

A Note on the Rough P Systems

Amged Fathey, Amr Badr, and Ibrahim Farag

Abstract—A new variant of P Systems is considered, a rough p system which discussed before as an open problem. The proposed rough p system definition is based on boundary rules and on conditional communication, where communication is controlled by the contents of the strings not by the evolution rules for obtaining these strings.

Index Terms—P systems, membrane computing, rough set, rough P systems.

I. INTRODUCTION

P systems are classes of distributed parallel computing models inspired from the way the living cells process chemical compounds, energy, and information. Many variants of P systems use string objects and context-free rules for processing them. Rewriting P systems with string objects were introduced in [1]. Several variants of P systems with string objects have also been investigated extensively. In this work, we concentrate on Rough P systems which introduced as open problem in [2], [3].

II. MEMBRANE COMPUTING

Membrane computing is a branch of natural computing with an initial goal of abstracting computing models from the structure and the functioning of living cells [1]. The initial goal was learning something that is possibly useful to computer science from cell biology, and the area quickly developed in this direction [1], [4].

Membrane structure (which is the basic idea of building a membrane system) is a structure composed of several cell-membranes, hierarchically embedded in a main membrane called the skin membrane [5]. A plane representation of a membrane structure, as in Fig. 1, can be given by means of a Venn diagram, without intersected sets and with a unique superset. The membranes delimit regions and we associate with each region a set of objects, Described by some symbols over an alphabet, and a set of evolution rules [1], [6].

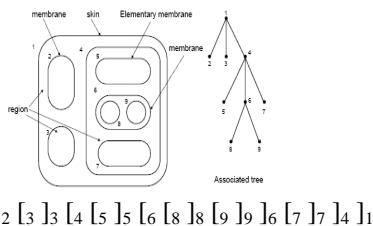


Fig. 1. Membrane structures.

Membrane Computing Computation

The computation mainly starts with an initial configuration of the system, where the input data of a problem is encoded [7].

A computation for a P system can be described as follows:

$$C_0 \Rightarrow C_1 \Rightarrow \dots \Rightarrow C_n, n \geq 0,$$

where C_0 is the initial configuration.

The transition from one configuration to another is performed by applying rules to the objects placed inside the regions [1]. A computation of the system is a tree of configurations, the rules is applied in a non-deterministic maximal parallel manner (all objects which can be evolved by such rules that have to evolve), in each step, in each region; each object that can be evolved according to some rule must do it made by transitions until reaching a halting configuration, where no more rules may be applied. The result of a halting computation is usually defined through the multi-set associated with a specific output membrane, or the environment, which is the space out of the skin [1].

Fig. 2 gives an example of a P system that computes the squares of natural non-null numbers (the output is read in membrane 1, which should be elementary at the end of a computation) [1].

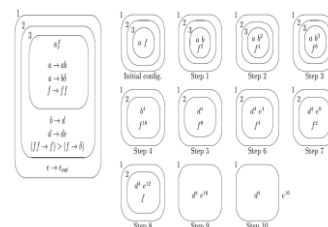


Fig. 2. A P system computes squares of natural non-null numbers.

III. ROUGH SET THEORY – FUNDAMENTALS

Rough set theory was developed by Zdzislaw Pawlak [9], [10], [11] in the early 1980's. It deals with the classificatory analysis of data tables. The data can be acquired from measurements or from human experts. The main goal of the rough set analysis is the synthesize approximation of concepts from the acquired data. Rough sets constitute a sound basis for KDD (Knowledge Discovery in Databases). It offers mathematical tools help to discover patterns hidden in data.

It can be used for feature selection, feature extraction, data reduction, decision rule generation, and pattern extraction (templates, association rules). Recent extensions of rough set theory have been developed as new methods for decomposition of large data sets, data mining in distributed and multi-agent systems, and granular computing.

