Available Membrane Computing Software

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. The simulation of a P system with current computers is a quite complex task. P systems are intrinsically non deterministic computational devices and therefore their computation trees are difficult to store and handle with one-processor (or bounded number of processors) computers. Nevertheless, there exists a first generation of simulators which can be successfully used for pedagogical purposes and also as assistant tools for researchers. This paper summarizes some of these simulators, presenting the state-of-the-art of the available software for simulating (different variants of) cell-like membrane systems.

1 Introduction

Since Gh. Păun initiated Membrane Computing [31] as a new branch of Natural Computing, a large number of variants have been considered, both concerning the syntax and the semantics of the model.

In many of these variants, cellular systems are seen as devices of a generative nature, that is, from a given initial configuration several distinct computations may be developed (in a non deterministic manner) and they may produce different outputs. We can consider in this case that the system *generates* the set of all the *outputs* of all the computations obtained from the initial configuration (and this set can be interpreted as the *language generated* by the system).

There are other approaches where P systems perform *computing tasks.* For example, if a certain number, n, is encoded somehow in the initial configuration and we consider the cardinality of the output multiset as the result of the successful computations, then we can interpret that the system *computes* a partial function from natural numbers onto sets of natural numbers.

Finally, membrane systems can also be used to deal with decision problems. In this cases, special objects *yes* and *no* are included in the working alphabet and thus the system is able to produce a boolean output (accepting or rejecting the input), in a *confluent* manner.

In all these approaches, we get the *output* of the computation from a final configuration, looking at the contents of the output membrane or considering the objects that have been sent out of the system, in the case of the *external output* variant.

Unfortunately, since it is a machine-oriented model of computation, it is usually a complex task to predict or to guess how a P system will behave when we are designing a cellular solution to a problem. Moreover, as there do not exist, up to now, implementations in laboratories (neither *in vitro* nor *in vivo* nor in any electronical medium), it seems natural to look for software tools that can be used as assistants that are able to simulate computations of P systems.

This is the initial motivation for programming simulators. It is clear that such software tools are very useful when trying to understand how a cellular system works (both for pedagogical purposes and as an assistant tool for researchers). Another important point is that the formal verifications of the cellular solutions designed in this framework are specially hard, and having a simulator at hand allows us to get quickly and easily information about the evolution of P systems that can be used as starting point for a formal verification, maybe suggesting invariants that can be useful for the proofs.

The paper is organized as follows. In the next section, some general considerations about the processes of the design and development of simulators of P systems are given. Section 3 is devoted to the simulators that work with transition P systems and run on sequential machines. Section 4 deals with parallel and distributed simulators (also simulating transition P systems) and section 5 presents simulators for P systems with active membranes, including a session of one of them. The paper ends with a section devoted to other software and some conclusions.

References

1. Alhazov, A. Maximally Parallel Multiset-Rewriting Systems: Browsing the Configurations. Unpublished manuscript.
2. Alhazov, A., Margenstern, M., Rogozhin, V., Rogozhin, Yu. and Verlan, S. Communicative P systems with Minimal Cooperation. *Lecture Notes in Computer Science.* Submitted.
3. Ardelean, I.I. and Cavaliere, M. Modelling biological processes by using a probabilistic P system software. *Natural Computing*, **2**, 2(2003), 173-197.
4. 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.
5. 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.
6. 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.
7. Attiya, H. and Welch, J. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*, McGraw-Hill, 2000.

