A Simulator for Confluent P Systems

M.A. Gutiérrez-Naranjo; M.J. Pérez-Jiménez; A. Riscos-Núñez

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: {magutier,marper,ariscosn}@us.es

Abstract. Software simulators for P system are nowadays the main tool to carry out experiments in the field of Membrane Computing. Although the simulation of a P system is a quite complex task, current simulators have been successfully used for pedagogical purposes and also as assistant tools for researchers. Up to now, simulators have always been designed to deal with a specific model of P systems. In this paper we present a new simulator which is not oriented to only one model of systems, but it allows the researcher to experiment with many existing models or even create new ones.

1 Introduction

Since Gh. Păun initiated Membrane Computing [21] as a new branch of Natural Computing, a large number of research lines have arisen in the field, both concerning the syntax and the semantics of the model. For example, we can consider P systems from a *generative* point of view, where starting from a fixed initial configuration, the system may generate nondeterministically a set of different outputs (that can be strings over some alphabet or also natural numbers); we can choose an *accepting* approach, where the system accepts or not at some input that is introduced at the beginning of the computation (this input can be a number, a word from a certain language, or an instance of a decision problem); we can also interpret the evolution of the P system as a *computing* process where some function is simulated (given an input n, the output of the system will be $f(n)$); or we can think of any other interpretation of the behavior of the P systems adequate to our purposes.

Irrespectively of the selected approach, it is usually a complex task to predict or to guess how a P system will behave. Moreover, as there do not exist, up to now, implementations in laboratories (neither *in vitro* nor *in vivo* nor in any electronical media), it seems natural to look for software tools that can be used as assistants that are able to simulate computations of P systems.

In the literature, several simulators can be found, but all of them have been designed to make experiments within a specific model of P systems, that is, both the set of possible types of rules and the way such rules are applied are

fixed. Thus, it becomes a difficult task to compare several solutions to a problem designed in different models, and these type of software cannot be used to investigate new possibilities beyond the usual models.

In this paper we propose a new tool which is born with the hope of becoming a tool for the creativity in Membrane Computing. It deals with several of the current models for P systems, also allowing to mix them in order to explore new possibilities. Indeed, its main feature is its modularity and the ability of embedding new P system models in future releases with a minimal modification of the code.

For example, in the current version, we have not included rules of endocytosis and exocytosis (mainly because, to the best of our knowledge, these have not been used yet in the solution of problems). Nevertheless, these rules can be included in future versions and coexist with the current ones. We will discuss more about the modularity of our simulator in Section 4.

The paper is organized as follows. In the next section, an overview of the current simulators is given. Section 3 presents the *simulator for confluent P systems (SCPS)*. Section 4 is devoted to discuss the improvements of this tool with respect to other previous simulators, and the paper ends with some final comments and conclusions.

References

1. Arroyo, F., Baranda, A.V., Castellanos, J., Luengo, C. and de Mingo, L.F. A Recursive Algorithm for Describing Evolution in Transition P Systems. *Pre-Proceedings of Workshop on Membrane Computing*, (C. Martín-Vide, Gh. Păun, eds.) Curtea de Arges, Romania, August 2001. Technical Report GRLMC 17/01, Rovira i Virgili University, Tarragona, Spain (2001) 19-30.
2. Arroyo, F., Baranda, A.V., Castellanos, J., Luengo, C. and de Mingo, L.F. Structures and Bio-Language to Simulate Transition P Systems on Digital Computers. In *Multiset Processing. Mathematical, Computer Science and Molecular Computing Points of View* (C.S. Calude, Gh. Păun, G. Rozenberg, A. Salomaa, eds.) Lecture Notes in Computer Science **2235**, (2001) 1-16.
3. Arroyo, F., Luengo, C., Baranda, A.V. and de Mingo, L.F. A Software Simulation of Transition P systems in Haskell, In *Membrane Computing WMC-CdeA 2002.* (Gh. Păun, G. Rozenberg, A. Salomaa, C. Zandron eds.) Lecture Notes in Computer Science, **2597**, (2003), 19–32.
4. Baranda, A.V., Castellanos, J., Arroyo, F. and Gonzalo, R. Data structures for implementing P systems in silico. In *Pre-proceedings of Workshop on Multiset Processing* (C.S. Calude, M.J. Dinneen, Gh. Păun, eds.) Curtea de Arges, Romania, CDMTCS TR 140, Univ. of Auckland, 2000, 21–34.
5. Baranda, A.V., Castellanos, J., Gonzalo, R., Arroyo, F. and de Mingo, F. Data Structures for Implementing Transition P systems in Silico. *Romanian Journal of Information Science and Technology* **4**, 1-2, (2001), 21-32.
6. Baranda, A.V., Castellanos, J., Arroyo, F. and Gonzalo, R. Towards an Electronic Implementation of Membrane Computing: A Formal Description of Nondeterministic Evolution in Transition P Systems. In *DNA Computing DNA 7* (N. Jonoska, N.C Seeman, eds.) Lecture Notes in Computer Science, **2340**, (2002), 350–359.

