

Membrane computing

Michal Kováč

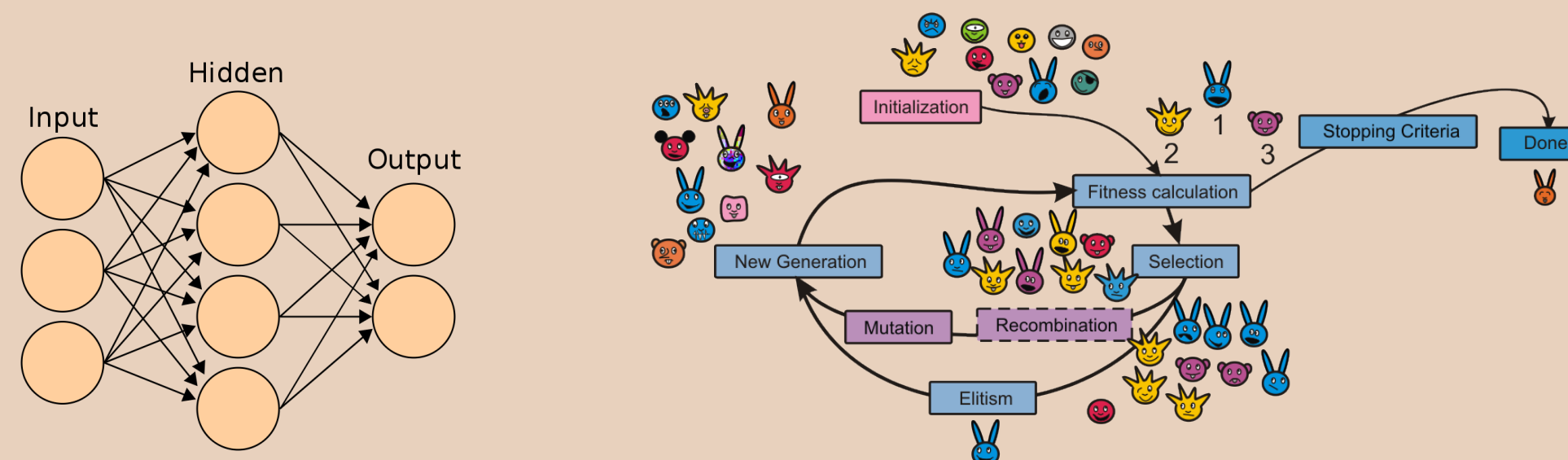
Supervisor: Damas Gruska

Katedra aplikovanej informatiky, FMFI UK, Mlynská Dolina, 842 48 Bratislava



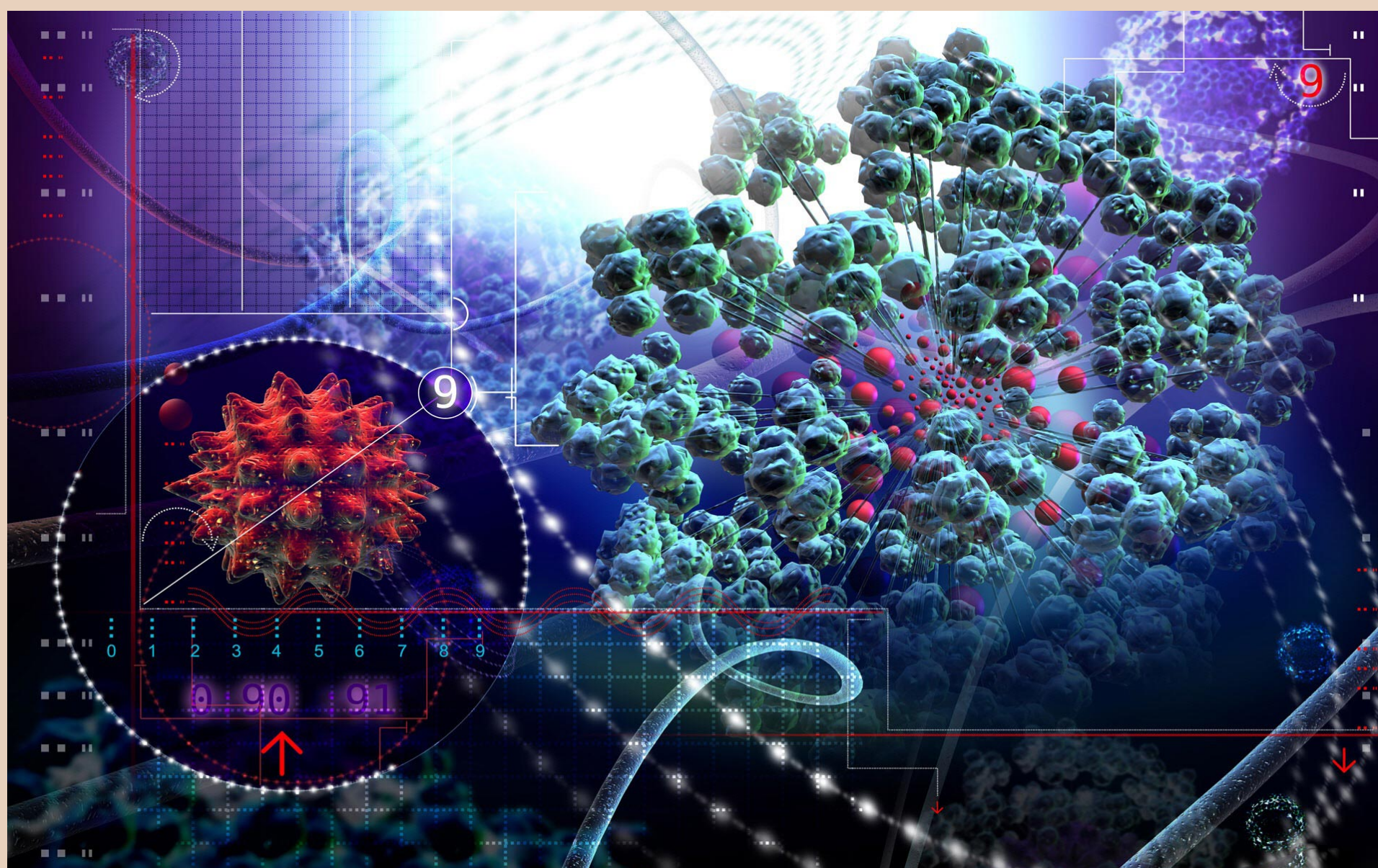
INTRODUCTION

Well established computation models motivated by biology such as neural networks and evolutionary algorithms has already proven that it is worth to be inspired by biology.



Other emerging areas are still awaiting for their more significant uses. One of them is the membrane computing. It is relatively young field of natural computing - in comparison: neural networks have been researched since 1943 and membrane systems since 1998[Păun, 1998].

Membrane systems are distributed parallel computing devices inspired by the structure and functionality of cells. Recently, many variants have been developed in order to simulate cells more realistically or just to improve the computational power.



MEMBRANE STRUCTURE

Biological systems usually have hierarchical structure where objects and information flows between regions, what can be interpreted as a computation process.

This hierarchical structure can be also interpreted as rooted tree as seen on the figures 1 and 2. The objects are stored in the tree nodes and can be sent through the edges.

