Computability of Spiking Neural P Systems with Anti-Spikes

Venkata Padmavati Metta, Kamala Krithivasan and Deepak Garg

Abstract

Biologically inspired computing which is a branch of natural computing is the field of investigation that draws upon metaphors or theoretical models of biological systems in order to design computational tools or systems for solving complex problems. Spiking neural P systems [2] (shortly called SN P systems) are parallel and distributed computing models inspired by the neurobiological behaviour of neurons sending electrical pulses of identical voltages called spikes to neighbouring neurons. Classic SN P system is pictorially represented as a directed graph where nodes represent the neurons having spiking and forgetting rules. The rules involve the spikes present in the neuron in the form of occurrences of a symbol a. The arcs indicate the synapses among the neurons. Similar to the neurons in the brain, the neurons in an SN P system also fire in parallel, with each neuron using only one rule in each step. Hence, in a classic SN P system there is only one type of objects called spikes which are moved, created and destroyed but never modified.

SN P system with anti spikes introduced in [3], is a variant of an SN P system consisting of two types of objects, spikes(denoted as a) and anti-spikes(denoted as $\overline{a}$). The inhibitory impulses/spikes are represented using anti-spikes. The anti-spikes behaves in a similar way as spikes by participating in spiking and forgetting rules. They are produced from usual spikes by means of usual spiking rules; in turn, rules consuming anti-spikes can produce spikes or anti-spikes (here we avoid the rule anti-spike producing anti-spike). The SN P system with anti-spikes consists of an implicit annihilation rule of the form $a\overline{a} \rightarrow \lambda$; if an anti-spike and a spike meet in a given neuron, they annihilate each other. This rule has the highest priority and does not consuming any time. The initial configuration of the system is represented by the number of spikes/anti-spikes present in each neuron. A computation halts if it reaches a configuration where no rule can be used. There are various ways of using such a device [1]. In the generative mode, one of the neurons is considered to be the output neuron, and its spikes are sent to the environment. With any computation halting or not we associate a spike train, a sequence of digits of 0 and 1, with 1 and 0 appearing in positions which indicate the steps when the output neuron sends spikes and anti-spikes respectively, out of the system. When both an input and an output neuron are considered, the system can be used as a transducer, both for strings and infinite sequences, as well as for computing numerical functions. Spikes can be introduced in the former one, at various steps, while the spikes of the output neuron are sent to the environment. The moments of time when a spike is emitted by

the output neuron are marked with 1, the moments of anti-spikes are marked with 0. The binary sequence obtained in this way is called the spike train of the system. A binary sequence is similarly associated with the spikes entering the system. In the transducer mode, a large class of (Boolean) functions can be computed. Thus SN P system with anti-spikes allows the modification of spikes and anti-spikes and is proved as computationally complete.

Classic Spiking neural P systems are used to simulate arithmetic and logic operations where the presence of spike is encoded as 1 and absence of spike as 0 [4] and the negative integers are not considered. They are also used to simulate the boolean circuits [5], with two spikes emitting out of the system encoded as 1 and one spike as 0. In this paper we use SN P systems with anti-spikes to simulate logic gates as the anti-spike and spike encode Boolean values 0 and 1 in a natural way. We design SN P systems with anti-spikes simulating the operations of AND, OR and NOT gates. The output of the system is 0(hence false) if the output neuron sends an anti-spike and output is 1(true) if a spike is sent to the environment. Hence we present a way to simulate any boolean circuit using these fundamental gates and synchronising SN P system with anti-spikes to establish synchronization (Each sub-system (that simulates the gates AND, OR, and NOT) to receive the input from the above gate(s) at the same time.) among the gates to output the correct result.

In this paper we also consider SN P system with anti-spikes as simple arithmetic device that can perform the arithmetic operations like 2's complement, addition and subtraction with input and output in binary form. The binary sequence of 0 and 1 are encoded as anti-spike and spike respectively and in each time step input is provided bit-by-bit starting from least significant bit. They can represent the negative numbers in 2's complement form, there by simulating the arithmetic operations on negative numbers. In this paper we simulate three arithmetic operations - 2's complement, addition and subtraction. This motivates the implementation of CPU using SN P system with anti-spikes.

References

1. Gheorghe, P.: Spiking Neural P Systems Used as Acceptors and Transducers, CIAA, LNCS **4783** (2007) 1–4.
2. Ionescu, M., Paun, Gh., Yokomori, T.: Spiking Neural P Systems, Fund. Infor. **71** (2006) 279–308.
3. Linqiang, P., Gheorghe, P.: Spiking Neural P Systems with Anti-Spikes, Int. J. of Computers, Communications and Control, **4** (2009) 273–282.
4. Miguel, A. G., Alberto, L.: First Steps Towards a CPU Made of Spiking Neural P Systems, Int. J. of Computers, Communications and Control, **4** (2009) 244–252.
5. Mihai, I., Dragos, S.: Some Applications of Spiking Neural P systems, J. of Computing and Informatics, **27** (2008) 515–528.