(Tissue) P Systems Working in the k-Restricted Minimally Parallel Derivation Mode

Rudolf Freund[1]
Sergey Verlan[2]
[1]Faculty of Informatics, Vienna University of Technology
Favoritenstr. 9, 1040 Vienna, Austria
Email: rudi@emcc.at
[2]LACL, Département Informatique
UFR Sciences et Technologie, Université Paris XII
61, av. Général de Gaulle, 94010 Créteil, France
Email: verlan@univ-paris12.fr

Abstract

Just recently several new derivaton modes for (tissue) P systems have been introduced and investigated in addition to the maximally parallel derivation mode used from the beginning in the area of membrane computing. A variant of the minimally parallel derivation is considered in this paper – we allow only a bounded number of rules to be taken from every set of the partitioning of the whole set of rules. The 1-restricted minimally parallel derivation mode especially fits to describe the way derivations take place in spiking neural P systems without delays, i.e., in every neuron where a rule is applicable exactly one rule has to be applied. Moreover, purely catalytic P systems working in the maximally parallel derivation mode can be described as P systems using the corresponding rules without catalysts when working in the 1-restricted minimally parallel derivation mode.

1 Introduction

In the original model of P systems introduced as membrane systems by Gh. Păun (see [5], [10]), the objects evolve in a hierarchical membrane structure; in tissue P systems, first considered by Gh. Păun and T. Yokomori in [13] and [14], also see [7], the cells communicate within an arbitrary graph topology. In the original model of membrane systems as well as in many variants of P systems and tissue P systems investigated during the last decade, the maximally parallel derivation mode was used. Just recently several new derivation modes for P systems and tissue P systems have been introduced and investigated, for example, the sequential and the asynchronous derivation mode as well as the minimally parallel derivation mode (see [3]). In [8], a formal framework for (tissue) P systems capturing the formal features of these derivation modes was developed, based on a general model of membrane systems as a collection of interacting cells containing multisets of objects (compare with the models of networks of cells as discussed in [2] and networks of language processors as considered in [4]). Continuing the formal approach started in [8], a variant of the minimally parallel derivation mode is considered in this paper: In the minimally parallel derivation mode, we consider a partitioning of the whole set of rules and allow only multisets of rules to be applied in parallel which cannot be extended by adding a rule from a set of rules from which no rule has already been taken into this multiset of rules. Whereas in the minimally parallel derivation mode, an arbitrary number of rules can be used from any of the sets of rules in the partitioning of rules, only a bounded number of at most k rules can be taken from each of these sets of the partitioning in the k-restricted minimally parallel derivation mode.

The rest of this paper now is organized as follows: In the second section, well-known definitions and notions are recalled. In the next section, we consider a general class of multiset rewriting systems containing, in particular, many variants of P systems and tissue P systems as well as even (extended) spiking neural P systems without delays and give formal definitions of the most important well-known derivation modes (maximally parallel, minimally parallel) as well as the new k-restricted minimally parallel derivation mode. The 1-restricted minimally parallel derivation mode especially fits to describe the way derivations take place in spiking neural P systems without delays, i.e., in every neuron where a rule is applicable exactly one rule has to be applied. Moreover, purely catalytic P systems working in the maximally parallel derivation mode can be described as P systems using the corresponding rules without catalysts when working in the 1-restricted minimally parallel derivation mode. An outlook to future research topics involving the k-restricted minimally parallel derivation mode concludes the paper.

2 Preliminaries

We recall some of the notions and the notations we use (see [12] for elements of formal language theory). Let V be a (finite) alphabet; then V^* is the set of all strings (a language) over V, and $V^+ = V^* - \{\lambda\}$ where λ denotes the empty string. RE, REG ($RE\,(T)$, $REG\,(T)$) denote the families of recursively enumerable and regular languages (over the alphabet T), respectively. For any family of string languages F, PsF denotes the family of Parikh sets of languages from F and NF the family of Parikh sets of languages from F over a one-letter alphabet. By $\mathbb{N}$ we denote the set of all non-negative integers, by $\mathbb{N}^k$ the set of all vectors of non-negative integers; $[k..m]$ for $k \leq m$ denotes the set of natural numbers n with $k \leq n \leq m$. In the following, we will not distinguish between NRE, which coincides with $PsRE\,(\{a\})$, and $RE\,(\{a\})$.

