Krassimir Markov, Vitalii Velychko,
Lius Fernando de Mingo Lopez, Juan Casellanos
(editors)

# New Trends in Information Technologies

I T H E A
SOFIA
2010

Krassimir Markov, Vitalii Velychko, Lius Fernando de Mingo Lopez, Juan Casellanos (ed.)

New Trends in Information Technologies

ITHEA®

Sofia, Bulgaria, 2010

ISBN 978-954-16-0044-9

First edition

Recommended for publication by The Scientific Concil of the Institute of Information Theories and Applications FOI ITHEA

This book maintains articles on actual problems of research and application of information technologies, especially the new approaches, models, algorithms and methods of membrane computing and transition P systems; decision support systems; discrete mathematics; problems of the interdisciplinary knowledge domain including informatics, computer science, control theory, and IT applications; information security; disaster risk assessment, based on heterogeneous information (from satellites and in-situ data, and modelling data); timely and reliable detection, estimation, and forecast of risk factors and, on this basis, on timely elimination of the causes of abnormal situations before failures and other undesirable consequences occur; models of mind, cognizers; computer virtual reality; virtual laboratories for computer-aided design; open social info-educational platforms; multimedia digital libraries and digital collections representing the European cultural and historical heritage; recognition of the similarities in architectures and power profiles of different types of arrays, adaptation of methods developed for one on others and component sharing when several arrays are embedded in the same system and mutually operated.

It is represented that book articles will be interesting for experts in the field of information technologies as well as for practical users.

General Sponsor: Consortium FOI Bulgaria (www.foibg.com).

Printed in Bulgaria

**ISBN 978-954-16-0044-9**

C\o Jusautor, Sofia, 2010

# INTELLIGENT P-SYSTEMS

**Alberto Arteta , Angel Luis Castellanos, Jose Luis Sanchez**

***Abstract****: Membrane computing is a recent area that belongs to natural computing. This field works on computational models based on nature's behavior to process the information. Recently, numerous models have been developed and implemented with this purpose. P-systems are the structures which have been defined, developed and implemented to simulate the behavior and the evolution of membrane systems which we find in nature. However no other technology has been used to simulate the behavior of the living cells. In this paper we present a proposal for a set of reactive robots that receives, process and learn from the living cells through the use of p-systems. When analyzing the properties of the proposed robot. we realize that we get interesting points compared to traditional p-systems.*

***Keywords****: Autonomous robots, membrane computing, p-systerms, artificial intelligence.*

## Introduction

Natural computing ia a new field within computer science which develops new computational models. These computational models can be divided into three major areas:.

1. Neuronal networks.
2. Genetic Algorithms
3. Biomolecular computation.

Membrane computing is included in biomolecular computation. Within the field of membrane computing a new logical computational device appears: The P-system. These P-systems are able to simulate the behavior of the membranes on living cells. This behavior refers to the way membranes process information. (Absorbing nutrients, chemical reactions, dissolving, etc)

Membrane computing formally represents, through the use of P-systems, the processes that take place inside of the living cells. In terms of software systems, it is the process within a complex and distributed software. In parallel computational models, p-systems might be as important as the Turing machine is in sequential computational models.

In this paper, we design an autonomous robot that it is capable to simulate the behavior the living cells to solve known problems through the use of p-systems. Although this has been done so far by traditional p-systems [1], we propose a new model. We will design several autonomous robots that behave as the living cells when processing information. This way these robots can obtain solutions to known problems. The interesting part here is that we state that performance improves by reducing computational complexity. Traditional P-systems are the structures used within membrane computing to do the living cells simulation. P-systems evolve in a parallel a non deterministic way. By using an autonomous robot, we will eliminate the non-determinism as the evolution will be orientated to obtain results in a faster way.

In the paper we are going to go through different topics: (1) Introduction to P-systems theory; (2) Description of autonomous robot; (3) Theoretical model of a robot processing information from the living cells; (4) Traditional P-system vs P-systems with robots; (5) Conclusions and further work.

