

**P automata:
Models, Results, and Research Topics
(Extended abstract)***

Erzsébet CSUHAJ-VARJÚ

Computer and Automation Research Institute
Hungarian Academy of Sciences
Kende utca 13-17, 1111 Budapest, Hungary
E-mail: csuhaj@sztaki.hu

1 Introduction

”Membrane systems are computing devices abstracted from the functioning of the living cells.” These and similar sentences can be found in hundreds of publications about models, results, and ideas in a recent area of molecular computing, in the theory of membrane systems or P systems. The research field has been launched by Gheorghe Păun in 1998 [22], with the inspiration to construct a framework which provides us with effective and powerful computational tools and, at the same time, it gives the possibility to study and simulate natural processes. Since 1998, the fruitful idea has been extensively and intensively explored, the theory of membrane systems or P systems has proved to be a successful and promising area of unconventional models of computation. Several variants of the basic notion have been introduced and investigated proving the power of the framework; the interested reader is referred to [23, 24, 27] for basic information, and to the book [26] for a summary of the achievements and open problems in the area. The reader also can consult the P systems web page with a lot of downloadable papers and information [28].

The main components of P systems are membrane structures consisting of membranes hierarchically embedded in the outermost skin membrane. Each membrane encloses a region containing a multiset of objects and possibly other membranes. Each region has an associated set of operators operating on the objects contained by the region. These operators can be of different types, they can modify the multisets of objects in the regions but also can provide the possibility of transferring the objects from one region to another one. The first type of these operations are the so-called evolution rules, which can be applied in parallel across all membranes or in a sequential manner. The rules to be applied are chosen nondeterministically, that is, if an object can evolve according to more than one evolution rule at the same time, than any one can be chosen. The second types of rules, providing the possibility of transportation of objects, are called communication rules.

*Research supported in part by project “MolCoNet” IST-2001-32008.

At any moment of time, the membrane system can be described by its configuration which consists of the actual membrane structure and the contents of the regions. (We note that some of the variants of P systems allow to dynamically change the membrane structure.) In this sense, membrane systems can be considered as computing devices: starting from an initial configuration, the system evolves by passing one configuration to another one, thus realizing a computation. If the system halts, that is, no rule can be applied anymore, the computation is successful. For more details and the different variants the reader is referred to [26, 28]. However, we also can see that the sequence of configurations describes the behaviour of the P system, thus they can be investigated from system theoretic points of view as well.

Considering the P systems briefly described above, the reader can observe that these constructs restrict their functioning to the membranes and to the contents of the regions, the system is not in interaction or communication with its surrounding environment, with the outside world. However, since P systems attempt to model living cells and the cell communicates with its biological environment, it is reasonable to take this aspect also into account. Thus, studying variants of P systems, where the system is in communication (in interaction) with its environment is a well-motivated area of research in membrane systems theory. This communication can be interpreted in several manner: One possibility is, for example, that the environment represents an infinite (finite) supply of objects, from which individual objects or multisets of objects can be or must be imported in the system by the skin membrane under the functioning of the system. It is easy to see, that these P systems can be considered as automata or accepting systems (P automata or variants of accepting P systems), since the configuration of the membrane system changes due to both the accepted (imported) objects and its actual state.

Although the idea of defining accepting variants of P systems is a very reasonable one, the theory of P automata or accepting P systems explicitly was inspired by two problems raised by Gheorghe Păun. The first, from [25] is the following. “What about the possibility of considering a class of P systems, meant to compute, where no rule for objects evolution appears, but only rules governing object communication from a region to another one”. The second, from [26], Problem Q32: “What about using P systems as accepting devices?”

