The Fifteenth International Symposium on Artificial Life and Robotics 2010 (AROB 15th '10),
B-Con Plaza, Beppu,Oita, Japan, February 4-6, 2010

Organization Levels in P Systems

Marion OSWALD
Computer and Automation Research Institute
Hungarian Academy of Sciences
Budapest, Hungary
marion@sztaki.hu

Abstract

We discuss different (structural) organization forms of a formal model of computation that is abstracted from the structure and functioning of living cells: P systems, introduced in 1998 by Gh. Păun. After a brief review of some of the main classes of P systems, having an underlying tree or graph structure, we present some indications that suggest to use a more general hierarchical structure. We highlight some examples and point out further applications where this broader approach could be useful.

1 Introduction

Membrane Computing has become a vivid and active research front in the past eleven years. (For an introduction to the field see [13], a complete bibliography can be found at [19]).

P systems (also called membrane systems) have originally been defined by Gh. Păun [12] as a computing model in the following sense. In a hierarchical membrane structure (that can also be represented as a tree), objects evolve according to so-called evolution rules. Starting from an initial configuration of the system, a computation is performed by applying the rules in a nondeterministic, maximally parallel way until the system eventually halts.

Dating back only a decade, many interesting aspects have already been investigated, e.g., the computational power of different variants, normal forms, the solution of NP-complete problems in polynomial or even linear time, etc., see for instance [14].

Although, as stated in [12], the original intention of P systems was not to model the cell, but to abstract from its structure and function to obtain a computing device, we can observe a recent trend in the area towards modeling biological phenomena in this versatile framework.

While the underlying structure of the original P system is a tree, where membranes can be nested, and hence be represented by a Venn diagram (but without intersecting sets), a more general structure has been taken into account a short time later: tissue P systems, having an underlying (arbitrary) graph structure. Only recently, an even more general (i.e., hypergraph) structure was taken into account.

Without going into formal details, we here give an outline of the structural concepts employed in membrane computing so far and point out possible extensions.

After some preliminaries about graph concepts and P systems in general, we briefly review recent variants that take into account a more general organization form in section 3. Some possible extension to a more general model are discussed in section 4, some notes on possible applications conclude the paper.

2 Preliminaries

2.1 (Hyper)Graphs

A *graph* G is a pair (V, E), where V is a non-empty set of nodes and E is a set of edges $e = (v_1, v_2)$, $v_1, v_2 \in V$. A *path* in G is a sequence of vertices of the form $v_0, v_1, ..., v_n$ such that $(v_i, v_{i+1}) \in E$ for $i = 0, ..., n-1$. In a closed path (also called *cycle*), $v_0 = v_n$. We call a graph G *connected* if there is a path between any two distinct vertices. A graph is a *tree* if and only if for every pair of distinct vertices v_1, v_2 there is exactly one path from v_1 to v_2. In other words, a tree is a connected graph without cycles.

While in graphs, edges are between two nodes, in a more general graph, a so-called *hypergraph*, edges are between sets of nodes (that may also intersect). The edges in a hypergraph are also called *hyperedges*.

Following [3], a hypergraph is defined as follows.

Let $V = \{v_1, v_2, ..., v_n\}$ be a finite set. A hypergraph on V is a family $H = \{E_1, E_2, ..., E_n\}$ of subsets of such that

- $E_i \neq \emptyset$, $1 \leq i \leq m$ and
- $\cup_{i=1}^{m} E_i = X$.

The notion of hypergraph can even be more generalized by having hyperedges pointing to hyperedges, thereby inducing a hierarchy, that then corresponds to a so called *directed acyclic graph* (or *dag*, for short). An example is shown in figure 1.

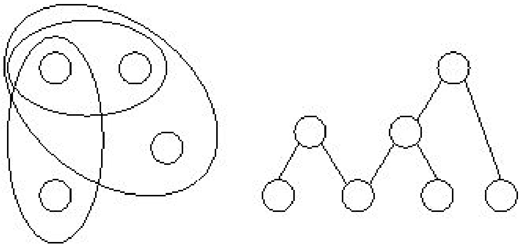

Figure 1: hypergraph and associated dag

For more details we refer to any textbook in the area, e.g. to [5].