## Introduction to P-systems theory

In this section we will study into detail all of the theories related to the paradigm of the P-systems. A P-system is a computational model inspired by the way the living cells interact with each other through their membranes. The elements of the membranes are called objects. A region within a membrane can contain objects or other membranes. A p-system has an external membrane (also called skin membrane) and it also contains a hierarchical relation defined by the composition of the membranes. A multiset of objects is defined within a region (enclosed by a membrane). These multisets of objects show the number of objects existing within a region. Any object 'x' will be associated to a multiplicity which tells the number of times that 'x' is repeated in a region.

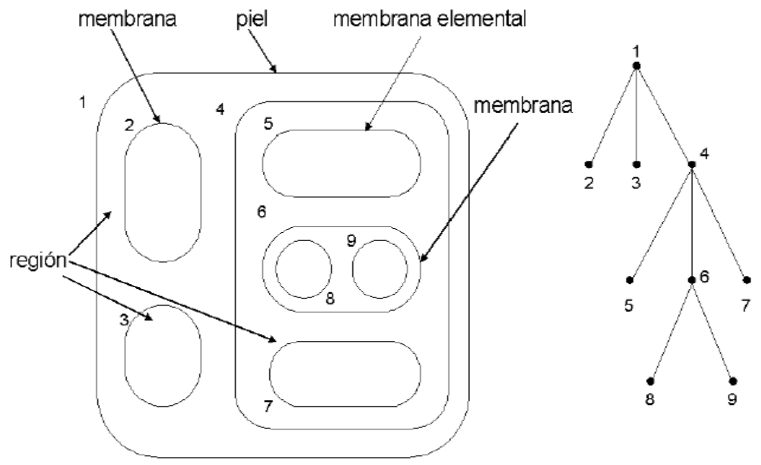

**Fig. 1.** The membrane's structure (left) represented in tree shape (right)

According to Păun 's definition, a transition P System of degree n, n > 1 is a construct: [Păun 1998]

$$\Pi = \left(V, \mu, \omega_1, .., \omega_n, (R_{1,}\rho_1), ..(R_{n,}\rho_n), i_0\right)$$

where:

*V* is an alphabet; its elements are called objects;

μ is a membrane structure of degree n, with the membranes and the regions labeled in a one-to-one manner with elements in a given set ; in this section we always use the labels 1,2,..,n;

$\omega_i\ 1 \le i \le n$, are strings from $V^*$ representing multisets over V associated with the regions 1,2,..,n of μ

$R_i\ 1 \le i \le n$, are finite set of evolution rules over V associated with the regions 1,2,..,n of μ; $\rho_i$ is a partial order over $R_i\ 1 \le i \le n$, specifying a priority relation among rules of $R_i$ . An evolution rule is a pair (u,v) which we will usually write in the form $u \to v$ where u is a string over V and v=v' or v=v'$\delta$ where v' is a string over $\left(V \times \{here, out\}\right) \cup \left(V \times \{in_j\ 1 \le j \le n\}\right)$, and $\delta$ is a special symbol not in. The length of u is called the radius of the rule $u \to v$

$i_o$ is a number between 1 and n which specifies the output membrane of $\Pi$

Let *U* be a finite and not an empty set of objects and N the set of natural numbers. A *multiset of objects* is defined as a mapping:

$$M : V \to \mathrm{N}$$
$$a_i \to u_1$$

Where $a_i$ is an object and $u_i$ its multiplicity.

As it is well known, there are several representations for multisets of objects.

$M = \{(a_1, u_1), (a_2, u_2), (a_3, u_3) \ldots\} = a_1^{u_1} \cdot a_2^{u_2} \cdot a_n^{u_n} \ldots\ldots$

*Evolution rule* with objects in *U* and targets in *T* is defined by $r = (m, c, \delta)$

where $m \in M(V), c \in M(VxT)\ and\ \delta \in \{to\ dissolve, not\ to\ dissolve\}$

From now on *'c'* will be referred to as the consequent of the evolution rule *'r'*

The *set of evolution rules* with *objects* in *V* and targets in *T* is represented by R *(U, T).*

We represent a rule as:

$x \to y \quad or \quad x \to y\delta$ where x is a multiset of objects in M((*V*)xTar) where Tar ={here, in, out} and y is the consequent of the rule. When $\delta$ is equal to “dissolve”, then the membrane will be dissolved. This means that objects from a region will be placed within the region which contains the dissolved region. Also, the set of evolution rules included on the dissolved region will disappear.