7. Balbontín-Noval, D., Pérez-Jiménez, M.J. and Sancho-Caparrini, F. A MzScheme Implementation of Transition P Systems. In *Membrane Computing 2002* (Gh. Păun, G. Rozenberg, A. Salomaa, C. Zandron, eds.) Lecture Notes in Computer Science, **2597**, (2003), 58–73.
8. Bratko, I.: *PROLOG Programming for Artificial Intelligence*, Third edition. Addison-Wesley, 2001.
9. Ciobanu, G. and Paraschiv D. P System Software Simulator. *Fundamenta Informaticae*, **49**, 1-3 (2002), 61–66.
10. Ciobanu, G. and Wenyuan, G. P Systems Running on a Cluster of Computers. In *Membrane Computing WMC 2003.* (C. Martín-Vide, Gh. Păun, G. Rozenberg, A. Salomaa, eds.) Lecture Notes in Computer Science, **2933**, (2004), 123–139.
11. Cordón-Franco, A. Gutiérrez Naranjo M.A., Pérez Jiménez M.J., Riscos Núñez A. and Sancho-Caparrini, F. Implementing in Prolog an effective cellular solution to the knapsack problem. *Membrane Computing WMC 2003* Lecture Notes in Computer Science **2933**, (2004), 140 - 152.
12. Cordón-Franco, A., Gutiérrez-Naranjo, M.A., Pérez-Jiménez M.J., and Riscos-Núñez A. and Sancho-Caparrini, F. Cellular Solutions of Some Numerical NP-Complete Problems: A Prolog Implementation. *Molecular Computational Models: Unconventional Approaches.* (M. Gheorge, ed.), Idea Group, Inc. (2005).
13. Cordón-Franco, A., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J. and Sancho-Caparrini, F. A Prolog Simulator for Deterministic P Systems with Active Membranes, *New Generation Computing*, **22**, 4 (2004) 349–364.
14. Cordón-Franco, A. Gutiérrez Naranjo M.A., Pérez Jiménez M.J. and Riscos Núñez A. Exploring Computation Trees Associated with P Systems *Membrane Computing WMC 2004* Lecture Notes in Computer Science **3365**, (2005), 278 – 286.
15. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J. and Riscos-Núñez, A. A Fast P System for Finding Balanced 2-partition. *Soft Computing* Springer-Verlag Heidelberg. **9**, 5, (2005).
16. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J. and Riscos-Núñez, A. Available Membrane Computing Software. Volume of Applications of P Systems. Lecture Notes in Computes Science. In press.
17. Izbasa, C.; Bonchis, C.; Garboni, C. and Ciobanu, G. A web-enabled P system simulator with query facilities. In this volume.
18. Luisi, P.L.: The Chemical Implementation of Autopoiesis, *Self-Production of Supramolecular Structures* (G.R. Fleishaker et al., eds.), Kluwer, Dordrecht, 1994
19. Maliţa, M. Membrane Computing in Prolog, *Pre-proceedings of the Workshop on Multiset Processing* (C.S. Calude, M.J. Dinneen, Gh. Păun, eds.) Curtea de Arges, Romania, CDMTCS TR 140, Univ. of Auckland, 2000, 159–175.
20. Nepomuceno-Chamorro, I.A. A Java simulator for basic transition P systems. In *Proceedings of the Second Brainstorming Week on Membrane Computing* (Gh. Păun, A. Riscos, A. Romero, F. Sancho, eds.) Report RGNC 01/04, 2004, 309–315.
21. Păun, Gh. Computing with membranes. Turku Centre for Computer Science, TUCS Technical Report,N.208, (1998)
22. Păun, Gh. Computing with membranes. *Journal of Computer and System Sciences*, **61**, 1 (2000), 108–143.
23. Păun, Gh. P systems with active membranes: Attacking NP-complete problems. *Journal of Automata, Languages and Combinatorics*, **6**, 1 (2001), 75–90.
24. Pérez-Jiménez, M.J. and Romero-Campero, F. A CLIPS simulator for recognizer P systems with active membranes. In *Proceedings of the Second Brainstorming Week on Membrane Computing* (Gh. Păun, A. Riscos, A. Romero, F. Sancho, eds.) Report RGNC 01/04, 2004, 387–413.

25. Pérez-Jiménez, M.J.; Romero-Jiménez, A.; Sancho-Caparrini, F. Solving VALIDITY Problem by Active Membranes with Input. In *Proceedings of the Brainstorming Week on Membrane Computing* (M. Cavaliere, C. Martín-Vide, Gh. Păun, eds.), Report GRLMC 26/03, 2003, 279–290.
26. Riscos-Núñez, A. Cellular programming: efficient resolution of numerical NP-complete problems. Ph.D. Thesis. University of Seville, 2004.
27. Suzuki, Y. and Tanaka, H. On a LISP implementation of a class of P systems. *Romanian Journal of Information Science and Technology*, **3**, 2 (2000), 173–186.
28. Syropoulos A., Mamatas E.G., Allilomes P.C. and Sotiriades, K.T. A Distributed Simulation of Transition P Systems. In C. Martín-Vide, Gh. Păun, G. Rozenberg, A. Salomaa (eds.) *Workshop on Membrane Computing 2003.* Lecture Notes in Computer Science, **2933**, (2004), 357–368.
29. `http://www.lpsi.eui.upm.es/nncg/`
30. Logic Programming: `http://www.afm.sbu.ac.uk/logic-prog/`