A CLIPS Simulator for Recognizer P Systems with Active Membranes

Mario J. PÉREZ JIMÉNEZ, Francisco José ROMERO CAMPERO

Dpto. de Ciencias de la Computación e Inteligencia Artificial
E.T.S. Ingeniería Informática. Universidad de Sevilla
Avda. Reina Mercedes s/n, 41012, Sevilla, España
E-mail: {Mario.Perez, Francisco-Jose.Romero}@cs.us.es

Abstract. In this paper we propose a new way to represent recognizer P systems with active membranes based on Production Systems techniques. This representation allows us to express the set of rules and the configurations in each step of the evolution as facts in a knowledge base. We provide a CLIPS program to simulate the evolutions of this variant of P systems.

1 Introduction

In [3] a new model of computation called P Systems, within the framework of Natural Computing (bio-inspired computing), was introduced by G. Paun.

It is based upon the notion of membrane structure that is used to enclose computing cells in order to make them independent computing units. Also, a membrane serves as a communication channel between a given cell and other cells "adjacent" to it. This model comes from the observation that the processes which take place in the complex structure of a living cell can be considered as computations.

Since these computing devices were introduced several variants have been considered. A fairly complete compendium about P systems can be found in [2].

The different variants of P systems found in the literature are generally thought as generating devices. Many of them have been proved to be computationally complete: they compute all Turing computable sets of natural numbers or all recursively enumerable languages, depending on the variant considered.

The model we study here, recognizer P systems with active membranes (a recognizing language device), works with symbol-objects, and it provides rules of division for membranes. In particular, P systems with active membranes are studied in [2] , section 7.2.

The main goal of this paper is to show a representation of P systems with active membranes based on Production Systems techniques.

Production Systems were first introduced by Post in 1943 in order to describe rewrite rules for symbol strings and are nowadays used as the basis for many rule-based systems. In this programming paradigm, rules are used to represent heuristics, or "rules of thumb", which specify a set of actions to be performed for a given situation. A rule is composed of a *left hand side portion* (if part) and a *right hand side portion* (then part). The LHS of a rule is a series of patterns which specify the facts (or data) which cause the rule to be applicable. The process of matching facts to patterns is called pattern matching. There exists a mechanism, called the *inference engine*, which automatically matches facts against patterns and determines which rules are applicable. The RHS of a rule is the set of actions to be executed when the rule is applicable. The actions of applicable rules are executed when the inference engine is instructed to begin execution. The inference engine selects using a *resolution strategy* a rule and then the actions of the selected rule are executed (which may affect the list of applicable rules by adding or removing facts). The inference engine then selects another rule and executes its actions. This process continues until no applicable rules remain.

The paper is organized as follows: Section 2 briefly presents some ideas about the analogies between P Systems and Production Systems. Section 3 recalls recognizer P systems with active membranes, input membrane and external output. Section 4 introduces CLIPS as a programming language used in Production Systems. Section 5 introduces the way to represent all basic ingredients of this variant of P systems. Section 6 studies the designed algorithm to simulate recognizer P systems and Section 7 presents a session with the simulator. Section 8 presents some conclusions.

References

[1] Cordón-Franco, A., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Sancho-Caparrini, F.: A Prolog simulator for deterministic P systems with active membranes, *New Generation Computing*, to appear

[2] Paun, G.: *Membrane Computing. An Introduction*, Springer-Verlag, 2002

[3] Paun, G.: Computing with membranes, *Journal of Computer and Systems Sciences*, 61(1), 2000, 108-143.

[4] Paun G.; Rozenberg, G.: A guide to membrane computing, *Theoretical Computer Sciences*, 287, 2002, 73-100.

[5] Paun, G.; Rozenberg, G.; Salomaa, A.: Membrane computing with external output, *Fundamenta Informaticae*, 41(3), 2000, 313,340.

[6] Pérez-Jiménez, M.J., Romero-Jiménez, A., Sancho-Caparrini F.: *Solving VALIDITY problem by active membranes with input.* M. Cavaliere , C. Martin-Vide and G. Paun (eds) Proceedings of the Brainstorming week on Membrane Computing, Report GRLMC 26/03, 279-290

[7] Pérez Jiménez, M.J., Romero-Jiménez, A. , Sancho-Caparrini, F.: *Teoría de la Complejidad en modelos de computacion celular con membranas.* Ed. Kronos, 2002

[8] Pérez Jiménez, M.J., Romero-Jiménez, A., Sancho-Caparrini, F.: A polynomial complexity class in P systems using membrane division, *Proceedings of the 5th Workshop on Descriptional Complexity of Formal Systems.* Budapest, Hungary, 2003, 284-294

[9] Pérez-Jiménez, M.J., Riscos-Núñez, A. A linear-time solution for the Knapsack problem using active membranes, C. Martin-Vide, G. Paun, G. Rozenberg, A. Salomaa (eds) *Workshop on Membrane Computing, WMC 2003* Lecture Notes in Computer Science, 2933, 2004, 250-268.

[10] CLIPS Web Page: http:// www.ghg.net/clips/CLIPS.html

[11] The P Systems Web Page: http://psytems.disco.unimib.it/