Membrane systems
(Extended Abstract)

Michal Kováč[1]*

Supervisor: Damas Gruska[1]†

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

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.

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

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 consists of two multisets of objects: left and right part. When a rule is applied, the left part is subtracted from the multiset of objects present in the region and the right part is added. Evolution rules are applied in a maximally parallel manner - in each step, a maximal multiset of rules is nondeterministically chosen and applied.

This computing device is called P system[1] and is Turing complete.

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.

One of the variants we studied 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. Our result is the following theorem.

Theorem 1. *Sequential P systems with inhibitors are Turing complete.*

The proof is done by a simulation of maximally parallel P system. The challenging part is the simulation of one maximally parallel step, which we split into multiple phases. Phases are represented as a special object present in each membrane. So different membranes can have different phases. The SYNCHRONIZE phase is needed in order to provide communication between regions. In the RUN phase, the rewriting of the objects occurs, until all of the present rules are unusable. When all of the rules are unusable, phase is changed from RUN to SYNCHRONIZE and a special object SYNCTOKEN is sent through the parent membrane. The skin membrane (root of the hierarchy) collects all the SYNCTOKENs and broadcasts the signal to start the next computation step.

There is still an issue, how can we know when a rule is unusable. This is exactly where the inhibitors comes handy.

The rule $R_i : ab \to c$ is unusable when there is no a and no b object present. We can simulate this behavior with inhibitors:

$$RUN \to RUN, UNUSABLE_i|_{\neg a}$$
$$RUN \to RUN, UNUSABLE_i|_{\neg b}$$

References

[Păun, 1998] Păun, G. (1998). Computing with membranes. Technical Report 208, Turku Center for Computer Science-TUCS. (www.tucs.fi).

*kovac@fmph.uniba.sk

†gruska@fmph.uniba.sk

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