Motivated by these questions, the first variant of P automata was introduced in [6], realizing a purely communicating, accepting P system (see also [7]). Almost at the same time, a closely related notion, the analysing P system was introduced in [10], formulating another concept for defining of an accepting P system. Both types of membrane systems proved that not only generating but accepting P systems are sufficiently powerful tools for computing. Since that time, several variants of P automata have been introduced and studied, for more information the reader is referred to the excellent summary [20] and for further details to the articles referred in the on-line bibliography [28]. Although all of these models are accepting P systems, they differ from each other in several properties: in the way of communication with the environment, in the types of communication rules used by the regions, in the way of the functioning of the membrane system (whether or not it has evolution rules), and in the way of defining the acceptance.

In the following, without the aim of completeness and without the formal details

- that can be found in the corresponding articles - we briefly recall some models and discuss their computational power. Finally, we propose some new topics and problems for future research.

2 Variants of P automata: power and size

As we mentioned above, the first variant of P automata was introduced in [6], as a purely communicating accepting P system with one-way (top-down) communication. According to this model, under some given conditions, a multiset of objects can be imported into a membrane i from its parent membrane j ; the skin membrane can import a multiset of objects from the environment (the outer region). The environment is supposed to contain infinitely many copies of any object. The conditional communication (transportation) rules are symport rules with promoters of the form $(y, in)|_x$, where x and y are multisets of objects. The rule $(y, in)|_x$ means the following: For $x, y \neq \varepsilon$, if x is contained in region i and y is contained in its parent region, then the objects of y must leave the parent region and enter region i . If $x = \varepsilon$, then the region i must be empty, if $y = \varepsilon$, then no object is requested from the parent region. Starting from an initial configuration, which is given as an n -tuple of multisets of objects - supposing that the P system consist of n membranes -, a computation is performed by applying the rules sequentially. That is, any step of the computation only one, non-deterministically chosen rule is applied in the region. The computation stops if the functioning of the system aborts, that is, no rule can be applied in some of the regions. The configuration of the system enters a so-called final state, which is defined by non-empty sets of multisets F_i assigned to any region i , if the contents of the region coincides with an element of F_i . If for some region j , $F_j = \emptyset$, then a configuration can be final regardless of the contents of region j . The sequence of multisets of objects requested by and entered the skin membrane during the computation forms the so-called input sequence of the P automaton; an input sequence is accepted by the membrane system if it is consumed by the skin membrane under the functioning of the system starting from the initial state and entering a final state. The language of the accepted input sequences is called the language of the P automaton.

The reader can observe that the sequence of multisets of objects requested by the skin membrane of the P automaton from the outer region can be considered as an input sequence, and the regions are related to tapes which change their contents in parallel depending on the input. Thus, we define a sequence of multisets of objects accepted by the P automaton as an input sequence which, after being consumed by the skin membrane, causes the system to enter a final state.

These variants of P systems proved to be very powerful computational tools: in [6] it was shown that any recursively enumerable language can be obtained as a mapping of the language of a one-way P automaton, even with systems with only seven membranes. To prove the result, a widely used technique from membrane computing was applied, the simulation of a two-counter machine by a P system. While the result proved to be useful and interesting, the question about the exact characterization of the language classes of P automata remained open.

Although the original model allowed only symport rules with promoters to use, and even in a sequential manner, it appeared to be reasonable to extend the formalism to systems with both symport and antiport rules and with or without promoters and inhibitors. Moreover, it also appeared to be natural to consider not only the sequential, but the maximally parallel manner use of communication rules as well.

The answer to the previously mentioned question was given in [5], where the classes of languages of P automata, with the above extensions of the original notion, were characterized. It was shown that if the rules of the P system are applied sequentially, then the accepted language class is strictly included in the class of languages accepted by one-way Turing machines with a logarithmically bounded workspace, while if the rules are applied in the maximal parallel manner, then the class of context-sensitive languages can be determined.