8. 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.
9. 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.
10. 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.
11. 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.
12. Bianco, L. A P System Simulator: Introduction to Psim. *Unpublished manuscript.*
13. Bianco, L., Fontana, F., Franco, G. and Manca, V. P Systems in Bio Systems. In this volume.
14. Binder, A., Freund, R., Lojka, G. and Oswald, M. Implementation of catalytic P systems, *Proceedings of CIAA 2004, Ninth International Conference on Implementation and Application of Automata*, Kingston, Canada (2004), 24–33.
15. Cavaliere, M. Evolution-Communication P Systems. In *Membrane Computing 2002* (Gh. Păun, G. Rozenberg, A. Salomaa, C. Zandron, eds.) Lecture Notes in Computer Science, **2597**, (2003), 134-145.
16. Cavaliere, M. and Ardelean, I.I. Modelling Respiration in Bacteria and Respiration/Photosynthesis Interaction in Cyanobacteria by Using a P System Simulator. In this volume.
17. Ciobanu, G. and Paraschiv D. P System Software Simulator. *Fundamenta Informaticae*, **49**, 1-3 (2002), 61–66.
18. Ciobanu, G.; Păun, Gh.; Ştefănescu, Gh. Sevilla Carpets Associated with P Systems, in M. Cavaliere, C. Martín–Vide and Gh. Păun (eds.), *Proceedings of the Brainstorming Week on Membrane Computing*, Tarragona, Spain, 2003, Report RGML 26/03, 135-140.
19. Ciobanu, G. and Wenyuan G., A parallel implementation of transition P systems. In *Preproceedings of the Workshop on Membrane Computing* (A. Alhazov, C. Martín-Vide, Gh. Păun, eds.) Tarragona, July 17-22 2003, 169-184.
20. 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.
21. 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.
22. 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. (In press).
23. 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.

24. 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. In press
25. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J. and Riscos-Núñez, A. On Descriptive Complexity of P Systems. In *Pre-proceedings of the Workshop on Membrane Computing WMC5* (G. Mauri, Gh. Păun and C. Zandron, eds.). Universitá di Milano-Bicocca, Italy, 2004, 245–255.
26. 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.
27. Manca, V. On the Dynamics of P Systems. In *Pre-proceedings of the Workshop on Membrane Computing WMC5* (G. Mauri, Gh. Păun and C. Zandron, eds.). Universitá di Milano-Bicocca, Italy, 2004, 29–43.
28. Margenstern, M., Rogozhin, V., Rogozhin, Y. and Verlan, S. About P systems with minimal symport/antiport rules and four membranes. In *Pre-proceedings of the Workshop on Membrane Computing WMC5* (G. Mauri, Gh. Păun and C. Zandron, eds.). Universitá di Milano-Bicocca, Italy, 2004, 283–294.
29. 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.
30. Nicolau Jr., D.V., Solana G., Fulga F. and Nicolau D.V. A C Library for Simulating P Systems. *Fundamenta Informaticae*, **49**, 1-3 (2002), 241–248.
31. Păun, Gh. Computing with membranes. Turku Centre for Computer Science, TUCS Technical Report,N.208, (1998)
32. Păun, Gh. Computing with membranes. *Journal of Computer and System Sciences*, **61**, 1 (2000), 108–143.
33. Păun, Gh. P systems with active membranes: Attacking NP-complete problems. *Journal of Automata, Languages and Combinatorics*, **6**, 1 (2001), 75–90.
34. Păun, G.: *Membrane Computing. An introduction*, Springer-Verlag, Berlin, 2002.
35. 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.
36. 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.
37. Pérez-Jiménez, M.J. and Sancho-Caparrini, F. A formalization of transition P systems. *Fundamenta Informaticae*, **49**, 1-3 (2002), 261–272.
38. Petreska, B. and Teuscher, C. A Reconfigurable Hardware Memebrane System. In *Membrane Computing WMC 2003*. (C. Martín-Vide, Gh. Păun, G. Rozenberg, A. Salomaa, eds.) Lecture Notes in Computer Science, **2933**, (2004), 269–285.
39. Riscos-Núñez, A. Cellular programming: efficient resolution of numerical NP-complete problems. Ph.D. Thesis. University of Seville, 2004.
40. 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.
41. Suzuki, Y., Fujiwara, Y., Tanaka, H. and Takabayashi, J. Artificial Life Applications of a Class of P Systems: Abstract Rewriting Systems on Multisets. In C.S. Calude, Gh. Paun, G. Rozenberg, A. Salomaa (eds.) *Multiset Processing. Mathematical, Computer Science, and Molecular Computing Points of View.* Lecture Notes in Computer Science, **2235**, (2001), 299-346.

42. 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.
43. `http://research.araneous.com`
44. `http://www.lpsi.eui.upm.es/nncg/`
45. `http://psystems.disco.unimib.it/`
46. `http://www.di.univr.it`
47. `http://www.gcn.us.es`
48. `http://www.teuscher.ch/psystems`