2.2 P Systems

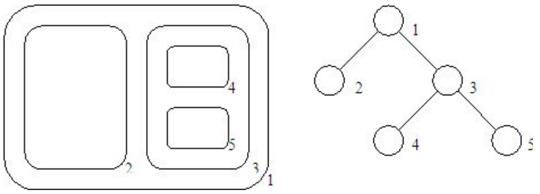

Figure 2: membrane structure and associated tree

P systems are computing devices abstracted from cell functioning and are based on the notion of a *membrane structure*. A membrane structure consists of membranes hierarchically embedded in the outermost skin membrane; every membrane encloses a region possibly containing other membranes; the region outside the skin membrane is called outer region or environment. A membrane structure can be graphically represented by a Venn diagram without intersecting sets, so that any two sets are either disjoint or one set is a subset of the other one. Since the membranes in such a structure (usually labelled in a one-to-one manner) are hierarchically arranged, it can also be represented by a tree (an example for a membrane structure and its associated tree is shown in figure 2). On the other hand, it can also be described by a string of correctly matching parentheses (e.g., the membrane structure from figure 2 corresponds to the string $[_1[_2]_2[_3[_4]_4[_5]_5]_3]_1$).

In the membranes, *multisets of objects* can be placed, which evolve according to given *evolution rules*. Depending on the model, these rules can be applied in parallel across all membranes or in a rather sequential manner. Usually, the rules to be applied are non-deterministically chosen (i.e., if an object can evolve according to more than one evolution rule at the same time, any one is chosen). A configuration can be illustrated by putting the objects and rules in the corresponding compartments of the membrane structure. In this way, a computing device is obtained in the following sense: Starting from an initial configuration (given by the membrane structure, the distribution of the objects, as well as evolution rules), the system evolves by passing from one configuration to another one, thereby performing a computation, see [13] for motivations and examples.

Many variants of P systems have been investigated since their introduction, by varying various ingredients (the nature of the objects, evolution rules, derivation modes, etc.) We refer the interested reader e.g. to [14] for more details.

3 Higher Organization Levels in P Systems

P systems with an underlying tree structure (as originally defined) offer the possibility of hierarchical organization: one of the main feature of membrane systems and their biological motivation is that it can also bee seen as having more layers of abstraction, e.g., atoms-molecules-cells-tissues-organs-organisms, etc.

However, the exhibited hierarchical organization is limited because on one hand, a tree is rather restrictive (in that it can only have one root node), and on the other hand, the nodes cannot overlap (i.e., the membranes are nested but do not intersect). Yet just this overlap could be extremely interesting. On one hand, it is naturally plausible (e.g., several signaling and/or metabolic networks may overlap, like for example the Wnt signaling pathway that plays a role in embryogenesis and cancer, see [7]) and it might well be used in an algorithmic way as well (see section 5).

Taking a more general (arbitrary) graph structure, the above mentioned restriction is partly overcome, but at the price of loosing the hierarchical information.

Yet in the last few years, more general forms of (structural) organization came to existence also in membrane computing, we give a few examples in chronological order:

- **Quorum sensing P systems** were introduced in 2005, and can in fact be considered the first ones that took this the mentioned overlap into account, although not explicitly. In [2], the authors model colonies of bacteria by means of Quorum sensing P systems, where each bacterium acts in its specific environment, and the different environments can overlap.
- **Multigraphical Membrane Systems**, introduced by Obtułowicz in 2007 [11] present a visual formalism that is based on multigraphs (i.e., graphs that can have multiple edges, but the edges cannot overlap).
- **Modularisation** is taken into account by Romero-Campero et al. [15], [16]. By means of Stochastic P systems, the authors investigate a very particular variant for a specific model, taking into account 3 levels: the molecular, the cell, and the colony level.
- **Hyperdag P Systems** were only recently introduced by Nicolescu et al. [10] as P systems with an underlying generalized multiset-based hypergraph (i.e., dag) structure, thereby also introducing rules for enhanced inter-level communication ("go-sibling", see [10]).