Let V be a (finite) set, $V = \{a_1, ..., a_k\}$. A *finite multiset* M over V is a mapping $M : V \longrightarrow \mathbb{N}$, i.e., for each $a \in V$, $M(a)$ specifies the number of occurrences of a in M. The size of the multiset M is $|M| = \sum_{a\in V} M(a)$. A multiset M over V can also be represented by any string x that contains exactly $M(a_i)$ symbols a_i for all $1 \leq i \leq k$, e.g., by $a_1^{M(a_1)}...a_k^{M(a_k)}$, or else by the set $\left\{a_i^{M(a_i)} \mid 1 \leq i \leq k\right\}$. The support of M is the set $supp(M) = \{a \in V \mid f(a) \geq 1\}$.

The set of all finite multisets over the set V is denoted by $\langle V, \mathbb{N}\rangle$. We may also consider mappings M of the form $M : V \longrightarrow \mathbb{N}_\infty$ where $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$, i.e., elements of M may have an infinite multiplicity; we shall call such multisets where $M(a_i) = \infty$ for at least one i, $1 \leq i \leq k$, *infinite multisets*. The set of all such multisets M over V with $M : V \longrightarrow \mathbb{N}_\infty$ is denoted by $\langle V, \mathbb{N}_\infty\rangle$. For $W \subseteq V$, W^∞ denotes the infinite multiset with $W(a) = \infty$ for all $a \in W$.

Let x and y be two multisets over V, i.e., from $\langle V, \mathbb{N}\rangle$ or $\langle V, \mathbb{N}_\infty\rangle$. Then x is called a submultiset of y, written $x \leq y$ or $x \subseteq y$, if and only if $x(a) \leq y(a)$ for all $a \in V$; if, moreover, $x(a) < y(a)$ for some $a \in V$, then x is called a strict submultiset of y. Observe that for all $n \in \mathbb{N}$, $n + \infty = \infty$, and $\infty - n = \infty$. The sum of x and y, denoted by $x + y$ or $x \cup y$, is a multiset z such that $z(a) = x(a)+y(a)$ for all $a \in V$. The difference of two multisets x and y, denoted by $x - y$, provided that $y \subseteq x$, is the multiset z with $z(a) = x(a) - y(a)$ for all $a \in V$. Observe that in the following, when taking the sum or the difference of two multisets x and y from $\langle V, \mathbb{N}_\infty\rangle$, we shall always assume $\{x(a), y(a)\}\cap\mathbb{N} \neq \emptyset$.

If $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_m)$ are vectors of multisets over V, then $X \leq Y$ if and only if $x_j \subseteq y_j$ for all j, $1 \leq j \leq m$; in the same way, sum and difference of vectors of multisets are defined by taking the sum and the difference, respectively, in each component.

Throughout the rest of the paper, we will not distinguish between a multiset from $\langle V, \mathbb{N}\rangle$ and its representation by a string over V containing the corresponding number of each symbol. Moreover, when we speak of a partitioning of a set R into a set $\{R_i \mid 1 \leq i \leq h\}$ of subsets of R, i.e., $R_i \subseteq R$, $1 \leq i \leq h$, the R_i are not necessarily disjoint.

3 Networks of Cells

In this section we consider membrane systems as a collection of interacting cells containing multisets of objects like in [2] and [8]. For an introduction to the area of membrane computing we refer the interested reader to the monograph [11], the actual state of the art can be seen in the web [15].