In the sequential case, the number of different multisets that may ever enter the system is finite which means that there is a natural one-to-one correspondence between these multisets and the symbols of a finite alphabet. This is not necessarily so when the rules are applied in the maximal parallel way, in this case P automata can be considered as devices accepting finite strings over an infinite alphabet. However, the case of infinite alphabets was not studied in the above paper, instead a mapping that maps the infinite set of different multisets to a finite alphabet was used, thus it was possible to speak of languages accepted by P automata using the rules in the sequential or in the maximal parallel manner, the languages being in both cases over a finite alphabet. We note that the case of infinite alphabets would be of particular interest for future research.

A concept developed from the original variant of P automata [6] was introduced in [19], we can call it P automata with states, where each membrane has a state from a given finite set, and the communication rules are of the form $(py, in)|_{qx}$, where p, q are states and x, y are multisets of symbols. The rule means the following: for $x, y \neq \varepsilon$, if x is contained in region i which is in state q and y is contained in its parent region, then the objects of y must leave the parent region and enter region i and then the state of the region i is changed for p . If $x = \varepsilon$, then the region i must be empty, if $y = \varepsilon$, then no object is requested from the parent region. The notion was also motivated by tissue like P systems, and unlike the sequential application of the rules in the original variant of P automata from [6], the authors consider maximal mode application of rules (given for tissue P systems). It was shown that these constructs describe the recursively enumerable sets of vectors of natural numbers.

Improvements of results from [6] and [19] were presented in [9]. It was shown that one-way P automata are able to compute any recursively enumerable language, even in the case of having only two membranes, and with promoters and moved multisets of size (elements) of two. The authors also proved that the result from [19] can be extended to languages, moreover, to obtain computational completeness it is sufficient to consider only rules of restricted forms.

Some further models which are closely related or related to the original one are the P automata with priorities (among the rules) [4], the evolution-communication P automata, where both communication and evolution rules are allowed to use [1], and the P automata with tapes [18], where the input can be found on a tape which is transferred across the membranes during the computation. We note that

although the concept of P automata was formulated to develop a concept for purely communicating, accepting P systems, if we consider P systems as models of evolving (dynamically changing) systems interacting with their environments, P automata with both communication and evolution rules are of particular interest.

Similarly, an interesting and well-motivated concept is introduced in [2, 3], where so-called active P automata is defined. In this case, unlike the previously defined variants, the construct computes with the structure of the membrane system, using operations like membrane creation, division, and dissolution. Briefly, in an active P automata the accepting computation starts with one membrane which contains the string to be accepted and some other information. The computation follows according to the input string and during the evolution membranes can be created or dissolved. The computation ends, when all input symbols are consumed and also some termination condition is fulfilled. The authors demonstrate how to apply these constructs in natural language processing, in parsing. As in the previous case, we would like to mention that these and similar variants of P automata, where the membrane structure can dynamically change under functioning is of particular interest, since it makes closer the concept to natural systems.

Another model, strongly motivated by natural processes taken place in cells is the so-called P automaton with membrane channels, see [13, 20, 21] for the details. In this case, the sets of communication rules of the regions are of the following forms: $\langle P; x, out; y, in \rangle$, the so-called activating rules, and $\langle b, out; Q \rangle$ or $\langle b, in; Q \rangle$, the so-called prohibiting rules. In this model, starting from the initial configuration, the system passes configurations by using its rules in a nondeterministic, maximally parallel manner, where the activating rules and the prohibiting rules mean the following: Let $x = x_1 \dots x_m$ and $y = y_1 \dots y_k$. Then, an activating rule $\langle P; x, out; y, in \rangle$ means that by the activator multiset P an output channel for each symbol x_i is activated, and for each y_j an in input channel is activated. Then, each activated channel allows the transport of one object x_i and y_j , provided that no prohibitor multiset Q is active by a prohibiting rule, respectively. This model proved to be computational complete as well, moreover, it was shown that one membrane with singleton activators and prohibitors is sufficient to obtain this computational power. Analogous results were obtained about so-called initial P automata with membrane channels. (Roughly speaking, in the case of an initial automaton, the multiset to be analysed is initially put into a specified membrane together with possibly some other objects (symbols).)

