Memory about the 14 th BWMC: SN P systems vs. ESN P systems with Transmittable States

Patricia Ribes Metidieri

Universitat de Barcelona
Email: ribesmetidieri@gmail.com

Summary. The objectives of this memory are, on one hand, to provide a general overview about the topic of Membrane Computing, answering basic questions as what is it, which are its basic elements, which problems it allows to solve, its current limitations... and, on the other hand, to provide a more specific information about the model of Spiking Neural P systems in both its original formulation and on the variant of the model suggested on the 14th edition of the Brainstorming Week on Membrane Computing, the Extended Spiking Neural P systems with Transmittable States.

The motivation to further explore the topic of Spiking Neural P systems (SN P systems for short) comes from the idea that they could be a really suitable framework in order to model chain-reaction processes as the fission of ^{235}U taking place inside a nuclear reactor.

1 General overview

Membrane computing is a branch of Natural computing, which, based on the idea that the processes within a cell can be interpretated as computations, abstracts computing models from the architecture of living cells and their organisation in tissues, organs, ... Devices of this model are called P systems and are defined as cell-like structures consisting on a set of hierarchically arranged membranes where multisets of objects (that represent the chemicals in the cell) evolve following sets of rules which only apply in the membrane where they are implemented. [1]

Just as in a biological cell, the rules in a P system are applied at random, which often results in multiple solutions being encountered if the computation is repeated, thus P-systems are non-deterministic. Furthermore, all the rules which can be applied in a computation step are executed whenever they are applicable, so the system evolves in each step until no more rules can be implemented, i.e. the rules are applied in a maximally parallel manner.

A computation in this kind of system is a sequence of instantaneous transitions between configurations guided by the rules executed, until it reaches a state where

no further reactions are possible, which marks the end of the computation. Its result is all those objects contained in a particular 'results' membrane.

So far, the implementation of this biological computers "in vitro" has not been achieved and technical problems such as the fact that the theoretical model is based on the assumption that all the reactions within each step of computation lasts the same time while within a real cell each reaction is completed in a characteristic time, has not yet been solved.

However, on the lasts years, simulators as MeCoSim have been developed [2] and, within the computational complexity theory, Membrane Computing has proven itself a really powerful tool to solve NP-complete problems in polynomial even linear time. Finally, P-systems have also been used lately to model biological phenomena within the framework of Computational Systems Biology.

2 Spiking Neural P Systems: basic concepts

Spiking Neural (SN) P systems are a class of neural-like P systems in which one-membrane cells (called *neurons*) placed in the nodes of a directed graph send electric signals or spikes (represented by a single object denoted s) under certain conditions (given by the rules defined on each cell) along their axons to all the neurons connected by *synapses* (arcs of the graph). [3]

A computation in this model starts by fixing the number of spikes in the input neurons, which propagate the spikes through the net of connected cells. Each neuron fires after having accumulated a certain number of spikes and the result of the computation is the number of steps elapsed between two spikes of the labeled output neuron have been sent to the environment.

It is interesting to recall that the rules defined in each neuron are applied in a non-deterministic but sequential mode with at most one rule used in each step, though the maximal parallelism condition is implemented at a system level, since in each step of the computation all neurons which can evolve (use a rule) have to do it. In this model only two different kind of rules are defined: *firing or spiking rules* and *forgetting rules*.

On the one hand, spiking rules are of the form $E/s^r \rightarrow s;t$, where E represents the required content of the neuron for the rule to apply and $r \geq 1$, the spikes consumed when the rule is executed. Once the neuron is fired, it produces a spike which will be sent to other neurons after $t \geq 0$ time steps and the cell is assumed closed in the period between firing and spiking (which represents the *refractory period*).

On the other hand, forgetting rules are of the form $s^k \rightarrow \lambda$, which means that, when the neuron contains exactly $k \geq 1$ spikes, k spikes are removed ("forgotten").