Definition 3.1. A *network of cells with checking sets of degree* $n \geq 1$ is a construct

$$\Pi = (n, V, w, R)$$

where

1. n is the number of cells;

2. V a *finite alphabet*;

3. $w = (w_1, \dots, w_n)$ where $w_i \in \langle V, \mathbb{N}_\infty \rangle$, for all $1 \leq i \leq n$, is the *multiset initially associated to cell* (in most of the cases, at most one cell, then being called *the environment*, will contain symbols occurring with infinite multiplicity);

4. R is a finite set of *interaction rules* of the form

   $$(E : X \to Y)$$

   where E is a recursive condition for configurations of Π (see definition below) as well as $X = (x_1, \dots, x_n)$, $Y = (y_1, \dots, y_n)$, with $x_i, y_i \in \langle V, \mathbb{N} \rangle$, $1 \leq i \leq n$, are vectors of multisets over V. We will also use the notation

   $$(E : (x_1, 1) \dots (x_n, n) \to (y_1, 1) \dots (y_n, n))$$

   for a rule $(E : X \to Y)$.

A network of cells consists of n cells, numbered from 1 to n, that contain (possibly infinite) multisets of objects over V; initially cell i contains w_i. A *configuration* C of Π is an n-tuple of multisets over V $(u_1, \dots, u_n)$; the *initial configuration* of Π, C_0, is described by w, i.e., $C_0 = w = (w_1, \dots, w_n)$. Cells can interact with each other by means of the rules in R. An interaction rule

$$(E : (x_1, 1) \dots (x_n, n) \to (y_1, 1) \dots (y_n, n))$$

is applicable to a configuration C if and only if C fulfills condition E; its application means rewriting objects x_i from cells i into objects y_j in cells j, $1 \leq i, j \leq n$.

The set of all multisets of rules *applicable* to C is denoted by $Appl\,(\Pi, C)$ (a procedural algorithm how to obtain $Appl\,(\Pi, C)$ is described in [8]).

For the specific *derivation modes* to be defined in the following, the selection of multisets of rules applicable to a configuration C has to be a specific subset of $Appl\,(\Pi, C)$; for the derivation mode ϑ, the selection of multisets of rules applicable to a configuration C is denoted by $Appl\,(\Pi, C, \vartheta)$.

Definition 3.2. For the *asynchronous* derivation mode (*asyn*),

$$Appl\,(\Pi, C, asyn) = Appl\,(\Pi, C)\,,$$

i.e., there are no particular restrictions on the multisets of rules applicable to C.

Definition 3.3. For the *sequential* derivation mode (*sequ*),

$$Appl\,(\Pi, C, sequ) = \{R' \mid R' \in Appl\,(\Pi, C) \text{ and } |R'| = 1\}\,,$$

i.e., any multiset of rules $R' \in Appl\,(\Pi, C, sequ)$ has size 1.

The most important derivation mode considered in the area of P systems from the beginning is the *maximally parallel* derivation mode where we only select multisets of rules R' that are not extensible, i.e., there is no other multiset of rules $R'' \supsetneqq R'$ applicable to C.

Definition 3.4. For the *maximally parallel* derivation mode (max),

$$\begin{aligned} Appl\,(\Pi, C, max) \quad = \quad & \{R' \mid R' \in Appl\,(\Pi, C) \text{ and there is} \\ & \text{no } R'' \in Appl\,(\Pi, C) \text{ with } R'' \supsetneqq R'\}\,. \end{aligned}$$

For the *minimally parallel* derivation mode, we need an additional feature for the set of rules R, i.e., we consider a partition of R into disjoint subsets R_1 to R_h. Usually, this partition of R may coincide with a specific assignment of the rules to the cells. For any set of rules $R' \subseteq R$, let $\|R'\|$ denote the number of sets of rules R_j, $1 \leq j \leq h$, with $R_j \cap R' \neq \emptyset$.