Continuing the investigations of the role of conditions associated with the constituents of the membrane systems, the notion of (initial) P automata with conditional communication rules associated with the membranes was introduced and studied (see [20], and [11] for background information). The rules of these constructs are of the form $(P_{in}, Q_{in}; P_{out}, Q_{out}; y, in; x, out)$, which means that the transporting of the multiset x outside the membrane and the transporting of multiset y inside the membrane is possible if and only if the promoting multisets P_{in} and P_{out} are present in the respective regions, while the inhibiting multisets Q_{in} and Q_{out} , respectively, cannot be found in them. As it was expected, these variants of P automata proved to be computationally complete, even in the case with one membrane and singleton promoters, inhibitors, and transported objects.

The original variant of P automata and the above variants used final configurations for determining acceptance (in some sense, we can consider active P automata to be with this feature as well). However, accepting computation can also be defined by halting, thus being closer to the original concept of defining computation in membrane systems.

In analysing P systems (with antiport rules), introduced in [10], this way of accepting is defined. The concept was published just after, almost at the same time as the one-way P automaton. According to this idea, the membranes of the P system have only communication rules which are antiport rules of the form $(x, out; y, in)$, which means that a multiset x is sent out from the membrane and a multiset y is taken into the region of the membrane from the surrounding region. (Notice that these rules are not with promoters and/or inhibitors). Starting from an initial configuration, which is given by the membrane structure and the initial multisets in the regions, the P system performs computation steps, by applying its rules in a maximal parallel way. The skin membrane communicates with the outside world, with the environment, where only terminal symbols as objects can be found. A sequence of computation steps is successful, if and only if the system halts after a while. Then, the analysed string is the sequence of terminal symbols that were taken from the environment. If more than one terminal symbol is taken from the environment in one step, then any permutation of these symbols is considered as a valid subword of the input string. It was shown that these systems compute any recursively enumerable language, moreover, only one membrane using antiport rules with radius $(1, 2)$ or $(2, 1)$ is sufficient to obtain this computational power. (The radius of x , respectively y , is the number of its elements.) Moreover, the computational completeness of these constructs was demonstrated also in the case, where the multiset over the set of terminal symbols is initially put into a specified membrane together with possibly some other, non-terminal symbols (for initial analysing P systems).

An important variant of the concept of accepting P systems are, the so-called catalytic P automata or P systems with catalysts [8, 12]. Catalytic P systems were introduced already in [23], these are membrane systems with special objects called catalysts which are used in the evolution rules. In these systems, the evolution rules are of the forms $a \rightarrow v$ or $ca \rightarrow cv$, where c is a catalyst, a is an object which is not catalyst, and v is a string from $((V \setminus C) \times \{here, out, in\})^*$. (V is the set of objects and C is a proper subset of V , the set of catalysts.) In these systems the transition between two configurations is governed by the evolution rules, done in parallel, that is, any object which can be a subject of evolution, must evolve according to a local rule. The transportation of the objects is realized through the targets added to the evolution rules, namely, *in, out, here*. Catalytic systems have been in the focus of interest since the beginning of P systems theory. For the case of P automata with catalysts particularly important and interesting results were obtained, in [8] it has been shown that for P systems with external output only one membrane and two catalysts are sufficient to obtain computational completeness. The proof is based on an inventive proof technique simulating the register machine.