Let $T = (U, A)$ and let $B \subseteq A$ and $X \subseteq U$. We can

Manuscript received March 25, 2012; revised May 6, 2012. The authors are with the Computer Science Department, Faculty of Computer Science, MTI University, Cairo, Egypt (e-mail: amged_fathey@hotmail.com, a.badr.fci@gmail.com).

approximate X using only the information contained in B by constructing the B-lower and B-upper approximations of X, denoted \underline{BX} and \overline{BX} respectively, where

$$\underline{BX} = \{x \mid [x]_B \subseteq X\},$$

$$\overline{BX} = \{x \mid [x]_B \cap X \neq \emptyset\}.$$

B-boundary region of X,

$$BN_B(X) = \overline{BX} - \underline{BX},$$

Consists of those objects that we cannot decisively classify into X in B.

B-outside region of X, $U - \overline{BX}$, consists of those objects that can be certified and classified as not belonging to X.

A set is said to be **rough** if its boundary region is non-empty, otherwise the set is crisp.

IV. ROUGH P SYSTEMS PROBLEM

In a P system, For instance, rough set theory helps to determine which rules should be used for evolving the objects from the border, and the determination of participation in the same (cooperative) rules with objects from the lower approximation and/or from outside the upper approximation or not [8].

In the P systems area only the idea of having an approximate knowledge of the content of a region: some objects are surely inside, others are surely outside, while for other objects we do not know where they are, they can be inside or outside the membrane with the same probability. Thus, a border region should be considered, as suggested in Figure 3.

We have here membrane i , placed inside region j . In the “surely-inside region” we use the rules from R_i , in the “surely-outside region” we use the rules from R_j , but objects from the border should be processed by both rules of R_i and R_j . Thus, the sets of rules are associated with the sure regions, while for borders we use the union of the adjacent sets of rules.

Moreover, the communication should be done in two steps as suggested in figure 3 and also used in P systems with inter-regions [1]

Note the important difference from the case from [1] that here we do not have a separate set of rules of the border (inter-region) the border is not a region, but a sort of “no men land”, where all neighboring rules can be applied.

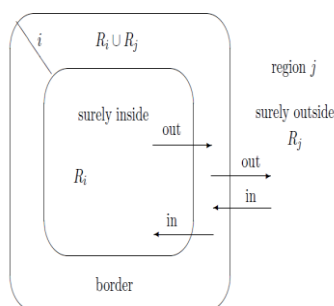


Fig. 3. Rules and regions.

V. ROUGH P SYSTEMS

We note that: the problem of rough p systems is to determine the rules *which* should be used here and *how* to be used.

The idea is making use of BP systems to make the rules related to borders and regions and in the same time the communication is done through some conditions to limit (optimize) the behavior of objects, which is the rough p system problem.

So the two appeared questions are: which rule and when to start communication?

By this way we answered the two questions “which” and “how”, “which” will be determined through the BP systems and “how” will be answered through conditional communications

Before starting the definition of the proposed Rough P system, we must define the main entities of the idea which are P System with Boundary and rewriting P systems with conditional communication.

In the P systems with boundary rules (PB systems), communication is allowed by rules of the form $x [i]y \rightarrow x' [i]y'$. the rule means that if the membrane i contains the multiset y and outside the membrane i is the present multiset x , then a communication can be established producing the multiset x' , y' outside and inside the membrane i , respectively. In the other side the transformation rules will be written in the form $[i]y \rightarrow [i]y'$.

A formal P System with Boundary Rules can be found [12].

In the rewriting P systems with conditional communication was introduced in [13]. This is achieved by considering certain types of *permitting* and *forbidding* conditions, based on the symbols or the substrings (arbitrary, or prefixes/suffixes) which appear in a given string.

According to PB systems and conditional communication a *rough p system* can be defined as follows.