There are several possible interpretations of this minimally parallel derivation mode which in an informal way can be described as applying multisets such that from every set R_j, $1 \leq j \leq h$, at least one rule – if possible – has to be used (e.g., see [3]). For the basic variant as defined in the following, in each derivation step we choose a multiset of rules R' from $Appl\,(\Pi, C, asyn)$ that cannot be extended to $R'' \in Appl\,(\Pi, C, asyn)$ with $R'' \supsetneqq R'$ as well as $(R'' - R') \cap R_j \neq \emptyset$ and $R' \cap R_j = \emptyset$ for some j, $1 \leq j \leq h$, i.e., extended by a rule from a set of rules R_j from which no rule has been taken into R'.

Definition 3.5. For the *minimally parallel* derivation mode (min),

$$\begin{aligned} Appl\,(\Pi, C, min) \quad = \quad & \{R' \mid R' \in Appl\,(\Pi, C, asyn) \text{ and} \\ & \text{there is no } R'' \in Appl\,(\Pi, C, asyn) \\ & \text{with } R'' \supsetneqq R',\ (R'' - R') \cap R_j \neq \emptyset \\ & \text{and } R' \cap R_j = \emptyset \text{ for some } j,\ 1 \leq j \leq h\}\,. \end{aligned}$$

In [8], further restricting conditions on the four basic modes defined above, especially interesting for the minimally parallel derivation mode, were considered. We now consider a restricted variant of the minimally parallel derivation mode allowing only a bounded number of at most k rules to be taken from each set R_j, $1 \leq j \leq h$, of the partitioning into a multiset of rules applicable in the minimally parallel derivation mode.

Definition 3.6. For the *k-restricted minimally parallel* derivation mode (min_k),

$$\begin{aligned} Appl\,(\Pi, C, min_k) \quad = \quad & \{R' \mid R' \in Appl\,(\Pi, C, min) \text{ and} \\ & |R' \cap R_j| \leq k \text{ for all } j,\ 1 \leq j \leq h\}\,. \end{aligned}$$

For all the derivation modes defined above, we now can define how to obtain a next configuration from a given one by applying an applicable multiset of rules according to the constraints of the underlying derivation mode:

Definition 3.7. Given a configuration C of Π and a derivation mode ϑ, we may choose a multiset of rules $R' \in \mathit{Appl}\,(\Pi, C, \vartheta)$ in a non-deterministic way and apply it to C. The result of this *transition step* from the configuration C with applying R' is the configuration $\mathit{Apply}\,(\Pi, C, R')$, and we also write $C \Longrightarrow_{(\Pi,\vartheta)} C'$. The reflexive and transitive closure of the transition relation $\Longrightarrow_{(\Pi,\vartheta)}$ is denoted by $\Longrightarrow^*_{(\Pi,\vartheta)}$.

Definition 3.8. A *computation* in a network of cells Π, $\Pi = (n, V, w, R)$, starts with the initial configuration $C_0 = w$ and continues with transition steps according to the chosen derivation mode ϑ; it is called *successful* if we reach a configuration C to which no multiset of rules can be applied with respect to the derivation mode ϑ anymore, i.e., $\mathit{Appl}\,(\Pi, C, \vartheta) = \emptyset$ (we also say that the computation *halts*).

As the results of a halting computation we take the number of objects in a specified output cell. We shall use the notation

$$O_m C_n\,(\vartheta)\,[\text{parameters for rules}]$$

to denote the family of sets of natural numbers generated by networks of cells $\Pi = (n, V, w, R)$ with $m = |V|$ as well as ϑ indicating the derivation mode; the *parameters for rules* describe the specific features of the rules in R. If any of the parameters m and n is unbounded, we replace it by $*$.

4 Specific Examples for the 1-Restricted Minimally Parallel Derivation Mode

In this section, we show how the 1-restricted minimally parallel derivation mode may capture characteristic features of well-known models of P systems.