Another interesting model was introduced in [15, 16, 17], as a restricted variant of communicating P systems, called the restricted communicating P system. In

this case, the objects taken from the environment can only be those one which have already been present in the system at the initialization, and were sent out from the system. The system has special evolution rules, which are of the form $a \rightarrow a_\tau$, or $ab \rightarrow a_{\tau_1}b_{\tau_2}$, or $ab \rightarrow a_{\tau_1}b_{\tau_2}c_{come}$. Letters a, b , and c denote objects, while $\tau, \tau_1, \tau_2 \in \{here, out\} \cup \{in_j \mid 1 \leq j \leq n\}$ (n is the number of the membranes in the system). For an object, *out* means that it is transported out from the region, in_j means that it is transported from membrane j in, while *here* refers to that it must stay in the same region. Evolution rules of the third type can be used only in the skin membrane, with the meaning that an object c is transported in from the environment. (Remember, that this is only possible if a c had already been sent out from the membrane system.) Thus, the number of objects in the P system and its environment together remains the same under the computation. It was shown that these constructs are equivalent to two-way multihead finite automata over bounded languages. Moreover, it also has been proved that the number of membranes in these P systems induces an infinite hierarchy according to their computational power. We should note that this type of P automata is particularly interesting, since in this case the system has a bounded environment, more precisely, its actual environment consists of those objects which are temporarily or for ever non-activated (they are sent out, are not in the membrane system).

Another variant of P automata, with well-founded motivation is the so-called ω - P automata [13, 14]. In this work, a special type of P automaton is introduced (with membrane channels and antiport rules), to simulate the functioning of ω -Turing machines, that is, actions of Turing machines on infinite words. To formulate the proper notion, special efforts and considerations had to be made, since in P systems successful computations are usually defined by halting and the failing computations are defined by non-halting (with failure symbols with rules for allowing infinite computations). In the case of infinite words to be analysed, failing computations must stop. The authors were able to show that for any well-known variant of acceptance mode of ω -Turing machines one can construct an ω - P automaton which simulates the computations of the corresponding ω -Turing machine. These types of P automata are of special interest, since assuming a P automaton as a system being with interaction in its environment, we also should consider communication processes not limited in time.

The previous models demonstrated that the different variants of P automata are - almost in any case - computationally complete computational devices, and this power can be obtained even with systems with bounded size. Although it is an important issue to determine the computational power of accepting P systems, we should also pay attention to the characteristics of the way of their functioning as well. Among these properties, determinism plays an outstanding role. However, while it is easy to formulate a concept describing determinism, for example, for context-free grammars, this is not the case for sophisticated constructs with sophisticated behaviour as P automata. A successful attempt has been made in [20], where so-called k -determinism (a "weak" type of determinism) was introduced and interpreted, based on the computation tree in P automata. This means, by [20], that for "every run starting from an initial configuration, if at any moment going at most k steps further for an arbitrary choice of productions to be applied, it can

be decided (i.e. syntactically checked) which might be a reasonable continuation that possibly may lead to successful acceptance.” An important and typical result is, that for every recursively enumerable set of vectors of natural numbers there exists a 2-deterministic initial analysing P system with antiport rules with radius $(2, 1)$ or $(1, 2)$. The notion of k -determinism, and the notion of determinism at all, raises several further interesting research topics for the future. Investigations in the determinism of P automata is certainly of among the most interesting topics, and a lot of new results in the area are expected in the future.

3 Topics for future research

Investigations in the theory of P automata expected to be continued in several directions. Firstly, P automata can be considered as constructs attempting to build a bridge between automata theory and membrane systems theory, thus similarities and differences between the two fields are certainly of interest. But, as we mentioned in the Introduction, P automata or accepting P systems are models of dynamically changing systems which are in communication (interaction) with their environments as well. According to this approach, the behaviour of the P automaton is of special interest, which can be interpreted as the set of accepted (consumed) multiset sequences, but also can be defined as the sequence of its configurations following each other. The next question immediately arises: Is there any difference between the so-called input-driven behaviour of the system and the so-called state-driven behaviour of the system. (This question, that is, the difference between the two interpretations of the notion of P automata was asked by György Vaszil.) Continuing this line of considerations, we can ask what about those variants of P automata, which communicate with their environment only by dynamically emerging request, that is, not in each computation step, but if it is necessary. This aspect would lead to asynchronous systems. Another possibility is, to assume communication as a tool for keeping equilibrium, keeping balance in the system. To do this, for example, we can pre-define some constraints that the regions should satisfy (this would mean a balanced situation), and then we can study which amount and which type of communication with the outside world is necessary to satisfy these conditions. Moreover, we can study the robustness or the tolerance of the P system, that is, how much and which kind of changes are caused in the behaviour of the system (in the sequence of its configurations) taking different amount of multisets imported or to be imported from outside into account. In most of the models, an infinite supply of objects is supposed to be found in the environment. The question, what about systems with finite, but dynamically changing (evolving) supplies of objects would also be interesting.