Finally, on [4] SN P systems are shown to be *computationally complete* when the number of spikes inside each neuron is not bounded, i.e., when the cell can accumulate an arbitrary number of spikes inside. However, it is also shown that, if a more realistic model with a bounded number of spikes present in any neuron

along a computation is considered, these devices lose their capacity to generate all Turing computable set of numbers.

3 Extended Spiking Neural P systems with Transmittable States

Since the emergence of the SN P systems in [4] in 2006, many variants of this model have been proposed. Particularly, in the 14th edition of the Brainstorming Week on Membrane Computing a new variant was suggested, so in this section a brief analysis of the new proposals is discussed, explaining its advantages from the point of view of the computational completeness.

Formally, Extended Spiking Neural P systems with Transmittable States are defined [5] as

$$\Pi = (O, Q, \sigma_1, \sigma_2, \ldots, \sigma_n, in, out) \tag{1}$$

being

- $O = \{s\}$ the objects conforming the system, with s a spike.
- Q is a finite alphabet of states, where $polarizations \subseteq states$
- σ_i are the neurons of the systems, defined as $\sigma_i = (\alpha_i, n_i, f_i, R_i)$ where
  - $\alpha \in Q$ is the initial state.
  - n_i is the initial number of spikes
  - f_i is the state combining function
  - R_i is a finite set of rules of the form

$$\alpha/a^c \rightarrow (t_1, a^{k_1}, \beta_1), \ldots, (t_m, a^{k_m}, \beta_n) \tag{2}$$

    1. send k_j spikes to neuron t_j after having accumulated a^c spikes, for $1 \leq j \leq m$,
    2. send state b_j to neuron σ_j, combine them using f_i and set the new state of neuron σ_i, combine them using f_i, and set the new state of neuron σ_i.

  and

$$\alpha/a^c \rightarrow \lambda \tag{3}$$

Comparing this definition with the given for SN P systems on the previous section, the new contributions can be appreciated. Firstly, the firing rules incorporate the elements α and β, which denote the initial and final states of a neuron after each step of the computation. This incorporation modifies the process of spiking, since now not only a spike, but the state of the neuron, is sent along the axon in the firing process.

Secondly, the state combining function, f_i is added in the definition of the neurons conforming the system in order to compute the state of the neuron receiving states of all those connected to it through synapses.

For instance, if the states being passed are polarisations $Q = \{+, -, 0\}$ (which represent the electrical charge of the membrane), a possible state combining function would assign positive polarisation to the neuron receiving the spikes if more positive than negative polarisations were received, negative polarisation in the opposite case and null polarisation if the number of received + and − polarisations were the same.

Finally, the time between firing and spiking is 0 for all the neurons.

In contrast with SN P systems, where computational completeness is achieved by allowing the non-realistic notion that the number of spikes within each cell is not bounded, the ESN P systems with Transmittable States are powerful enough for universal computation considering the number of spikes in each active neuron reduced to 1 and replacing the unbounded number of spikes to an unbounded number of neurons.

This model can reproduce the rules of Conway's Game of Life using only 2 states $Q = \{0, 1\}$ (which represent "dead" and "alive") and, since this kind of cellular automata has the power of a universal Turing machine, by reproducing the rules of the Game, the computational completeness of the ESN P systems with Transmittable States is proved.

References

1. Gh. Păun *Membrane Computing: An Introduction*
2. J. M. Cecilia, J. M. García, G. D. Guerrero, M. A. Martínez-del-Amor, I. Pérez-Hurtado, M. J. Pérez-Jiménez *Simulation of P systems with active membranes on CUDA*
3. J. Wang, L. Zou, H. Peng, G. Zhang *An extended Spiking Neural P System for fuzzy knowledge representation*
4. M. Ionescu, Gh. Păun, T. Yokomori *Spiking Neural P Systems*
5. R. Freud, S. Ivanov *Extended SNP systems with States*