We first consider extended spiking neural P systems (without delays, see [1]), where the rules are applied in a sequential way in each neuron, but on the level of the whole system, the maximally parallel derivation mode is applied (every neuron which may use a spiking rule has to spike, i.e., to apply a rule, see the original paper [9]). When partitioning the rule set according to the set of neurons, the application of the 1-restricted minimally parallel derivation mode exactly models the original derivation mode defined for spiking neural P systems.

Already in the original paper of Gh. Păun (see [10]), membrane systems with catalytic rules were defined, but used together with other noncooperative rules. In [6] it was shown that only three catalysts are sufficient in one membrane, using only catalytic rules with the maximally parallel derivation mode, to generate any recursively enumerable set of natural numbers. Hence, by showing that P systems with purely catalytic rules working in the maximally parallel derivation mode can be considered as P systems working with the corresponding noncooperative rules in the 1-restricted minimally parallel derivation mode when partitioning the rule sets for each membrane with respect to the catalysts, we obtain the astonishing result that in this case we get a characterization of the recursively enumerable sets of natural numbers by using only noncooperative rules, i.e., $NRE = O_*C_1\,(min_1)$ [noncoop].

4.1 Extended Spiking Neural P Systems

An *extended spiking neural P system* (of degree $m \geq 1$) (in the following we shall simply speak of an *ESNP system*) is a construct

$$\Pi = (m, S, R)$$

where

- m is the number of *neurons*; the neurons are uniquely identified by a number between 1 and m;
- S describes the *initial configuration* by assigning an initial value (of spikes) to each neuron;
- R is a finite set of *rules* of the form $\left(i, E/a^k \to P\right)$ such that $i \in [1..m]$ (specifying that this rule is assigned to neuron i), $E \subseteq REG\,(\{a\})$ is the *checking set* (the current number of spikes in the neuron has to be from E if this rule shall be executed), $k \in \mathbb{N}$ is the "number of spikes" (the energy) consumed by this rule, and P is a (possibly empty) set of *productions* of the form (l, a^w) where $l \in [1..m]$ (thus specifying the target neuron), $w \in \mathbb{N}$ is the *weight* of the energy sent along the axon from neuron i to neuron l.

A *configuration* of the ESNP system is described by specifying the actual number of spikes in every neuron. A *transition* from one configuration to another one is executed as follows: for each neuron i, we non-deterministically choose a rule $\left(i, E/a^k \to P\right)$ that can be applied, i.e., if the current value of spikes in neuron i is in E, neuron i "spikes", i.e., for every production (l, w) occurring in the set P we send w spikes along the axon from neuron i to neuron l. A *computation* is a sequence of configurations starting with the initial configuration given by S. An ESNP system can be used to generate sets from NRE (we do not distinguish between NRE and $RE\,(\{a\})$) as follows: a computation is called *successful* if it halts, i.e., if for no neuron, a rule can be activated; we then consider the contents, i.e., the number of spikes, of a specific neuron called *output neuron* in halting computations.

We now consider the ESNP system $\Pi = (m, S, R)$ as a network of cells $\Pi' = (m, \{a\}, S, R')$ working in the 1-restricted minimally parallel derivation mode, with

$$\begin{aligned} R' \quad = \quad & \left\{\left(E : \left(a^k, i\right) \to (a^{w_1}, l_1) \ldots (a^{w_n}, l_n)\right) \mid \right. \\ & \left. \left(i, E/a^k \to (l_1, a^{w_1}) \ldots (l_n, a^{w_n})\right) \in R\right\} \end{aligned}$$

and the partitioning R'_i, $1 \leq i \leq m$, of the rule set R' according to the set of neurons, i.e.,

$$\begin{array}{rcl} R'_i & = & \left\{ \left(E : \left(a^k, i\right) \rightarrow (a^{w_1}, l_1) \ldots (a^{w_n}, l_n) \right) \mid \right. \\ & & \left. \left(E : \left(a^k, i\right) \rightarrow (a^{w_1}, l_1) \ldots (a^{w_n}, l_n) \right) \in R' \right\} . \end{array}$$

