New applications for an old tool

Luis Valencia-Cabrera, David Orellana-Martín,
Ignacio Pérez-Hurtado, Mario J. Pérez-Jiménez

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

Summary. First, the dependency graph technique, not so far from its current application, was developed trying to find the shortest computations for membrane systems solving instances of `SAT`. Certain families of membrane systems have been demonstrated to be non-efficient by means of the reduction of finding an accepting computation (respectively, rejecting computation) to the problem of reaching from a node of the dependency graph to another one.

In this paper, a novel application to this technique is explained. Supposing that a problem can be solved by means of a kind of membrane systems leads to a contradiction by means of using the dependency graph as a reasoning method. In this case, it is demonstrated that a single system without dissolution, polarizations and cooperation cannot distinguish a single object from more than one object.

An extended version of this work will be presented in the 20th International Conference on Membrane Computing.

1 Introduction

The *computational efficiency* of a model in a computing paradigm refers to its ability to provide polynomial time solutions for computationally hard problems, generally achieved by making use of an exponential workspace constructed in a natural way. Aspects related to the computational efficiency within membrane computing were first analyzed in 1999, with the introduction of a new computing model called *P system with active membranes* [5]. These systems are non-cooperative (the left hand side of any rule consists of only one object) and their membranes play a relevant role in computations to the extent that new membranes can be created by division rules. The membranes of these systems are supposed to have one of three possible electrical polarizations: positive, negative or neutral. In this context, it was given an *ad-hoc* solution to the Boolean satisfiability problem (`SAT`)

by means of such kind of P systems. More specifically, a P system with active membranes which makes use of *simple* object evolution rules (only one object is produced for this kind of rules), dissolution rules and division rules for elementary and non-elementary membranes, is associated with every instance φ of SAT. Thus, the syntactic structure of the formula is "captured" by the description of the system and, furthermore, in this context a P system can only process one instance of the problem. The solution provided runs in linear time with respect to the *size* of the input formula φ, that is, the maximum between the number of variables and the number of clauses in φ.

Usually, computational complexity theory deals with decision problems, that is, problems requiring a yes/no answer. Each decision problem has a language associated with it, in a natural way, so that solving such problems is defined through the recognition of the corresponding language. Thus, in order to describe in a formal way what solving a decision problem means, basic *recognizer transition P systems* (initially called *decision P systems*) were defined [7].

Let us recall that an abstract problem can be solved by using a single Turing machine, that is, for every instance of the problem, the Turing machine receiving the input corresponding to that instance returns the correct answer. This is due to the fact that these machines have an unlimited and unrestricted memory, given the infinite tape it includes (consisting of an infinite number of cells). Bearing in mind that the ingredients necessary to define a membrane system are finite, an abstract problem should be solved, in general, by an infinite numerable family of membrane systems, in such a way that each system in the family is in charge of processing all the instances having the same size.

It seems interesting to analyze what kind of membrane systems are capable of solving decision problems through only one unique system. In this context, it is essential to clarify how the instances of the problem are introduced into the system. Next, we consider the case in which the instances are directly introduced inside the system (*free* of resources) by means of a representation of the problem to be solved. It is important to remark that this means that the input alphabet of the P system is the same one that the alphabet of the problem, so there is no possibility of encoding, for instance, an instance of a problem from **P** to an object yes or an object no.

Definition 1. *Let $X = (I_X, \theta_X)$ be a decision problem where I_X is a language over a finite alphabet Σ_X. Let $\mathcal{R}$ be a class of recognizer membrane systems <u>with input membrane</u>. We say that problem X is solvable in polynomial time by a single membrane system Π from $\mathcal{R}$, free of resources, denoted by $X \in \mathbf{PMC}_{\mathcal{R}}^{1f}$, if the following holds:*

- *The input alphabet of Π is Σ_X.*
- *The system Π is* polynomially bounded *with regard to X; that is, there exists a polynomial $p(r)$ such that for each instance $u \in I_X$, every computation of the system Π with input multiset u performs at most $p(|u|)$ steps.*

- *The system Π is* sound *with regard to X; that is, for each instance $u \in I_X$, if there exists an accepting computation of the system Π with input multiset u then $\theta_X(u) = 1$.*
- *The system Π is* complete *with regard to X; that is, for each instance $u \in I_X$ such that $\theta_X(u) = 1$, every computation of the system Π with input multiset u is an accepting computation.*

From the previous definition it is easy to prove that $\mathbf{PMC}_{\mathcal{R}}^{1f} \subseteq \mathbf{PMC}_{\mathcal{R}}$, for every class $\mathcal{R}$ of recognizer membrane systems <u>with</u> input membrane.

