# Chapter 1 : P Systems with Symport/Antiport Rules: The Traces of Objects | Paun Mihai - blog.quintoapp.c

*P systems with symport/antiport are one of the most studied models in membrane blog.quintoapp.com simple and elegant way to operate has caught the interest of several researchers and many papers have been written on this model or inspired by the operations used by it.*

We continue here the study of those P systems where the computation is performed by the communication of objects, that is, systems with symport and antiport rules. Instead of the number of objects collected in a specified membrane, as the result of a computation we consider the itineraries of a certain object through membranes, during a halting computation, written as a coding of the string of labels of the visited membranes. The family of languages generated in this way is investigated with respect to its place in the Chomsky hierarchy. When the symport and antiport rules are applied in a conditional manner, promoted or inhibited by certain objects which should be present in the membrane where a rule is applied, then a characterization of recursively enumerable languages is obtained; the power of systems with the rules applied freely is only partially described. Chomsky hierarchy, membrane computing, P system 1. We introduce here a novel approach to the result of a computation in a P system, looking not for certain objects, present in a certain membrane or sent to the environment during a computation , but to the traces of a certain object, that is, to the string of labels of the membranes visited by this object. In short, we have a membrane structure, consisting of several membranes embedded in a main membrane called the skin and delimiting regions where multisets of certain objects are placed Figure 1 illustrates these notions ; the objects evolve according to given evolution rules, which are applied nondetermin- Grammars 5: Printed in the Netherlands. The objects can also be communicated from a region to another one. In this way, we get transitions from a configuration of the system to the next one. A sequence of transitions constitutes a computation; to each halting computation we associate a result, the number of objects from a specified output membrane. Details can be found at the web address: The P systems we consider here model the real life phenomenon of trans- membrane transport in pairs of chemicals. When two chemicals can pass together through a membrane in the same direction, the process is called symport. When the two chemicals pass only with the help of each other, but in opposite directions, one says that we have antiport. Technically, the rules used in P systems as models of these biological processes are of the form x, in and x, out , as models of symport, and x, out; y, in as a model of antiport, where x, y are strings of symbols representing multisets of chemicals. Of course, this is a generalization of what happens in biology, where mainly pairs of chemicals are coupled. Because such systems work with multisets of objects, it is natural that the result of a computation is a number or a vector of numbers , describing the multiset of objects present at the end of a computation in a specified output membrane. In the present paper we propose another idea: In this manner, a P system will generate a language. Such a characterization is obtained when using the rules conditionally, in the presence of certain promoters or inhibitors: Other research topics are also pointed out. A membrane structure is pictorially represented by an Euler-Venn diagram like the one in Figure 1 ; it can be mathematically represented by a tree or by a string of matching parentheses associated in a standard manner with a tree. A multiset over a set X is a mapping M: Here we always use multisets over finite sets X that is, X will be an alphabet. Such a device is a construct: V is the alphabet of chemicals we call them objects ; 2. In the case of x, in , the multiset of objects x enters the region defined by the mem- brane, from the immediately upper region; this is the environment when the rule is associated with the skin membrane. In the case of x, out , the objects specified by x are sent out of membrane i, into the region immediately outside; this is the environment in the case of the skin membrane. The use of a rule x, out; y, in means expelling from membrane i the objects specified by x at the same time with bringing in membrane i the objects specified by y. The objects from E are supposed to appear in arbitrarily many copies in the environment because we only move objects from a membrane to another membrane, hence we do not create new objects in the system, we need a supply of objects in order to compute with arbitrarily large multisets. The rules are used in the nondeterministic maximally parallel manner specific to P systems with symbol-objects. In this way, we obtain transitions between the configurations of the system. A configuration is described by the m-tuple of the multisets of objects present in the m regions of the system, as well as the multiset of objects which were sent out of the system during the computation, others than the objects appearing in the set E; it is important to keep track of such objects because they appear in a finite number of copies in the initial configuration and can enter again the system. We do not need to take care of the objects from E which leave the system because they appear in arbitrarily many copies in the environment the environment is supposed inexhaustible: The initial configuration is w1 ,. Therefore, a transition means a redistribution of objects among regions and environment , which is maximal for the chosen set of rules. A sequence of transitions between configurations of the system constitutes a computation; a computation is successful if it halts, i. The result of a successful computation is the number of objects present within the membrane with the label io in the halting configuration. A computation which never halts yields no result. Also, we use NRE to denote the family