division of non-elementary membranes (see [11]) or with membrane creation (see [7]). This last possibility for producing working space, cell creation rules, has been only recently considered for tissue P systems [1]. Let us recall that this kind of rules does not perform replication of objects, as it happens with cell-division rules, and it is an open question whether tissue P systems with communication and membrane creation rules can solve efficiently computationally hard problems.

Acknowledgements. The support of this research through the project TIN2006-13425 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the project of excellence TIC-581 of the Junta de Andalucía, is gratefully acknowledged.

Bibliography

[1] D. Díaz–Pernil: *Sistemas celulares de tejidos: Formalización y eficiencia computacional*, PhD Thesis, University of Sevilla, 2008.

[2] D. Díaz–Pernil, M.A. Gutiérrez–Naranjo, M.J. Pérez–Jiménez, A. Riscos–Núñez: Solving subset sum in linear time by using tissue P systems with cell division. In J. Mira, J.R. Alvarez, eds. *Bio-Inspired Modeling of Cognitive Tasks, Second International Work–Conference on the Interplay between Natural and Artificial Computation, IWINAC 2007*, La Manga del Mar Menor, Spain, June 2007,Part I, LNCS 4527, Springer, 2007, 170–179.

[3] D. Díaz–Pernil, M.A. Gutiérrez–Naranjo, M.J. Pérez–Jiménez, A. Riscos–Núñez: Solving the partition problem by using tissue-like P systems with cell division. In D. Díaz, C. Graciani, M.A. Gutiérrez, Gh. Păun, I. Pérez–Hurtado, A. Riscos, eds., *Proceedings of the Sixth Brainstorming Week on Membrane Computing*, Report RGNC 01/08, Fénix Editora, 2008, 123–134.

[4] D. Díaz–Pernil, M.A. Gutiérrez–Naranjo, M.J. Pérez–Jiménez, A. Riscos–Núñez: A linear-time tissue P system based solution for the 3-coloring problem, *Electronic Notes in Theoretical Computer Science*, 171 (2007), 81–93.

[5] D. Díaz–Pernil, M.J. Pérez–Jiménez, A. Riscos–Núñez, A. Romero–Jiménez: Computational efficiency of cellular division in tissue-like membrane systems, submitted, 2008.

[6] P. Frisco, H.J. Hoogeboom: Simulating counter automata by P systems with symport/antiport. In Gh. Păun, G. Rozenberg, A. Salomaa, C. Zandron, eds., *Membrane Computing. International Workshop WMC 2002, Curtea de Argeş, Romania, Revised Papers*, LNCS 2597, Springer, 2003, 288–301.

[7] M.A. Gutiérrez–Naranjo, M.J. Pérez–Jiménez, F.J. Romero–Campero: A linear time solution for QSAT with membrane creation. In R. Freund, Gh. Păun, G. Rozenberg, A. Salomaa, eds., *Membrane Computing, 6th International Workshop, WMC 2005, Vienna, Austria, July 18-21, 2005, Revised Selected and Invited Papers*, LNCS 3850, Springer, 2006, 241–252.

[8] Gh. Păun: *Computing with Membranes: An Introduction*, Springer, Berlin, 2002.

[9] M.J. Pérez–Jiménez: An approach to computational complexity in membrane computing. In G. Mauri, Gh. Păun, M.J. Pérez-Jiménez, G. Rozenberg, A. Salomaa, eds., *Membrane Computing, 5th International Workshop, WMC5, Revised Selected and Invited Papers*, LNCS 3365, Springer, 2005, 85–109.

[10] M. Pérez–Jiménez, A. Romero–Jiménez, F. Sancho–Caparrini, *Teoría de la Complejidad en Modelos de Computatión Celular con Membranas*, Editorial Kronos, Sevilla, 2002.

[11] P. Sosik, A. Rodríguez-Patón: Membrane computing and complexity theory: A characterization of PSPACE, *International Journal of Foundations of Computer Science*, 73, 1 (2007), 137–152.

Gheorghe PĂUN[1,2], Mario J. PÉREZ-JIMÉNEZ[2],
Agustín RISCOS-NÚÑEZ[2]
[1]Institute of Mathematics of the Romanian Academy
PO Box 1-764, 014700 Bucureşti, Romania

[2] Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
Technical Higher School of Computer Science Engineering
University of Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: {gpaun, marper, ariscosn}@us.es

Mario J., Pérez–Jiménez received his degree in mathematics from the Barcelona University and doctor degree in mathematics from the Sevilla University. In the past, he was associated professor at the University of Barcelona. Currently, he is titular professor of Computer Science and Artificial Intelligence at the University of Sevilla, where is the head of the Research Group on Natural Computing. His main research fields are computational complexity theory, unconventional models of computation, natural computing, membrane computing, bioinformatics, and computational modelling for systems biology. He has published twelve books in computer science and mathematics, and over 100 scientific papers in international journals (collaborating with many researchers worldwide). He is the main researcher in various European, Spanish and Andalusia research grants. He has been and independent expert to the evaluation of NEST (New and Emergent Science and Technology) proposals under the Sixth Framework Programme of the European Community, and from May 2006 he is an European Science Foundation peer reviewer.

Gheorghe Păun, graduated the Faculty of Mathematics, University of Bucharest, in 1974 and received his Ph.D. in mathematics (with specialization in computer science) from the same university in 1977. He held a research position at the University of Bucharest, and from 1990 he is at the Institute of Mathematics of the Romanian Academy, where he is currently a senior researcher. He visited numerous universities in Europe, Asia, and North America, with frequent and/or longer stays in Turku (Finland), Leiden (The Netherlands), Magdeburg (Germany, including an Alexander von Humboldt fellowship, in 1992-93), Tarragona, Madrid, and Sevilla (Spain, including a Ramon y Cajal scholarship, in 2001–2006), London-Ontario (Canada), Rome, Milano, Pisa (Italy), Tokyo (Japan), Warsaw (Poland), Vienna (Austria), Budapest (Hungary), etc.

His main research areas are formal language theory and its applications, computational linguistics, DNA computing, and membrane computing; this last research area was initiated by him, in 1998, and the respective models are now called P systems, see `http://ppage.psystems.eu`).

He has published a large number of research papers (collaborating with many researchers worldwide), has lectured at over 100 universities, and gave numerous invited talks at recognized international conferences. He has published eleven monographs in mathematics and computer science (some of them translated in Japanese, Chinese, Russian), has (co)edited over seventy collective volumes and special issues of journals, and also published many popular science books, books on recreational mathematics (games), and fiction books.

He is a member of the editorial board of more than a dozen international journals and was/is involved in the program/steering/organizing committees for many recognized conferences and workshops.

In 1997 he was elected a member of the Romanian Academy and from 2006 he is a member of Academia Europaea. He also got other honors, in Romania or abroad (professional and literary prizes, honorary citizenship titles, doctor honoris causa of the Silesian University in Opava, Czech Republic, etc.).