2 Previous uses of the dependency graph

The dynamics of a membrane system provides, in a natural way, a tree of computation. More precisely, the *computation tree* of a membrane system Π, denoted $Comp(\Pi)$, is a rooted labelled maximal tree, whose maximal branches will be called *computations* of Π. A computation of Π is a halting computation if and only if it is a finite branch. The labels of the leaves of $Comp(\Pi)$ are called *halting configurations*.

Given a semi-uniform or uniform solution (in polynomial time) for a decision problem by means of a family of recognizer membrane systems, every instance of the problem is processed by a system of the family. This system must be confluent, so in order to know its answer for any instance it is enough to consider only one computation of such system. In this context, an exciting challenge would be looking for a computation with minimum length. For that, some *weak metrics* on the degree of closeness configurations of a membrane system with a fixed structure of membranes have been studied in [2]. In this context, in order to search for the shortest paths in a graph providing a *sound* computation of the system, the *dependency graph* associated with the set of rules of a recognizer membrane system was introduced. This concept is based on the dependence among elements of the alphabet with respect to the set of rules of the P system. Several weak metrics over the set of configurations of the system based on the concept of dependency graph were considered, starting from the notion of *distance* between two nodes of the graph (the length of the shortest path connecting v_1 and v_2, or *infinite* if there is no path from v_1 to v_2).

Also, in some kind of recognizer membrane systems, it is possible to consider a directed graph (also called *dependency graph*) verifying the following properties: (a) it can be constructed from the set or rules of the system in polynomial time, that is, in a time bounded by a polynomial function depending on the total number of rules and the maximum length of them; and (b) the accepting computations of such systems can be characterized by means of a "reachability" property in the dependency graph associated with it (the existence of a path in the graph between two specific nodes). Therefore, dependency graphs provide a technique to tackle the limits on efficient computations in membrane systems; that is, the non-efficiency of such systems.

3 Dependency graph as a technique to prove negative results in membrane systems

Let $\mathcal{R}$ be a class of recognizer membrane systems such that every system from $\mathcal{R}$ is associated with a dependency graph verifying the following property: a computation of a system from $\mathcal{R}$ is an accepting computation if and only if there exists a path between two distinguished nodes in the dependency graph associated with the system. In this situation, it is possible to show that some decision problem $X = (I_X, \theta_X)$ *cannot be solved* in polynomial time in a uniform way by means of a single membrane system, free of resources, from $\mathcal{R}$. This remark is illustrated by an example.

The ONLY-ONE-OBJECT problem is the decision problem $X = (I_X, \theta_X)$ defined as follows: $I_X = \{a^n \mid n \in \mathbb{N}, n \geq 1\}$ and $\theta_X(a^n) = 1$ if and only if $n = 1$. It is easy to design a deterministic Turing machine which takes two computation steps, solving the ONLY-ONE-OBJECT problem. Let us see that ONLY-ONE-OBJECT $\notin \mathbf{PMC}^{1f}_{\mathcal{AM}^0(-d,+ne)}$.

Theorem 1. *There is no recognizer membrane system from the class $\mathcal{AM}^0(-d,+ne)$ solving the* ONLY-ONE-OBJECT *problem in polynomial time by a single membrane system and free of resources.*

Proof. Let us assume that there exists a recognizer membrane system Π from $\mathcal{AM}^0(-d,+ne)$ verifying the following: (a) the input alphabet of Π is the singleton $\{a\}$; (b) every computation of Π with input multiset $\{a\}$ is an accepting computation; and (c) every computation of Π with input multiset $\{a^n\}$, for each $n > 1$, is a rejecting computation.