The 1-restricted minimally parallel derivation mode chooses one rule – if possible – from every set R_i and then applies such a multiset of rules in parallel, which directly corresponds to applying one spiking rule in every neuron where a rule can be applied. Hence, it is easy to see that Π' and Π generate the same set from $RE\,\{a\}$ if in both systems we take the same cell/neuron for extracting the output. Due to the results valid for ESNP systems, see [1], we obtain

$$NRE = O_1 C_3\,(min_1)\,[\mathrm{ESNP}] .$$

4.2 Purely Catalytic P Systems

A noncooperative evolution rule is of the form $(I : (a, i) \rightarrow (y_1, 1) \ldots (y_n, n))$ where a is a single symbol and I denotes the condition that is always fulfilled. A catalytic rule is of the form $(I : (c, i)\,(a, i) \rightarrow (c, i)\,(y_1, 1) \ldots (y_n, n))$ where c is from a distinguished subset $V_C \subset V$ such that in all rules (noncooperative evolution rules, catalytic rules) of the whole system the y_i are from $(V - V_C)^*$ and the symbols a are from $(V - V_C)$. Imposing the restriction that the noncooperative evolution rules and the catalytic rules in a network of cells allow for finding a hierarchical tree structure of membranes such that symbols either stay in their membrane region or are sent out to the surrounding membrane region or sent into an inner membrane, then we get the classical catalytic P systems without priorities. Allowing regular sets checking for the non-appearance of specific symbols instead of I, we even get the original P systems with priorities. Catalytic P systems using only catalytic rules are called purely catalytic P systems. As we know from [6], only two (three) catalysts in one membrane are needed to obtain NRE with (purely) catalytic P systems without priorities working in the maximally parallel derivation mode, i.e., we can write these results as follows:

$$NRE = O_* C_1\,(max)\,[\mathrm{cat}_2] = O_* C_1\,(max)\,[\mathrm{pcat}_3] .$$

If we now partition the rule set in a purely catalytic P system according to the catalysts present in each membrane, this partitioning replaces the use of the catalysts when working in the 1-restricted minimally parallel derivation mode, because by definition from each of these sets then – if possible – exactly one rule (as with the use of the corresponding catalyst) is chosen: from the set of purely catalytic rules R we obtain the corresponding set of noncooperative rules R' as

$$\begin{array}{rcl} R' & = & \{ (I : (a, i) \rightarrow (y_1, 1) \ldots (y_n, n)) \mid \\ & & (I : (c, i)\,(a, i) \rightarrow (c, i)\,(y_1, 1) \ldots (y_n, n)) \in R \} \end{array}$$

as well as the corresponding partitioning of R' as

$$\begin{array}{rcl} R'_{i,c} & = & \{ (I : (a, i) \rightarrow (y_1, 1) \ldots (y_n, n)) \mid \\ & & (I : (c, i)\,(a, i) \rightarrow (c, i)\,(y_1, 1) \ldots (y_n, n)) \in R \} . \end{array}$$

Considering purely catalytic P systems in one membrane, we therefore infer the following quite astonishing result that when using the 1-restricted minimally parallel derivation mode for a suitable partitioning of rules we only need noncooperative rules:

$$NRE = O_*C_1\,(min_1)\,[\text{noncoop}]\,.$$

5 Conclusions

The main purpose of this paper was to introduce the k-restricted minimally parallel derivation mode and to elaborate how this new derivation mode allows for capturing the main characteristics of well-known variants of membrane systems. The 1-restricted minimally parallel derivation mode, for example, allows us to interpret the way of how spiking neural P systems (without delays) work in a sequential way on the level of cells, but in the maximally parallel way on the level of the whole system by using this 1-restricted minimally parallel derivation mode for the whole system with a partitioning of the rules given by the individual neurons. P systems with purely catalytic rules working in the maximally parallel derivation mode can be considered as P systems working with the corresponding noncooperative rules in the 1-restricted minimally parallel derivation mode when partitioning the rule sets for each membrane with respect to the catalysts.