Definition: Rough P System is a construct: (of degree $m \geq 1$) is a construct

$$\Pi = (V, T, \mu, M_1, \dots, M_m, P_1, F_1, \dots, R_m, P_m, F_m);$$

where:

- 1) V is the alphabet;
- 2) $T \subseteq V$ is the terminal alphabet;
- 3) μ is the membrane structure;
- 4) M_1, \dots, M_m are finite languages over V , representing the strings initially present in the m regions;
- 5) R_i, \dots, R_m are finite sets of context-free rules over V of the following two forms:
 - $xx' [i]y'y \rightarrow xy' [i]x'y$, for $x, y, x', y' \in V^*$ and $1 \leq i \leq m$ (communication rules),
 - $[i]y \rightarrow [i]y'$, for $y, y' \in V^*$ and $1 \leq i \leq m$ (transformation rules),
- 6) P_i and F_i are permitting and forbidding conditions associated with the regions;

The conditions may be formatted as the following:

- 7) *empty*: no restriction is imposed on strings,
 - empty permitting condition by $(true, X)$, $X \in \{in; out\}$
 - empty forbidding condition by $(false, notX)$, $X \in \{in, out\}$

8) *symbols checking*:

- Each P_i is a set of pairs (a, X) , $X \in \{in; out\}$, for $a \in V$
- Each F_i is a set of pairs $(b, notX)$, $X \in \{in; out\}$, for $b \in V$
- A string w can go to a lower membrane only if there is a pair $(a; in) \in P_i$ with $a \in alph(w)$ and for each $(b, notin) \in F_i$ we have $b \notin alph(w)$
- Sending the string w out of membrane I , it is necessary to have $a \in alph(w)$ for at least one pair $(a, out) \in P_i$ and $b \notin alph(w)$ for all $(b, notout) \in F_i$

9) *substring checking*:

- Each P_i is a set of pairs (u, X) , $X \in \{in, out\}$, for $u \in V^+$
- Each F_i is a set of pairs $(v, notX)$; $X \in \{in, out\}$, for $v \in V^+$
- A string w can go to a lower membrane only if there is a pair $(u, in) \in P_i$ with $u \in Sub(w)$, and for each $(v, notin) \in F_i$ we have $u \in Sub(w)$
- Sending the string w out of the membrane i it is necessary to have $u \in Sub(w)$ for at least one pair $(u, out) \in P_i$ and $v \notin Sub(w)$ for all $(v, notout) \in F_i$

1) *Prefix/suffix checking*: exactly as in the case of substrings checking, with the checked string being a prefix or a suffix of the string to be communicated.

We assume that we have conditions of the types *empty*, *ymb*, *sub_k*, *pre f_k*, *suff_k*, respectively, where k is the length of the longest string in all P_i, F_i .

The transitions of the system are defined in the following way. In each region, each string which can be rewritten is rewritten by a rule from that region. The rule is applied and the nonterminal it rewrites is non-deterministically chosen. The string obtained in this way is checked against the conditions P_i, F_i from that region. If it fulfills the required conditions, then it will be immediately sent out of the membrane or to an inner membrane, if any exists; if it fulfills both *in* and *out* conditions, then it is sent to a membrane non-deterministically choosing the direction- and non-deterministically choosing the inner membrane in the case when several directinner membranes exist. If a string does not fulfill any condition, or it fulfills only *in* conditions and there is no inner membrane, then the string remains in the same region. If a string cannot be rewritten, then it is directly checked against the communication conditions. That is, the rewriting has priority over communication.

A sequence of transitions form a computation and the result of a halting computation is the set of strings over T sent out of the system. In the case of non-extended systems, all strings sent out are accepted. A computation does not yield a result if it does not halt. A string which remains inside the system or, in the case of extended systems, which exits but contains nonterminal symbols does not contribute to the generated language. The language generated by a system Π is denoted by $L(\Pi)$.

VI. POWER OF ROUGH P SYSTEMS

The power of rough p system can be taken from the power

of both the PB system and conditional communication.