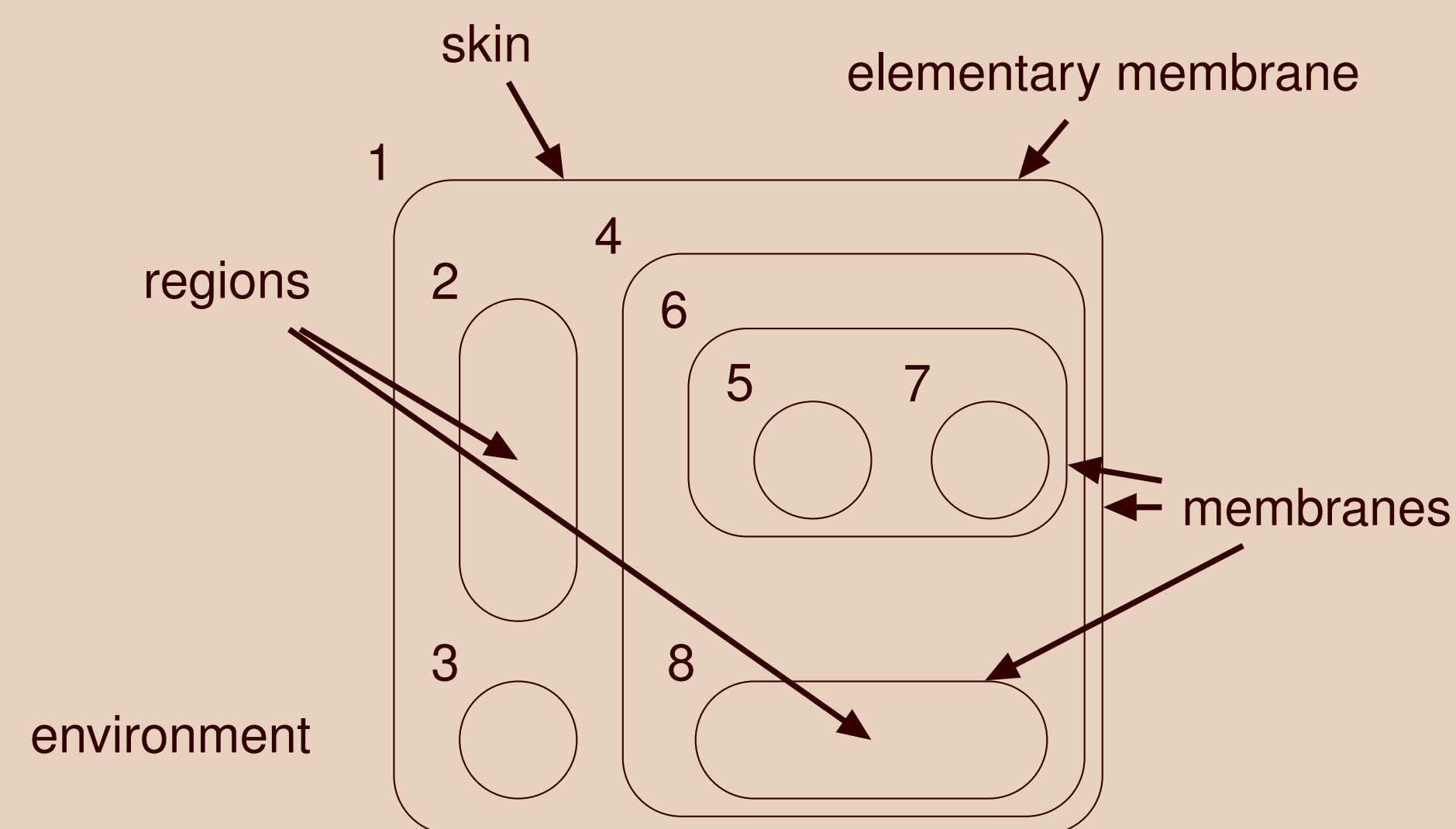


Figure 1: A membrane structure

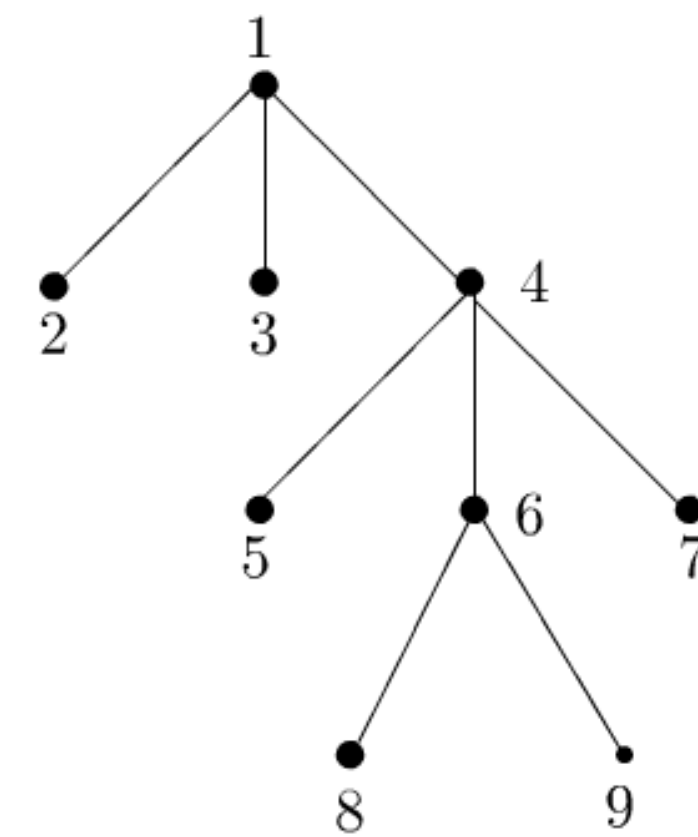


Figure 2: The tree describing the membrane structure from Figure 1

P SYSTEMS

Membranes and regions delimited by them has clearly 1:1 correspondence. Each region contains a multiset of objects. These objects can evolve according to evolution rules which are associated with membranes. Evolution rules are applied in a maximally parallel manner - in each step, a maximal multiset of rules is non-deterministically chosen and applied.

This computing device is called P system¹ and is Turing complete. Next figure demonstrates a computation of a P system for Fibonacci numbers.

VARIANTS

Since the first publication in 1998, huge amount of variants has been proposed. Some of them are Turing complete, others are not. Maximal parallelism is one of the most questioned attribute. P systems in sequential mode are not Turing complete. Various combinations of variants have been studied and some of them have been shown Turing complete also in sequential mode.

