Pumps Systems of Membranes

Gabriel CIOBANU

Institute of Computer Science of the Romanian Academy
Iaşi, Romania
E-mail: gabriel@iit.tuiasi.ro

Abstract. The reactivity of the living cells provides the real complexity of biological systems. This paper presents a reactive P system where the rules are triggered by certain conditions, and pumps are the parallel processors of a membrane. The computation of a reactive P system is given by parallel composition of conditional rules regarding the pumps, and by sequential and parallel composition of internal rules regarding the coordination activity of the membrane nucleus. Starting from this description, it is possible to define the operational semantics and various behavioural equivalences between membranes. Finally, some perspectives and open problems are presented. Overall this approach is a step toward a programming paradigm inspired by membrane activity.

1 Reactivity

Membrane computing is a branch of natural computing aiming to abstract a computability model from the structure and the functioning of the living cell [8]. The membrane structure is given by the cell compartments; membrane structure corresponds to a cell-like hierarchical arrangement of membranes. In each membrane there exist multisets of objects, together with some reaction rules. Currently the rules are applied "in a maximal parallel and non-deterministic manner". Consequently, it is easy to prove that these systems are computationally universal. Moreover, using a space-time trade-off, it is possible to get polynomial solutions to NP-complete problems whenever an enhanced parallelism is available (e.g., via membrane division). According to this view, each membrane is a powerful machine able to compute all Turing computable vectors of natural numbers. The goal of this short informal paper is to identify some elements, including hardware and software units of membrane computing, in order to get a more reasonable picture.

An important drawback of the current membrane systems is that they are not explicitly reactive. The living cells are not working in a maximal parallel and non-deterministic manner. Their activity is usually quite low, and they become more active as response to certain events. There are good reasons to consider the reactivity of the living cells as the root of their complex behaviour, and consequently it implies the real complexity of biological systems. The rules of a *reactive P system* are essentially conditional multiset rewriting rule, defined by a reaction condition and its related rewriting action. When a multiset of objects satisfies the reaction condition, then it can be rewritten in the way stated by its rewriting action. The reaction conditions depend on some global parameters,

as well as on some local parameters. A final result is obtained when a stable state is reached, and a first approximation of such a steady state is when no further reaction can take place. A more adequate notion of equilibrium can make the formalism more realistic.

2 Parallelism

When it is necessary, the rules of a reactive P system are applied in parallel. To clarify this parallelism, we should identify the processors able to execute the rules. The computing devices are inspired by the living cells. Therefore we look at the biological description of a membrane, namely at its functionality. The main functionality is given by the membrane transport. Transporters are of two general classes: carriers and channels. Carrier proteins (pumps) are transmembrane proteins, too large to move across the membrane. They have a general behaviour given by a conformation change between two conformations in which a solute binding site is accessible on one side of the membrane or the other. A pump transports only one or few molecules per conformational cycle. There are three classes of pumps: uniport, symport, and antiport. The pumps have their own rules, and these rules can be written as P system rules. Various classes $P_1, \ldots, P_k$ of pumps are defined as the set of similar pumps (Na pump, Ca pump, etc), together with their corresponding rules.

We advance the following idea: *pumps are the parallel processors of a membrane.* Then the parallel reactive program of a membrane is given by a parallel composition of conditional rules activating various classes of pumps. The corresponding reaction condition is usually related to the gradient of the specific substance s_i of the pump, and such a gradient condition can be expressed by comparing the inside and outside multiplicities of s_i. For instance, the parallel program of a membrane can be

$$In(s_i) < Out(s_i)/2 \longrightarrow P_i || In(s_j) > 3 * Out(s_j) \longrightarrow P_j || cond(s_k) \longrightarrow P_k \ ,$$

where P_i denotes the rules of the pump class regarding the substance s_i. If several disjoint pairs of elements satisfy the condition, then the comparisons and their corresponding rules can be performed in parallel. The parallel program is repeated until no reaction condition holds. It is also possible to consider a local clock, and the conditions are checked at each clock cycle.

3 Internal Rules

Cells are composed of various entities which interact in a very optimized way. Ways in which cells can be modeled as a computational device, together with several metaphors introduced to enable suitable reflections on cellular computational properties. We model the molecular interactions and coordination of cell processes using computer science concepts.

The genome is located in the cell nucleus, where it is safeguarded by the nuclear membrane from chemical changes that take place in the cell. The human genome contains 30,000 genes distributed in 24 different DNA molecules. The length of the DNA as extended molecules (2m) would vastly exceed the dimension of the nucleus (10 μm). In order to avoid entanglement and breakage, cells use DNA binding proteins (histone and nonhistone proteins) to bind to each DNA molecule and to wrap it up into more manageable chromosomes.