In the general framework considered in this paper, many other variants of static P systems and tissue P systems can be considered, hence, a great variety of such systems working in the k-restricted minimally parallel derivation mode, especially for $k = 1$, remains to be investigated in the future. Moreover, the basic k-restricted minimally parallel derivation mode may be restricted as exhibited for the other derivation modes as shown in [8], eventually with other variants of halting. As a specific example of new results of that kind we should like to mention that, with the variant of the k-restricted minimally parallel derivation mode having to include at most k rules from each set of the partitioning from which a rule is applicable to the current configuration into a multiset of rules to be applied, together with partial halting we can only obtain regular sets of natural numbers.

Acknowledgements

The authors gratefully acknowledge the useful suggestions and remarks from Artiom Alhazov and Markus Beyreder.

References

[1] A. Alhazov, R. Freund, M. Oswald, M. Slavkovik: Extended spiking neural P systems generating strings and vectors of non-negative integers. In:

H.J. Hoogeboom, Gh. Paun, and G. Rozenberg, (Eds.), *Pre-proceedings of Membrane Computing, International Workshop, WMC7*, Leiden, The Netherlands, 2006, 88–101.

[2] F. Bernardini, M. Gheorghe, M. Margenstern, S. Verlan: Networks of cells and Petri nets. In: M. A. Gutiérrez-Naranjo, Gh. Păun, A. Romero-Jiménez, A. Riscos-Núñez: *Proc. Fifth Brainstorming Week on Membrane Computing*, Sevilla, 2007, 33–62.

[3] G. Ciobanu, L. Pan, Gh. Păun, M.J. Pérez-Jiménez: P systems with minimal parallelism, *Theoretical Computer Science* **378** (1) (2007), 117–130.

[4] E. Csuhaj-Varjú: Networks of language processors. *Current Trends in Theoretical Computer Science* (2001), 771–790.

[5] J. Dassow, Gh. Păun: On the power of membrane computing, *Journal of Universal Computer Science* **5** (2) (1999), 33–49.

[6] R. Freund, L. Kari, M. Oswald, P. Sosík: Computationally universal P systems without priorities: two catalysts are sufficient. *Theoretical Computer Science* **330** (2005), 251–266.

[7] R. Freund, Gh. Păun, M.J. Pérez-Jiménez: Tissue-like P systems with channel states. *Theoretical Computer Science* **330** (2005), 101–116.

[8] R. Freund, S. Verlan: A formal framework for P systems. In: G. Eleftherakis, P. Kefalas, Gh. Paun (Eds.), Pre-proceedings of Membrane Computing, International Workshop – WMC8, Thessaloniki, Greece, 2007, 317–330.

[9] M. Ionescu, Gh. Păun, T. Yokomori: Spiking neural P systems. *Fundamenta Informaticae* **71**, 2–3(2006): 279–308.

[10] Gh. Păun: Computing with membranes, *J. of Computer and System Sciences* **61**, 1 (2000), 108–143, and TUCS Research Report 208 (1998) (`http://www.tucs.fi`).

[11] Gh. Păun: *Membrane Computing. An Introduction.* Springer-Verlag, Berlin, 2002.

[12] G. Rozenberg, A. Salomaa (Eds.): *Handbook of Formal Languages* (3 volumes), Springer-Verlag, Berlin, 1997.

[13] Gh. Păun, Y. Sakakibara, and T. Yokomori: P systems on graphs of restricted forms. *Publicationes Matimaticae* **60**, 2002.

[14] Gh. Păun and T. Yokomori: Membrane computing based on splicing. In: E. Winfree and D. K. Gifford (Eds.), *DNA Based Computers V*, volume 54 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 217–232. American Mathematical Society, 1999.

[15] The P Systems Web Page: `http://ppage.psystems.eu.`