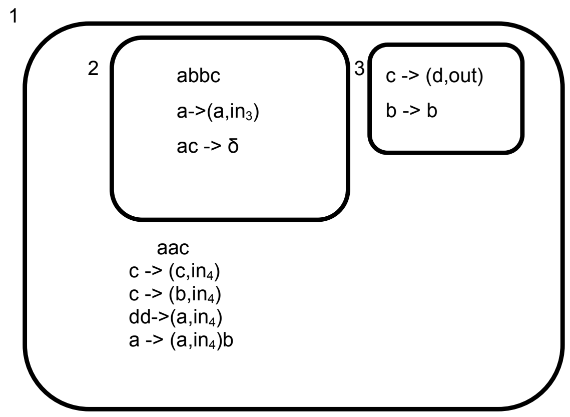

**Fig 2** P-system with 3 regions and multiset of objects in each region

P-systems evolve, which makes it change upon time; therefore it is a dynamic system. Every time that there is a change on the p-system we will say that the P-system is in a new transition. The step from one transition to another one will be referred to as an evolutionary step, and the set of all evolutionary steps will be named computation. Processes within the p-system will be acting in a massively parallel and non-deterministic manner. (Similar to the way the living cells process and combine information). We will say that the computation has been successful if:

1. The halt status is reached.
2. No more evolution rules can be applied.
3. Skin membrane still exists after the computation finishes.

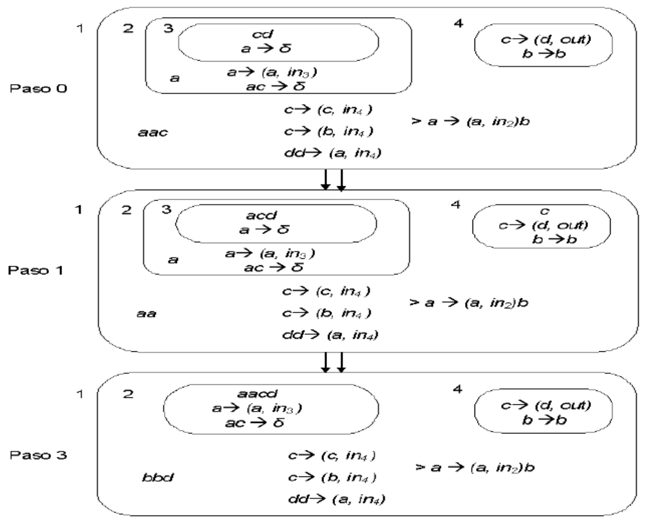

**Fig 3.** Example of the evolution of a p-system

Currently There are some simulations of p-systems in different languages and hardware[Arroyo, 2001], [Arroyo, 2003]

## Autonomous robots

Nowadays humans and robots are used to interact with each other. We could say that robots try to simulate humans' actions. Part of those humans' actions is making decisions. If a robot can make decision, can we certainly state that the robot is intelligent? Today we still have no a unique definition of what intelligence means. What we know is that the concept: "intelligence" [Brooks, 1986] is related to:

- Understanding
- Problem solving
- Learning.

Even if we are able to create robots that solve problems we cannot assure that they understand or learn. That is the reason why the definition of intelligence does not cover all the possible meanings.

Although we cannot say that robots are intelligent, we can certainly establish a classification for them:

In general terms we can separate the robots into two classes:

- Deliberative robots
- Reactive Robots

The first kind is the traditional one. This types of robots work based on scheduled actions and known information.

Reactive robots interact with the environment, they observe, process information, and make decisions based on the environment they are.

From our perspective the second type is the interesting one. These robots can adapt to the environment they live and make decisions based on that. Not only making decisions but also solving problems. Adaptability is other characteristic of intelligence.