While in [2], [15], and [16], the focus is on biological modelling, the investigations in [10] are of a more general kind, even proposing a new planar representation of hyperdag P systems. While this definitely has its merits in some areas, it also has limitations when it e.g. comes to biological modelling, where the visualization is of great importance. Hypergraphs might not be the best choice due to its difficulty of being drawn as a pictorial representation. Yet a closely related concept called metagraphs (introduced by Basu and Blanning [1]) seems to be a better choice in this respect. Metagraphs support nodes, edges and even subnetworks contained and nested within a scalable structure, where the nodes can also overlap. Easy transitions between different levels of resolution are supported, called "semantic zooming" (see [8]), a concept that is already adopted by some software tools (e.g., VisANT, see [20]).

4 Towards a more general structural approach

As already stated in the previous sections, it might well be useful to think of P systems having an underlying generalized hypergraph or directed acyclic graph structure. From a theoretical point of view, it is obvious that the system under consideration does not loose its computational power if only the structure is changed in this way (since a hypergraph is more general than a graph, i.e., each graph is a hypergraph, in which the cardinality of each hyperedge is exactly two.)

Yet from the modeling perspective, (and not only concerning computer networks, as explored in [10]), it offers a new opportunity to include more complex system behavior (as e.g. argued by Mesarovic and Sreenath [9] in terms of "complex "systems biology). In this respect, in addition to the more general organization levels, many possible extensions could facilitate the accuracy of the model.

Like in the category theory framework of memory evolutive systems as explored in [6] (also see [18]), one could incorporate different timescales for the levels (as, e.g., reactions at an atomic level happen faster than those on the molecular level.)

On the other hand, and especially when the system becomes more complex, we could include some regulation mechanism.

For example, we could introduce regulators that have a certain "radius" (i.e., be responsible for some elements, e.g., for a certain number of nodes in one level). Now if a regulator observes a specific pattern, then it can induce an action in the corresponding cells, like e.g. changing the ruleset of a cell.

If we look at the evolution of the system, it is also natural to not only let the symbols evolve in the course of time, but also the underlying structure. Some parts of the system might grow, while others could (structurally) reorganize themselves or even die.

This idea is not new in the area of membrane computing (so-called P systems with active membranes have already been introduced in 2001, originally mainly used to solve NP-complete problems in polynomial, sometimes even linear time), but in this more general framework of course needs some adaptations to capture also the possible overlap of the cells, i.e., the nodes in the hypergraph (e.g., we could consider rules like $[_i a]_i [_j b]_j \to [_i c [_j d]_i e]_j$ to account for this). The structural change could be induced by rules (e.g., in the sense of [4]), or be incited by the regulators.

While all the above mentioned extensions can be defined (i.e., incorporated in a formal definition) in a straightforward way, a big challenge remains in defining concepts that allow for emergence, a central concept of complex systems.

5 Conclusion

We discussed different structural organization forms of P systems, highlighting the usefulness of a more general underlying structure, i.e., (generalized) hypergraphs that might enrich especially the modelling of hierarchical systems (biological, social, ecological, cultural systems, etc.).

On the other hand, it seems also to be worthwhile to explore the proposed systems for their algorithmic capabilities. Many computations comprise a common set of known interaction and computation patterns. One particular example are overlapping subproblems (i.e., subproblems that are reused several times). Problems that have this property are often solved by dynamic programming techniques more efficiently than with other methods. Yet very often complex data dependencies occur and complicate parallelization. Investigations concerning the practicability of P systems (with an underlying hypergraph structure) in this respect are currently performed by the author (focusing on so-called nonserial polyadic dynamic programming, like e.g., matrix chain multiplication and RNA secondary structure prediction).

Acknowledgements We gratefully acknowledge interesting and fruitful discussions with E. Csuhaj-Varjú, T. Roska and Gy. Vaszil.