Within a chromosome, DNA is a relatively inert molecule. In order to be expressed, genetic information encoded in DNA has first to be transcribed into specific RNA molecules.

These molecules are then either used directly, as in the case of structural RNAs (transfer RNAs and ribosomal RNAs) or, as messenger RNA (mRNA) that are translated into proteins. The proteins will determine the cell chemical, physical and functional properties. There are several key points at which gene expression is controlled, including transcription initiation and termination, pre-mRNA processing, mRNA transport and translation. Gene regulatory elements represented by promoter and enhancer sequences each serve to bind general transcription factors and cell-specific transcription activators. The transcriptional activators and their corresponding binding sites in the promoter or enhancer provide the living cells an ability to respond to developmental or environmental stimuli. They act in a defined order and help basal transcription machinery to initiate transcription.

Inspired by this biological description, membranes activity is coordinated by some internal rules which are located in its nucleus. These internal rules correspond to DNA computations over strings. In membrane computing we use multisets of strings corresponding to chromosomes, and evolution rules corresponding to the transcription factors and activators. Moreover, it seems that the rules act in a sequential way; they eventually trigger some parallel actions of the pumps (membrane computational units). The rules are conditional, similar to the previous rules, but they involve specific factors and activators.

These internal rules are interacting with the pump rules. We consider two composition operations for the construction of complex membrane programs starting from basic computational elements (rules): sequential composition, and parallel composition. The intuition behind sequential composition $P_1 \circ P_2$ is that the stable multiset reached after the execution of program P_2 is given as argument to program P_1. On the other hand, the result of $P_1||P_2$ is obtained by executing the actions of P_1 and P_2 in parallel, terminating only when no reaction condition holds.

4 Conclusion

To summarize, we argue and informally present the reactive P systems. Our view of a membrane computation is given by parallel composition of conditional rules regarding the pump system of membranes, and by sequential and parallel composition of internal rules regarding the coordination activity of the membrane nucleus. This view add more realistic aspects to a membrane system, and we expect that it provides more restrictions than the rules applied "in a maximal parallel and non-deterministic manner". Consequently, we expect these systems, together with some additional requirements regarding the finite nature of the involved multisets, will not be computationally universal. This will open the way of some interesting technical results and classifications. An extended version of this paper will define the operational semantics for these systems, and then various behavioural equivalences between programs, and between membranes. Open problems are related to the compositionality of the parallel operator, as well as the decidability results regarding these systems.

Membrane computing is an intellectual challenge of finding whether the structure and the functioning of the living cell can provide new suggestions and paradigms to computer science. The main results of this new field are presented in the first book devoted to membrane systems [9]. Similar ideas and an approach related to this paper are presented in [6]. A message of this informal paper is that the pump systems of membranes are interesting subjects to study. We had some previous attempts to formalize these systems, starting with a description of sodium-potassium exchange pump in terms of the π-calculus process algebra [4]. A more detailed description, together with a temporal logic able to

describe the pump properties is given in [3]. Moreover, [3] presents a software tool able to check various properties and to provide confidence in the formal description of the pump. Regarding the internal rules of the membrane nucleus, a computer-oriented view is presented in [5]. An algorithmic description of the communicating membrane systems is presented in [2].

References

[1] B. Alberts et al. *Molecular Biology of the Cell*, 4th ed., Garland Science, 2002.

[2] G. Ciobanu. Distributed algorithms over communicating membrane systems. *BioSystems*, 70, 2 (2003), 123–133.

[3] G. Ciobanu. Software verification of biomolecular systems. In G. Ciobanu, G. Rozenberg (Eds.): *Modelling in Molecular Biology*, Natural Computing Series, Springer, 41–60, 2004.

[4] G. Ciobanu, V. Ciubotariu, B. Tanasă. A pi-calculus model of the Na pump, *Genome Informatics*, 469–472, Universal Academy Press, 2002.

[5] G. Ciobanu, B. Tanasă. Gene expression by software mechanisms, *Fundamenta Informaticae*, 49, 1-3 (2002), 67–80.

[6] D. Harel. A grand challenge for computing: Towards full reactive modelling of a multicellular animal. In Gh. Păun, G. Rozenberg, A. Salomaa (Eds.), *Current Trends in Computer Science* vol. I: Algorithms and Complexity. World Scientific, 559–568, 2004.

[7] F.C.P. Holstege, E.G. Jennings, J.J. Wyrick, T.I. Lee, C.J. Hengartner, M.R. Green, T.R. Golub, E.S. Lander, R.A. Young, Dissecting the Regulatory Circuitry of a Eukaryotic Genome, *Cell*, 95 (1998), 717–728.

[8] Gh. Păun. Computing with membranes. *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143.

[9] Gh. Păun. *Computing with Membranes: An Introduction*. Springer, 2002.