Let us denote by $G_{\Pi+\{a\}}$ (respectively, $G_{\Pi+\{a^n\}}$, for each $n > 1$) the dependency graph associated with the system $\Pi + \{a\}$ (resp. $\Pi + \{a^n\}$). Then, for each $n > 1$, we have $G_{\Pi+\{a\}} = G_{\Pi+\{a^n\}}$, since there would always be an edge $(s_\Pi, (a, i_{in}))$ in the dependency graph, and the rest of the graph would remain the same. Besides, every computation of $\Pi + \{a\}$ is an accepting computation if and only if every computation of $\Pi + \{a^n\}$, for each $n > 1$, is an accepting computation, which is a contradiction of the initial hypothesis, thus there cannot exist such membrane system.

4 Conclusions

Along this work, some of the main results concerning the use of dependency graphs within membrane computing to analyze the computational efficiency of computing models have been reviewed. It is worth pointing out that, albeit the $\mathbf{P}$ *versus* $\mathbf{NP}$ problem is the most important one in Computer Science, there are other interesting problems in the field of Computational Complexity Theory, also below $\mathbf{P}$. When using polynomial precomputed resources, problems from $\mathbf{P}$ can be easily solved. But considering membrane systems *free of precomputed resources*, things change

in such a way that there cannot be trivial solutions that could be obtained at first. Then, it would be useful to study these kinds of systems to solve problems below this complexity class. In this case, with this technique it has been demonstrated that there is no solution to the `ONLY-ONE-OBJECT` problem by means of a single membrane system from $\mathcal{AM}^0(-d, +ne)$.

Adapting currently used methodologies to new applications is an interesting future research line to improve existing results and obtain new ones. Besides, the search for new techniques to demonstrate the non-efficiency or the inability for certain membrane systems to solve some decision problems is critical when addressing the **P** *versus* **NP** and other interesting problems in the field of Computational Complexity Theory.

Acknowledgements

References

1. A. Alhazov, M.J. Pérez-Jiménez. Uniform solution to QSAT using polarizationless active membranes. In J. Durand-Lose, M. Margenstern (eds.) *Machines, Computations, and Universality 5th International Conference, MCU 2007, Orléans, France, September 10-13, 2007. Proceedings. Lecture Notes in Computer Science*, **4664** (2007), 122-133.
2. A. Cordón-Franco, M. A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez. Weak metrics on configurations of a P system. In Gh. Paun, A. Riscos, Á. Romero, F. Sancho (eds.) *Proceedings of the Second Brainstorming Week on Membrane Computing*, Report RGNC 01/2004, 2004, pp. 139-151.
3. M. A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez, F.J. Romero-Campero. On the power of dissolution in P systems with active membranes. In R. Freund, Gh. Păun, G. Rozenberg, A. Salomaa (eds.) *Membrane Computing, 6th International Workshop, WMC 2005, Vienna, Austria, July 18-21, 2005, Revised Selected and Invited Papers. Lecture Notes in Computer Science*, **3850** (2006), 224-240.
4. C.H. Papadimitriou. *Computational Complexity*, Addison–Wesley, Massachusetts, 1995.
5. Gh. Păun. P systems with active membranes: attacking **NP**–complete problems, *Journal of Automata, Languages and Combinatorics*, **6**, 1 (2001), 75-90. A preliminary version in *Centre for Discrete Mathematics and Theoretical Computer Science, CDMTCS Research Report Series*-102, May 1999, 16 pages.
6. Gh. Păun. Further twenty six open problems in membrane computing. In M. A. Gutiérrez-Naranjo, A. Riscos-Núñez, F.J. Romero-Campero, D. Sburlan (eds.) *Proceedings of the Third Brainstorming Week on Membrane Computing*, Fénix Editora, Sevilla, 2005, 249-262.

7. M.J. Pérez-Jiménez, Á. Romero-Jiménez, F. Sancho-Caparrini. Decision P systems and the $\mathbf{P} \neq \mathbf{NP}$ conjecture. In Gh. Păun, Gr. Rozenberg, A. Salomaa, C. Zandron (eds.) Membrane Computing 2002. *Lecture Notes in Computer Science*, **2597** (2003), 388-399. A preliminary version in Gh. Păun, C. Zandron (eds.) *Pre-proceedings of Workshop on Membrane Computing 2002*, MolCoNet project-IST-2001-32008, Publication No. 1, Curtea de Arges, Romanian, August 19-23, 2002, pp. 345-354.