Computer Science Journal of Moldova, vol.**13**, no.1(37), 2005

Abstracts of Dr. Thesis

Title: Head systems and applications to bio-informatics
Author: Serghei Verlan
Supervisor: Maurice Margenstern, director of LITA, professor at the University of Metz
University: University of Metz
Date of defence: 21 June 2004

The increasing interest for bioinformatics is not astonishing, because this discipline might potentially explain the functioning of some natural processes. One of relations between biology and computer science is the application of methods of computer science in biology. But we can investigate another relation trying to apply biological methods to solve computer science problems. This affirmation is based on the fact that a lot of biological processes, in particular DNA manipulations, may be viewed as information transformations. The study of biologically inspired systems is very exciting and can bring colossal gains.

From one hand, the theory of formal languages is grounded on rewriting operations. From the other hand, the nature uses different operations like copy and paste as well as different data structures. This is why it is very important to reconstruct old computational paradigms in this new framework.

The thesis is devoted to studying of H systems and their extensions.

H splicing systems were introduced by Thomas Head as a new language generating device. This abstract notion is good motivated biologically.

We considered two possible definitions of the splicing: 1-splicing and 2-splicing, and we studied for the first time the relation between classes of languages based on 1-splicing and on 2-splicing. We showed that one family is strictly included into another. We also found several non-trivial examples of splicing languages, and of regular languages that cannot be splicing languages.

We considered time-varying distributed H systems (or TVDH systems) that were introduced by Gh. Paun in 1998. Initially it was showed that 7 components are enough in order to generate all recursively enumerable languages. M. Margenstern and Yu. Rogozhin showed that TVDH systems with 2 components are able to do universal computations and that 2 components are enough to generate all recursively enumerable languages. The same authors showed that TVDH systems with one component are universal and that it is possible to generate all recursively enumerable languages with only one component in a sequential way.

We studied an extension of TVDH systems: enhanced time-varying distributed H systems, or ETVDH systems, which were introduced by M. Margenstern and Yu. Rogozhin in 2000. This model is a small modification of TVDH systems but it introduces more parallelism. Now one component is not enough in order to obtain a big computational power and, in this case, the generated language is limited by the family of regular languages. But two components are sufficient to generate all recursively enumerable languages. We studied the same problem in the parallel context and found solutions, at first with 4 components, after that with 3 components, and, finally, with 2 components that is a minimum.

We considered another extension of H systems which is inspired simultaneously by H systems and distributed grammars: test tube systems (or TT systems), introduced by E. Csuhaj-Varju, L. Kari and G. Paun. The computation in such system consists of two iteratively repeated steps: a computation step and a communication step. The number of tubes necessary to obtain the computational power of a Turing machine was established to two, while systems having one tube can generate only regular languages.

We proposed two variants of these systems: test tube systems with alternating filters and modified test tube systems. These variants differ from the original definition by the communication protocol. We showed that in the first variant two tubes are enough in order to produce all recursively enumerable languages. Moreover, it is possible to formulate rules in such a way that the second tube will be used as a garbage collector only having no associated rules. Therefore the universality is obtained with "one tube and a half"! This result shows also that H systems are already powerful enough and we can go from the regularity to the universality after small modifications.

For most of the systems we showed their equivalence with a formal grammar or a Turing machine. We analyzed previous proofs in the area of splicing systems and developed a new method, the method of directing molecules, which permits to decrease surprisingly the complexity of corresponding systems. We applied this method to TVDH systems, ETVDH systems, test tube systems with alternating filters, modified test tube systems and splicing membrane systems. In all these cases the new method permitted to simplify considerably the proofs.

We studied membrane systems, or P systems, which are a model of computing inspired by the structure and the functioning of a living cell. It was introduced in 1998 by Gh. Paun. We considered several models of computing issued from the combination of these systems and the splicing operation. We showed the structural aspect introduced by the presence of membranes combined with the splicing operation, which gives big computational power to systems.

We showed that the original definition of splicing P systems is not complete, and we proposed several variants in order to improve this lack.

We considered other variants of splicing P systems, non-extended splicing P systems and splicing P systems with immediate communication, and we showed how it is possible to decrease the number of membranes. We presented a final solution, as we showed a frontier between the decidability and undecidability for these systems.

We considered a variant of membrane systems where rules are assigned to membranes. We considered two types of rules: splicing rules and cutting/recombination rules. We showed that one membrane is sufficient in order to generate all recursively enumerable languages.

The thesis was written in French and English. See additional details at `http://lita.sciences.univ-metz.fr/~verlan/`.