In [12] they proved that PB systems with bidirectional communication performed by rules and only three membranes are able to characterize the family of recursively enumerable sets of vectors of natural numbers $P sRE$.

And also in [12] they proved that when using communication rules which restricted form PB systems they are equals in power with p systems which don't use priority among rules and which don't use any operator (in ,here, out) for modifying membrane structure.

In [13], it was proved that P systems of degree 4 with permitting conditions of type *sub2* and forbidden conditions of type *ymb* are computationally universal. This result has been improved from 4 to 3 membranes in [13]. And in [14] an improvement is done, and shows that universality can be achieved with 2 membranes in this case.

The universality result for P systems with both permitting and forbidden conditions of type *ymb* has been improved from 6 to 5 membranes in [14]. Here we give a universality result with only 3 membranes in [2].

There is a characterization of recursively enumerable languages by P systems with permitting conditions of type *prefsuff2* and forbidden conditions of type *empty* in [13] without a bound on the number of membranes. It was conjectured that such a characterization holds also for a reduced number of membranes. We settle this conjecture in the posi tive here and show that *eight* membranes are enough for achieving the universality.

VII. CONCLUSION AND FUTURE WORK

We introduced here a proposed solution to the open problem of rough p system using both boundary rules and on conditional communication to optimize the boundaries according to the rough set theory, the power of rough p system discussed according to the power of both boundary rules and on conditional communication. It is an open problem whether or not the model can be improved or enhanced.

REFERENCES

- [1] G. Păun, "Computing with Membranes," *Journal of Computer and System Sciences* vol. 61, Issue 1, August 2000, pp. 108-143.
- [2] G. Păun, "Tracing Some Open Problems in Membrane Computing," *Romanian Journal of Information Science and Technology* vol. 10, no. 4, 2007
- [3] M. Cavaliere, Carlos Martin-Vide, and Gheorghe Paun, "Problems Circulated Before the Brainstorming Week on Membrane Computing," Brainstorming Week on Membrane Computing; Tarragona, Feb 5-11 2003.
- [4] A. Adleman, L. M., "On Constructing a Molecular Computer," 1st DIMACS Workshop on DNA Based Computers, Princeton, 1995. In DIMACS Series, vol. 27, pp. 1-21, 1996.
- [5] B. Nadia and Za. Claudio," Computing with Genetic Gates, Protiens and membranes," *7th International Workshop, WMC*. pp. 250-265, 2006.
- [6] G. Paun, Membrane Computing. An Introduction. Springer-Verlag, Berlin.
- [7] G. Paun, "Membrane Computing and Brane Calculi," *Electronic Notes in Theoretical Computer Science* vol. 171, Issue 2, pp. 3-10, 2007.
- [8] G. Păun, "Rough P Systems," Brainstorming Week on Membrane Computing; Tarragona, pp. 9-9, 2003.
- [9] Z. Pawlak, "Rough Sets," *International Journal of Computer and Information Sciences*, vol.11, pp. 341-356, 1982.

- [10] Z. Pawlak, *Rough Sets - Theoretical Aspect of Reasoning about Data*, Kluwer Academic Publishers, 1991.
- [11] B. Nguyen, Hung Son, and Szczuka, Marcin, "A View on Rough Set Concept Approximations," *Fundamenta Informaticae*, vol. 59, pp. 107-118, 2004.
- [12] F. Bernardini and V. Manca, "P Systems with Boundary Rules," In *Preliminary Proceedings of Second Workshop on Membrane Computing*, pp. 97-101, 2002.
- [13] P. Bottoni, A. Labella, C. Martin Vide, and Gh. P. aun, "Rewriting P systems with Conditional Communication," LNCS 2003, Springer, pp. 325-353, 2002.
- [14] H. Ramesh and Raghavan Rama, "Rewriting P Systems with Conditional Communication: Improved Hierarchies," *Computing and Informatics* vol. 27 no.3, pp. 453-465, 2008.