of recursively enumerable sets of natural numbers that is, the family of the length sets of recursively enumerable languages. The rules associated with a membrane i can also be used in a conditional manner, depending on the contents of the region enclosed by membrane i. Specifically, a symport rule with a promoter is of the form x, in b or x, out b , where x is a string representing a multiset of objects and b is an object. The meaning is that the multiset represented by x enters or exits, respectively, the membrane only if b is present in the membrane. That is, b acts as a promoter of the rule. We say that we have a permitting context use of the rule. Similarly, we can have antiport rules with promoters, of the form x, out; y, in b , with the obvious meaning. When applying a rule x, out b or x, out; y, in b in a region i, we must have present the multiset xb, that is, an object is not considered as a promoter of a rule if it evolves by the same rule. Dually, the object b can be used as an inhibitor: We say that the rules are used in the forbidding context mode. When no symport or antiport rule has a pro- moter, then we write sym instead of psym resp. In the case of using rules in the forbidding context mode we write NP Pm f symp , f antiq for the family corresponding to NP Pm psymp , pantiq. The use of the symport rules in the conditional manner is rather useful: Specifically, we consider P systems of the form: PAUN of objects present in arbitrarily many copies in the environment, and R1 ,. The traveler is present in exactly one copy in the system, that is, w1. Before starting to investigate the power of our machineries, let us examine an example. From membrane 4, all copies of d are carried into membrane 5; the computation stops with the traveler in membrane 4. Clearly, this language is not a context-free one. The result for the context-free case can be slightly improved. It is easy to see that: In these proofs, one starts from a matrix grammar with appearance checking â€" see definitions in the next section having only one terminal symbol, a, and one constructs P systems such that the only rules associated with the output membrane hence in the set Rio and involving the symbol a are of the form a, in. In this way, if we denote by 1RE the family of one-letter recursively enumerable languages, from Theorems 2. Note that these results do not improve the assertion from Theorem 4. The passing from languages over the one-letter alphabet to languages over an arbitrary alphabet does not seem trivial. Actually, the number of symbols appearing in the strings of a language obviously induces an infinite hierarchy with respect to the number of membranes used by a P system able to generate that language: Otherwise stated, the hier- archies on the number of membranes are infinite for all types of systems and irrespective of the weights of rules. A partial result in this respect is the following one: We only have to prove the inclusion, its properness is known from The- orems 4. Assume that we have k rules in P , labeled with r1 ,. The carrier will bring states of A from the environment and place them in membranes 5, 6,. Note that only s0 is present in the skin membrane â€" and from now on, only one state will be available in the skin membrane. Assume that in the skin membrane we have, together with t, a state s such that ri: This simulates the use of the rule ri in the automaton A: It remains a topic for a further research to see how far we can go in this way. Characterizations of Recursively Enumerable Languages At the price of using antiport rules of an arbitrary weight depending on the num- ber of symbols appearing in the strings of our language , as well as promoters or inhibitors, we can generate all recursively enumerable languages. In the proofs of these results, we essentially use the characterization of re- cursively enumerable languages by means of matrix grammars with appearance checking. The rules of a matrix are applied in order, skipping the rules in F that cannot be applied â€" therefore we say that these rules are applied in the appearance checking mode. The family of languages of this form is denoted by MATac. The unique matrix of type 4 is used only once, in the last step of a derivation. Membranes 1 and 2 are used in order to generate the language val L. Here are the details of this process. We start with XA in the skin membrane and gt in membrane 2 , corresponding to the start matrix of G. In this way, the matrix is correctly simulated. The symbols c and d enter immediately membrane 2, d in exchange with g, hence from now on the rule a, in g cannot be used any more. The result is obtained as the number of occurrencs of the symbol b. At the same time, the remainder is obtained in the form of a symbol ei. The same result is obtained if we do not have sufficient copies of b or ei in the skin membrane: Specifically, we use the symbol g, placed initially in membrane 2, in a forbidding mode: After finishing the simulation of a derivation in G, the symbol Z introduces the symbols c and d into the system, and d will entail the removing of g from membrane 2 and will also prevent the entrance of any further copy of a. From now on we work in the forbidding mode in the same way as we have proceeded in the previous proof in the permitting mode. Consequently, we have obtained the following result. Can the results from Theorems 5. Final Remarks We have introduced a new way of defining the result of a computation in a P system with symport and antiport rules, namely taking into consideration the trace of a specified object through membranes during a computation. This idea is attractive because it leads to a string associated with a computation rather than a number we still work with multisets of symbol-objects, but the result is qualitatively different, as it is a string. Also, we find that the power of P systems with the output defined in this way is rather large. The relevance of cell membranes for P systems. General aspects, Fundamenta Informaticae,  Membrane systems with coupled transport:

## Chapter 2 : (M. Cavaliere, D. Genova) P Systems with Symport/Antiport of Rules

*Cell-like P systems where communication between the regions are carried out by rules of type symport/antiport are considered. These systems compute by changing the places of objects with respect to the membranes, and not by changing the objects themselves.*

This is an open access article distributed under the Creative Commons Attribution License , which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. In this work, cell separation, inspired from membrane fission process, is introduced in the framework of tissue P systems with evolutional communication rules. The computational complexity of this kind of P systems is investigated. It is proved that only problems in class P can be efficiently solved by tissue P systems with cell separation with evolutional communication rules of length at most , for each natural number. In the case where that length is upper bounded by , a polynomial time solution to the SAT problem is provided, hence, assuming that a new boundary between tractability and NP-hardness on the basis of the length of evolutional communication rules is provided. Finally, a new simulator for tissue P systems with evolutional communication rules is designed and is used to check the correctness of the solution to the SAT problem. Introduction A cell is the basic unit of biological organization that constitutes all living organisms. There are many different types of biological cells, which have different specialized functions that maintain an organism working properly. A large number of theoretical models were proposed [ 4 â€" 6 ] and they have been used to solve various real problems [ 7 â€" 10 ]. These distributed and parallel computation models investigated in membrane computing are called P systems. In general, there exist two main families of P systems: An overview of membrane computing can be found in [ 13 ]. The present work deals with tissue-like P systems. Since the notion of tissue P systems was proposed, numerous research topics have been arisen [ 15 â€" 18 ], and various ingredients energy, catalyst, mitosis, etc. In [ 19 ], tissue P systems with channel states controlling the communication between two regions were proposed, and several Turing universality results were achieved, where the systems work in a maximally parallel way with sequential behavior on channels. In [ 20 ], a general model of tissue P systems with channel states that allow us to model hybrid cooperating grammar systems was considered, where the results were established for strings and arrays. Tissue P systems have been used to find polynomial time solutions to NP-complete problems. In [ 21 ], membrane division rules used in P systems with active membranes have been introduced into tissue P systems yielding tissue P systems with cell division, and a polynomial time uniform solution to the SAT problem was shown. Since then, tissue P systems with cell division were also considered to solve other NP-complete problems: Cell division rules have a replication functioning; that is, two new created cells by a division rule have exactly the same objects except for at most a pair of different objects. Inspired from membrane fission process, cell separation rules are another way to obtain an exponential workspace in polynomial time, but they do not have the duplication functioning; that is, when a cell is separated, the objects in the cell are distributed in each of the newly created cells. Tissue P systems with cell separation have also been used to solve NP-complete problems in polynomial time; one can refer to [ 25 , 26 ] for these investigations. Computational complexity theory in the framework of tissue P systems was introduced in [ 21 ] and it has been studied in [ 27 â€" 30 ]. It was shown that in the framework of tissue P systems with cell division, only tractable problems can be efficiently solved by using communication rules with length at most one [ 31 ] the length of such a rule is the total number of objects involved in it , but a uniform polynomial time solution to the HAM-CYCLE problem by a family of such P systems using communication rules with length at most two has been given [ 30 ]. On the other hand, in the framework of tissue P systems with cell separation, by using communication rules with length at most two, only tractable problems can be efficiently solved, but the SAT problem can be solved by this kind of P systems using communication rules with length at most three [ 26 ]. It is shown that a limit on the efficiency of TESAD P systems is provided with evolutional communication rules of length at most 2. However, it is still an open problem as formulated in [ 33 ] related to the role of evolutional communication rules in tissue P systems with cell separation from a computational complexity point of view. During those computational complexity studies for new variants of P systems, the solutions designed for NP-complete problems are frequently difficult to follow, and requerying makes sure that the evolution of the systems is exactly as expected. In this context, the aid of computer tools to assist in both the design and verification tasks may be crucial, producing much more reliable solutions. In this sense, the development of P-Lingua [ 34 â€” 36 ] implied a significant progress. This open source framework includes a standard language aiming to specify the elements of different types of P systems using a notation very close to the researchers in membrane computing community. Besides, it contains simulation engines for a number of P system types. On top of that, MeCoSim [ 37 , 38 ] provides an additional layer of abstraction with a visual application where researchers can explore at a higher level the evolution of their solutions based on P systems. Contributions of the present work are summarized as follows: We further show that the SAT problem can be solved in polynomial time by a family of systems from TSEC , hence, assuming that , a new boundary between tractability and NP-hardness on the basis of the length of evolutional communication rules, is provided. By using the software MeCoSim, we can analyse the designed P systems, then run the simulation, and obtain the computation results. An alphabet is a nonempty set. Any sequence of elements from is called a string over. The length of , is the number of occurrences in of symbols from A multiset over an alphabet is a function to the set of natural numbers. The multiplicity of a symbol in the multiset.