Another characteristic of traditional robotics is the use of a centralized system. This centralized system stores all the information of the environment. The information is represented in a symbolic way. After processing the information is possible to calculate the next action. The problem here is when the environment changes constantly. That is why traditional robots become useless in this scenario. The aim is trying to build a robot able to adapt to its environment regardless the type of environment It is. In that way, the robot will have some basic actions and it will create a bigger knowledge by learning from the environment. The learning process comes from the reactions that robots do.

Learning process is a complex one.

If a robot is created to experiment in a lab, it is possible to determine all the situations in where the robot is going to be. On the contrary, if the robot created wants to be useful in the real world, there is no way for us to determine how many different situations the robot is going to face. Thus, we must be able to create a robot that learns and therefore be able to react correctly when a new circumstance arises.

These are the different ways for implementing the learning process.

There are four types of learning.

- Auto-organized: It works with random variables.
- Supervised: any action has different data and variables.
- Hybrid: A combination of the two previous ones
- Feedback: The response from the environment influences on its actions.

The robots we propose will learn by receiving feedback from the environment.

## Robots working as p-systems

Once we have seen the characteristics of p-systems and autonomous robots separately, we are going to propose a community of robots that act as a p-system but let us say as a learning p-system.

The scenario we propose is:

Since we have regions in the living cells we are going to place a robot in every region delimited by a membrane.

The way to do this is:

In a the regions, given a set of membranes $M = \{m_i \mid i \in \mathrm{N}, 1 \le i \le n\}$ where $m_i$ is a membrane, we allocate a robot:

Thus, we will need to define a function that for each membrane or region

$$f_{robot} : M \to A$$
$$f_{robot}(m_i) = R_i \ \forall i \in \mathrm{N} \ i \le n, \quad \text{n number of membranes}$$

The p-system considers three major stages:

1. Static structure of the p-system
2. Dynamic behavior of the p-system
3. Synchronism between membranes.

1. The static structure of the p-systems is updated by the existence of a robot $r_i$

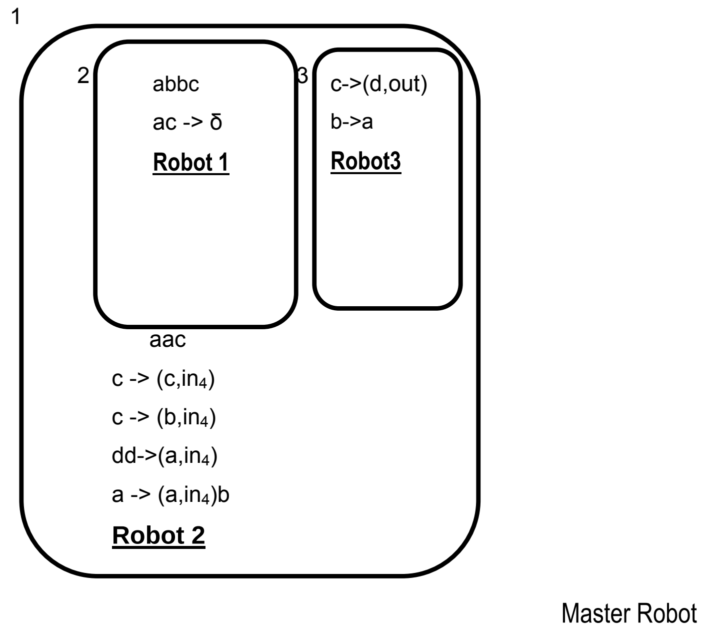

**Fig 4** P-system controlled by reactive robots. The robot outside controls the evolution of the p-system.

Every robot controls a region. The robot is storing the information about the information processing for every region. The p-system evolves following the robots patterns.

For every region the robot store information and makes decisions based on the information is stored. Moreover the robot outside the p-system controls the execution steps of the p-system.

Robots learn from the times that evolution rules are applied and the results obtained.

The aim of this p-system supervised by robots is not just to find solutions to known problems, but also improving performance on that.

According to Paun's model all the rules application processes occur in every region in a parallel manner. Moreover the process is non deterministic.