References

- [1] Alhazov, A. Minimizing Evolution-Communication P Systems and EC P Automata. In: Cavaliere, M., Martín-Vide, C., Păun, Gh. (eds): Brainstorming Week on Membrane Computing, Technical Report 26/03 of the Research

Group on Mathematical Linguistics, Rovira i Virgili University, Tarragona, Spain (2003) 23-31

- [2] Bel-Enguix, G., Gramatovici, R.: Active P Automata and Natural Language Processing. In: Alhazov, A., Martín-Vide, C., Păun, Gh. (eds.): Workshop on Membrane Computing, WMC-2003, Tarragona, July 17-22, 2003. Technical Report 28/03 of the Research Group on Mathematical Linguistics, Rovira i Virgili University, Tarragona, Spain (2003) 61-71
- [3] Bel-Enguix, G., Gramatovici, R.: Parsing with Active P Automata. In: Martín-Vide, C., Mauri, G., Păun, Gh., Rozenberg, G., Salomaa, A. (eds.): Membrane Computing. International Workshop WMC 2003. Lecture Notes in Computer Science, Vol. 2933. Springer, Berlin (2004) 203-217
- [4] Cienciala, L., Ciencialova, L.: P Automata with Priorities. In: Alhazov, A., Martín-Vide, C., Păun, Gh. (eds.): Workshop on Membrane Computing, WMC-2003, Tarragona, July 17-22, 2003. Technical Report 28/03 of the Research Group on Mathematical Linguistics, Rovira i Virgili University, Tarragona, Spain (2003) 161-168
- [5] Csuhaaj-Varjú, E., Ibarra, O.H., Vaszil, Gy.: On the Computational Complexity of P Automata. Conference DNA 10, accepted.
- [6] Csuhaaj-Varjú, E., Vaszil, Gy.: P Automata. Păun, Gh., Zandron, C. (eds.): Pre-Proceedings of the Workshop on Membrane Computing WMC-CdeA 2002, Curtea de Argeş, Romania, August 19-23, 2002. Pub. No. 1 of MolCoNet-IST-2001-32008 (2002) 177-192
- [7] Csuhaaj-Varjú, E., Vaszil, Gy.: P Automata. Păun, Gh., Rozenberg, G., Salomaa, A., Zandron, C. (eds.): Membrane Computing. Lecture Notes in Computer Science, Vol. 2597. Springer, Berlin (2003) 219-233
- [8] Freund, R., Kari, L., Oswald, M., Sosik, P.: Computationally Universal P Systems without Priorities: Two Catalysts Are Sufficient. To appear.
- [9] Freund, R., Martín-Vide, C., Obtulowicz, A., Păun, Gh.: On Three Classes of Automata-like P Systems. In: Ésik, Z., Fülöp, Z. (eds.): Developments in Language Theory. 7th International Conference, DLT 2003, Szeged, Hungary, July 2003. Proceedings. Lecture Notes in Computer Science, Vol. 2710. Springer, Berlin (2003) 292-303
- [10] Freund, R., Oswald, M.: A Short Note on Analysing P Systems. Bulletin of the EATCS **78** (October 2002) 231-236
- [11] Freund, R., Oswald, M.: P Systems with Conditional Communication Rules Assigned to Membranes. In: Alhazov, A., Martín-Vide, C., Păun, Gh. (eds.): Workshop on Membrane Computing, WMC-2003, Tarragona, July 17-22, 2003. Technical Report 28/03 of the Research Group on Mathematical Linguistics, Rovira i Virgili University, Tarragona, Spain (2003) 231-240