INHIBITORS

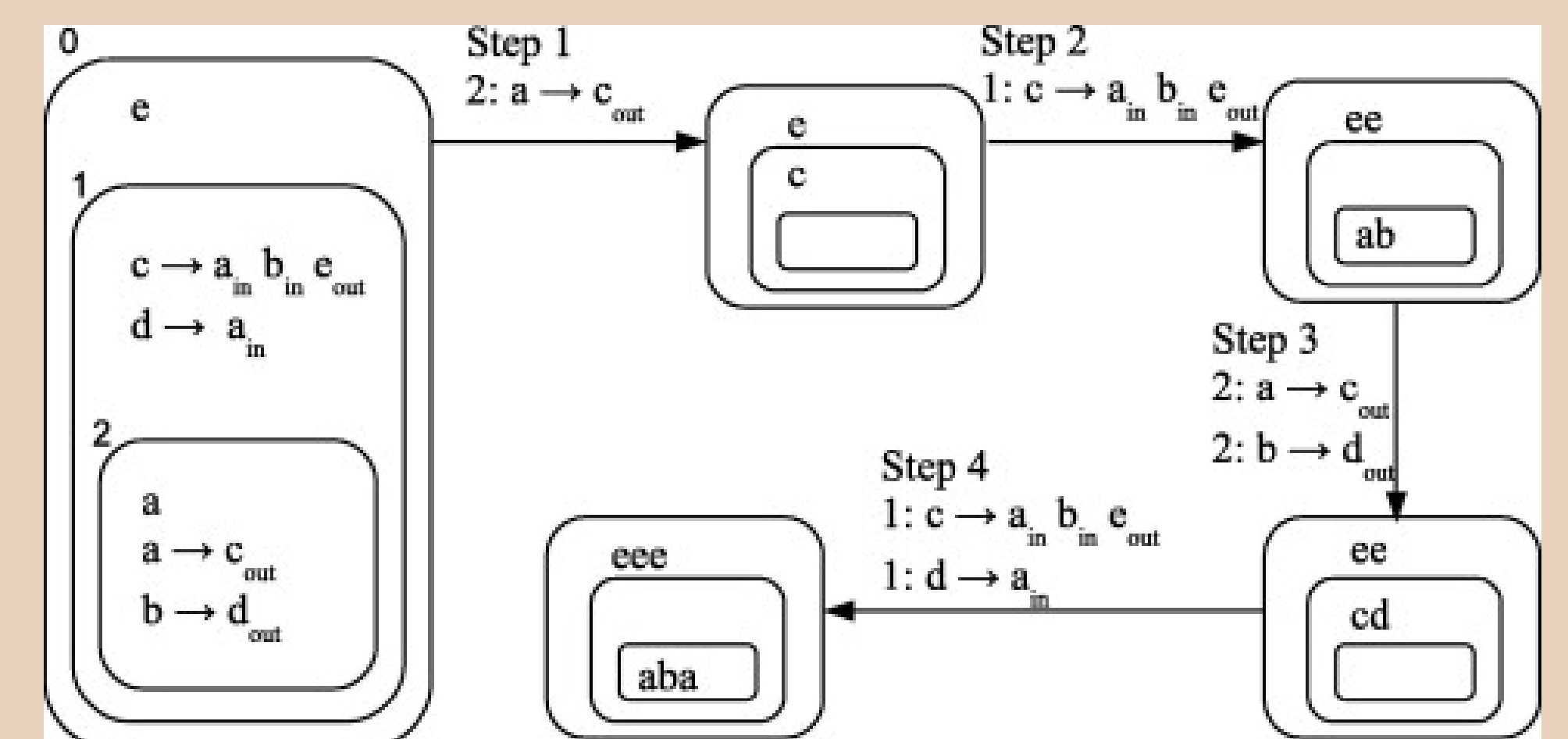
One of the variants is P system with inhibitors. Some of the objects (inhibitors) can be used in evolution rules such that their presence in the region prevents the application of the rule. One of our results is the proof that sequential P systems with inhibitors are Turing complete. It is proven by simulation of the maximally parallel P system, which is Turing complete. The challenging part was the simulation of maximally parallel step.

SAT IN LINEAR TIME

Linear time solutions to NP-complete problems by means of P systems are achieved by trading time (number of computation steps) for space (number of membranes and objects). This is inspired by the capability of cells to produce an exponential number of new membranes in linear time.

However, many simulators of P system are inefficient since they cannot handle the parallelism of these devices. Nowadays, we are witnessing the consolidation of the GPUs as a parallel framework to compute general purpose applications. The simulation of P systems with active membranes using GPUs is analysed in [Cecilia et al., 2010] and an efficient linear solution to the SAT problem is illustrated.

P SYSTEM COMPUTING FIBONACCI SEQUENCE



References

- [Cecilia et al., 2010] Cecilia, J. M., García, J. M., Guerrero, G. D., del Amor, M. A. M., Pérez-Hurtado, I., and Pérez-Jiménez, M. J. (2010). Simulating a p system based efficient solution to sat by using gpus. *Journal of Logic and Algebraic Programming*, 79:317–325. Membrane computing and programming.
- [Păun, 1998] Păun, G. (1998). Computing with membranes. Technical Report 208, Turku Center for Computer Science-TUCS. (www.tucs.fi).

¹P in P systems stands for surname of it's founder Gheorghe Păun.