Every robot $R_i$ collects information about the membrane $m_i$ in where he is allocated.

- The evolution rules that are applied in $m_i$ and number of times that every rule $r_i$ is applied to obtain a maximal multiset [Arteta, 2008]

For every region the robot $R_i$ stores records as: ( $(r_1,1),(r_2,2),..,(r_i,7),..,(r_m,3)$

- The robot outside the p-systems:

  - Process the information arriving from all the robots.
  - It keeps the number of computation steps until the program finishes.
  - Stores information of the communication phase between membranes
  - It tells the robots what decisions to make in order to reduce computation steps. (Deterministic way)

The more time the p-system works the better results are obtained. At the end the robot will tell the robot in each membrane which evolution rules should be applied and also how many times those rules must be applied in order to obtain fast results. This will reduce the execution time.

## Traditional p-systems versus p-systems with robots

Implementing membrane computing with robots shows a few advantages compared to the traditional ones.

The election for the evolution rules to be applied would be taken according the robot's indications. At first, robots will not be very accurate on the elections but during the learning process programs will take shorter to finish and to find solutions. Finding solutions in a faster way requires certain degree of what we call intelligence. P-systems with robots can be referred as intelligent p-systems. Although this is the main advantage we obtain when using p-systems with robots, there are certain disadvantages too:

- The need of auxiliary space to store information about the rules' election
- The need of extra time to calculate the times that the evolution rules have to be applied
- In the beginning the results might take longer than when using traditional p-systems. This might occur because as there are not enough information from the p-system computation, the decisions made by the robot about the rules could be worse than electing rules in a non deterministic way (as the traditional p-systems do)
- When changing non-determinism by intelligent elections, Paun's biological model is not usefull anymore to implement the living cells which means that p-system of robots needs cannot be used to implement the living cells model.

## Conclusions

This paper presents an update of what we know about membrane computing. P-systems are the structures that implement Paun's biological model. Membrane computing has proved to be able to reduce computational complexity when solving known problems as the "*knapsack problem".* What we propose is a new model based on some of the inherent properties of p-systems plus the learning capability which is achieved by a set of autonomous robots. Although non determinism is not a property of our system anymore, we can certainly state that this new model of p-systems can obtain optimal results to known problems due to the capacity of learning.

This idea is in an early stage because there are no formal implementations of membrane computing yet. However the idea is promising. A theoretical p-system that is able to learn by using intelligent systems as robots, can make right decisions when applying evolution rules in p-systems' regions.

A further study is necessary to define formally the new model created by the combination of p-systems and robots.

## Bibliography

[Păun, 1998] "Computing with Membranes", Journal of Computer and System Sciences, 61(2000), and Turku Center of Computer Science-TUCS Report nº 208, 1998.

[Brooks, 1986] Achieving AI through Building Robots [Periodic publication] // AI Memo. - Massachusets : [s.n.], 1986.

[Arroyo, 2001] "Structures and Bio-language to Simulate Transition P Systems on Digital Computers," Multiset Processing

[Arroyo, 2003] "A Software Simulation of Transition P Systems in Haskell, Membrane Computing,"

[A. Arteta, 2008] "Algorithm for Application of Evolution Rules based on linear diofantic equations" Synasc 2008 (IEEE), Timisoara Romania September 2008[1] A. Syropoulos, E.G. Mamatas, P.C. Allilones, K.T. Sotiriades "

## Authors' Information

***Alberto Arteta Albert*** *– Associate professor U.P.M Crtra Valencia km 7, Madrid-28031, Spain; e-mail: aarteta@eui.upm.es*

*Research: Membrane computing, Education on Applied Mathematics and Informatics*

***Angel Luis Castellanos Peñuela*** *– Associate professor U.P.M Ciudad Universitaria, Madrid, Spain; e-mail: angel.castellanos@upm.es*

*Research:, Education on Applied Mathematics and Informatics*

***Jose Luis Sanchez Sanchez*** *– Associate professor U.P.M Crtra Valencia km 7, Madrid-28031, Spain; e-mail: jlsanchez@eui.upm.es*

*Research: Membrane computing*