- [12] Freund, R., Oswald, M., Sosik, P.: Reducing the Number of Catalysts Needed in Computationally Universal Systems without Priorities. In: Csuhaj-Varjú, E., Kintala, C., Wotschke, D., Vaszil, Gy. (eds.): Fifth International Workshop Descriptive Complexity of Formal Systems. MTA SZTAKI, Budapest (2003) 102-113
- [13] Freund, R., Oswald, M., Staiger, L.: ω - P Automata with communication rules. In: Alhazov, A., Martín-Vide, C., Păun, Gh. (eds.): Workshop on Membrane Computing, WMC-2003, Tarragona, July 17-22, 2003. Technical Report 28/03 of the Research Group on Mathematical Linguistics, Rovira i Virgili University, Tarragona, Spain (2003) 252-265
- [14] Freund, R., Oswald, M., Staiger, L.: ω - P Automata with communication rules. In: Martín-Vide, C., Mauri, G., Păun, Gh., Rozenberg, G., Salomaa, A. (eds.): Membrane Computing, International Workshop WMC 2003. Lecture Notes in Computer Science, Vol. 2933. Springer, Berlin (2004) 203-217
- [15] Ibarra, O. H.: The Number of Membranes Matters. In: Alhazov, A., Martín-Vide, C., Păun, Gh. (eds.): Workshop on Membrane Computing, WMC-2003, Tarragona, July 17-22, 2003. Technical Report 28/03 of the Research Group on Mathematical Linguistics, Rovira i Virgili University, Tarragona, Spain (2003) 273-285
- [16] Ibarra, O. H.: The Number of Membranes Matters. In: Martín-Vide, C., Mauri, G., Păun, Gh., Rozenberg, G., Salomaa, A. (eds.): Membrane Computing, International Workshop WMC 2003. Lecture Notes in Computer Science, Vol. 2933. Springer, Berlin (2004) 218-231
- [17] Ibarra, O. H.: On the Computational Complexity of Membrane Systems. To appear in Theoretical Computer Science.
- [18] Krithivasan, K.: P automata with Tapes. In: Cavaliere, M., Martín-Vide, C., Păun, Gh. (eds.): Brainstorming Week on Membrane Computing, Technical Report 26/03 of the Research Group on Mathematical Linguistics, Rovira i Virgili University, Tarragona, Spain (2003) 216-225
- [19] Madhu, M., Krithivasan, K.: On a Class of P Automata. Submitted
- [20] Oswald, M.: P Automata. PhD dissertation, Technical University of Vienna, 2003.
- [21] Oswald, M., Freund, R.: P Automata with Membrane Channels. In: Sugisaka, M., Tanaka, H. (eds): Proc. of the eight Int. Symp. on Artificial Life and Robotics, Beppu, Japan (2003) 275-278
- [22] Păun, Gh.: Computing with membranes. TUCS Report 208, Turku Centre for Computer Science, 1998.
- [23] Păun, Gh.: Computing with membranes. An introduction. Bulletin of the EATCS **67** (1999) 139-152

- [24] Păun, Gh.: Computing with membranes. *Journal of Computer and System Sciences* **61**(1) (2000) 108-143
- [25] Păun, Gh.: Computing with membranes (P Systems): Twenty six research topics. CDMTCS Technical Report 119, Univ. of Auckland, 2000.
- [26] Păun, Gh.: Membrane Computing. An Introduction. Springer Verlag, Berlin-Heidelberg, 2002.
- [27] Păun, Gh., Rozenberg, G.: A guide to membrane computing. *Theoretical Computer Science* **287** (2002) 73-100
- [28] The *P* systems web page at <http://psystems.disco.unimib.it>