# Chapter 3 : The Computational Complexity of Tissue P Systems with Evolutional Symport/Antiport Rules

*of rules is a P system with simple evolution rules (used to evolve the symbol-objects in the classical way but without communication targets) and symport/antiport rules that are used, during the computation, to move the evolution rules across the membranes.*

Biological motivations The abstract model of computation described in this article has been inspired by the protein-mediated transport present in living cells. In these cells one of the functions of the plasma membrane is to maintain the internal composition of the cell. This membrane forms a barrier that blocks the free exchange of most biological molecules between the cytoplasm and the environment. Facilitated diffusion is performed by specific transport proteins, channel and carrier proteins, mediating the selective passage of molecules across the membrane. Carrier proteins bind specific molecules to be transported and undergo conformational changes that allow the molecules to pass across the membrane. This transport is performed by see Figure 1: Within each compartments there may be objects evolving and moving to neighbouring membranes following rules specified for the particular system. Recent overviews Frisco P. This means that the system contains unstructured objects that are not rewritten or changed in any other way. A configuration of the system is given by a finite multiset for each of the compartments; each compartment contains a finite number of objects, whereas the objects initially present in the environment are assumed to have infinite unbounded supply. The degree of a system is the number of compartments it has while the weight of a system is a pair of numbers given by the maximum weights for the symport and antiport present in the system. If a system does not have any symport, then the maximum weight for symports is 0; similarly for antiports. Computational steps consist of the application of a multiset of these rules, under the requirement usual in membrane computing of maximal parallelism: A computation is successful if it starts in the initial configuration specified as a multiset of objects for each membrane, and an infinite supply of objects in the environment and if it ends in a halting configuration, i. The result of a successful computation is the number of objects present in a designated membrane, the output membrane. By convention this membrane is elementary, i. We remind the reader that the system works under the requirement of maximal parallelism: What can this system do? In the following we will assume that this rule is not applied for a few transitions. What do we have now? Here we just described one possible sequence of applied rules computation for the system in Figure 2 , but in general, what does this system do? It should not be too difficult to understand that when this system halts that is, when no rule can be applied , then: In the previous section we said that the result of a computation is the number of objects present in a designed membrane called output membrane and that this output membrane has to be elementary. What is the class of numbers that can be generated by these systems? How this classes of numbers change is one imposes restrictions on the systems? This kind of research is part of formal language theory, a branch of theoretical computer science. Even if we will keep the current presentation informal, readers who are not familiar with formal language theory are advised to read Rozenberg and Salomaa if they want to make more sense of what follows. Minimal degree and weight This section summarises the known answers to the following question: Readers interested in these details can refer to the given references on P systems.

# Chapter 4 : P systems with symport-antiport - Scholarpedia

*P Systems with Symport/Antiport and Time Hitesh Nagda1, Andrei P aun1; 2, and Alfonso Rodr guez-Pat on 1 Department of Computer Science/IfM, Louisiana Tech University P.O. Box , Ruston, LA.*

# Chapter 5 : P systems with symport/antiport rules: When do the surroundings matter? - CORE

*The family of all sets N(Æ’) computed by P systems with symport/antiport with maximal parallelism of degree at most m, using symports of weight at most x and antiports of weight at most y is denoted by N Â¢ PP m (sym x ; anti y).*

## Chapter 6 : Cell-Like P Systems With Channel States and Symport/Antiport Rules.

*Reversible P Systems with Symport/Antiport Rules dual P systems with determinism leads reversibility. In [2], reversible P systems with symport/antiport rules are considered.*