References

[1] A. Basu, R.W. Blanning, A. Shtub: Metagraphs in Hierarchical Modeling. Management Science 43 (5) 1997, 623–639.

[2] F. Bernardini, M. Gheorghe, N. Krasnogor: Quorum sensing P systems. Theoretical Computer Science 371 (2007) 20—33.

[3] C. Berge: *Hypergraphs.* North-Holland Mathematical Library, Amsterdam, 1998.

[4] E. Csuhaj-Varjú, A. Di Nola, Gh. Păun, M. J. Pérez-Jiménez, Gy. Vaszil: Editing Configurations of P Systems. Fundamenta Informaticae 82 (1-2), 2008, 29–46.

[5] R. Diestel: *Graph Theory.* 3rd edition, Graduate Texts in Mathematics 173, Springer, Heidelberg, 2005.

[6] A.C. Ehresmann, J.-P. Vanbremeersch: *Memory Evolutive Systems; Hierarchy, Emergence, Cognition* (Studies in Multidisciplinarity). Elsevier Science, 2007.

[7] H. A. Kestler, M. Kühl: From individual Wnt pathways towards a Wnt signalling network. Phil. Trans. R. Soc. B 363 (1495) 2008, 1333–1347.

[8] Z. Hu, J. Mellor, J. Wu, M. Kanehisa, J. M. Stuart, C. DeLisi: Towards zoomable multidimensional maps of the cell. Nature Biotechnology 25 (5) 2007, 547–554.

[9] M. Mesarovic, S.N. Sreenath: Beyond the flat earth perspective in systems biology. Biological Theory 1 (1) 2006, 33-34.

[10] R. Nicolescu, M.J. Dinneen, Y.-B. Kim: Structured Modelling with Hyperdag P Systems: Part A. in: M. A. Martínez-del-Amor, E. F. Orejuela-Pinedo, Gh. Păun, I. Pérez-Hurtado, A. Riscos-Núñez (Eds.): Seventh Brainstorming Week on Membrane Computing, Sevilla, February 2-6, 2009, Volume II, 85 – 108.

[11] A. Obtułowicz: Multigraphical Membrane Systems: a Visual Formalism for Modeling Complex Systems in Biology and Evolving Neural Networks. In: G. Eleftherakis, P. Kefalas, Gh. Păun (Eds.): Proceedings of the 8th InternationalWorkshop on Membrane Computing (WMC8), SEERC 2007, 509–512.

[12] Gh. Păun: Computing with Membranes. Journal of Computer and System Sciences 61, 1 (2000), 108–143.

[13] Gh. Păun: Computing with Membranes: An Introduction. Springer, Berlin, 2002.

[14] Gh. Păun, G. Rozenberg, A. Salomaa (eds.): *The Oxford Handbook on Membrane Computing.* Oxford University Press, to appear.

[15] F. J. Romero-Campero, J. Twycross, M. Bennett, M. Càmara, and N. Krasnogor: Modular Assembly of Cell Systems Biology Models Using P Systems. Int. J. Found. Comput. Sci. 20(3) 2009, 427–442.

[16] F. J. Romero-Campero, J. Twycross, H. Cao, J. Blakes, and N. Krasnogor: A Multiscale Modelling Framework Based On P Systems. In: D.W. Corne, P. Frisco, Gh. Păun, G. Rozenberg, A. Salomaa (Eds.): Membrane Computing. 9th InternationalWorkshop, WMC 2008 Edinburgh, UK, July 28-31, 2008, 63-77.

[17] E. Van Hensbergen, C. Forsyth, J. McKie, R. Minnich: Holistic aggregate resource environment. ACM SIGOPS Operating Systems Review 42 (1) 2008, 85–91.

[18] The MES Web Page,
`http://pagesperso-orange.fr/vbm-ehr/`

[19] The P Systems Web Page,
`http://ppage.psystems.eu`

[20] The VisANT Web Page,
`http://visant.bu.edu/`