Neurocomputing 000 (2017) 1–9

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Semantics of deductive databases with spiking neural P systems

Daniel Díaz-Pernil[a], Miguel A. Gutiérrez-Naranjo[b,*]

[a] CATAM Research Group - Department of Applied Mathematics I, University of Seville, Spain
[b] Department of Computer Science and Artificial Intelligence, University of Seville, Spain

ARTICLE INFO

Article history:
Received 20 October 2016
Revised 23 January 2017
Accepted 2 July 2017
Available online xxx

Communicated by Dr Hongyi Li

Keywords:
Membrane computing
P systems
Neural-symbolic integration

ABSTRACT

The integration of symbolic reasoning systems based on logic and connectionist systems based on the functioning of living neurons is a vivid research area in computer science. In the literature, one can find many efforts where different reasoning systems based on different logics are linked to classic artificial neural networks. In this paper, we study the relation between the semantics of reasoning systems based on propositional logic and the connectionist model in the framework of membrane computing, namely, spiking neural P systems. We prove that the fixed point semantics of deductive databases without negation can be implemented in the spiking neural P systems model and such a model can also deal with negation if it is endowed with anti-spikes and annihilation rules.

1. Introduction

Searching new ways for automating reasoning is one of the engines in computer science. In the last decades, the two main research areas devoted to this aim, namely, connectionist systems ans logic-based systems, have explored very different techniques, approaches and representation systems.

On the one hand, logic-based systems are based on a symbolic representation of the knowledge and derivation rules which allow to obtain new formulae from previous ones. Many efforts have done in order to prove the correctness and soundness of the rules depending on the expressiveness of the language and the chosen semantics.

On the other hand, the traditional connectionist systems used to handle automatic reasoning are based on artificial neural nets inspired in the network of biological neurons in a human brain. These nets have been also widely studied and provide a point of view where the data is encoded in real numbers and the synapses between neurons determine the flow of information.

The integration of both paradigms is a vivid area in artificial intelligence (see, e.g., [1–3]). In the framework of membrane computing, several studies have been presented where P systems are used for representing logic-based information and performing reasoning by the application of bio-inspired rules (see [4,5]). These papers study approaches based on cell-like models, as P systems with active membranes, and deal with procedural aspects of the computation. The approach in this paper is different in both senses.

On the one hand, the connectionist model of P systems is considered, i.e, the model of P system inspired by the neurophysiological behavior of neurons sending electrical impulses along axons to other neurons (the so-called spiking neural P systems or SN P systems for short). On the second hand, we consider the semantics of propositional deductive databases in order to show how SN P systems can deal with logic-based representing and reasoning systems.

One of the key features of the integrate-and-fire formal spiking neuron models [6] (and, in particular, of the SN P systems) is the use of the *spikes* as support of the information. Such spikes are short electrical pulses (also called action potentials) between neurons and can be observed by placing a fine electrode close to the soma or axon of a neuron. From the theoretical side, it is crucial to consider that all the biological spikes of an alive biological neuron look alike. This means that we can consider a bio-inspired binary code which can be used to formalize logic-based semantics: the emission of one spike will be interpreted as *true* and the absence of spikes will be interpreted as *false*.

As we will show below, SN P systems provide a natural way for dealing with this binary behavior in a connectionist model. SN P systems suffice for dealing with the semantics of propositional logic systems which do not use negation. The semantics of reasoning systems with negated information, even in the propositional case, needs to add new elements for dealing with the difference between negated information and absence of information (see [7]).

The literature of SN P system provides an efficient tool for dealing with such negated information: the use of anti-spikes and

* Corresponding author.
E-mail addresses: sbdani@us.es (D. Díaz-Pernil), magutier@us.es (M.A. Gutiérrez-Naranjo).

http://dx.doi.org/10.1016/j.neucom.2017.07.007
0925-2312/

annihilation rules. SN P systems with anti-spikes were presented in [8] as a formalization of the idea of inhibiting in some way the communication between neurons. Such extended model has been widely studied (see, e.g., [9–11]) and it has inspired the use of negative information (see, e.g., [12–14]). Recently, the study of SN P systems has been extended with different features, see e.g., SN P systems *with thresholds* [15], SN P systems *with request rules* [16], SN P systems *with structural plasticity* [17] or *cell-like* SN P systems [18]. Among the recent contributions, the approaches shown in [19–21] deserve to be cited. In such papers, the interaction between reasoning systems and membrane computing is also studied. Instead of dealing with propositional logic, the authors consider fuzzy logic and the applications focus on fuzzy representation of the knowledge and fault diagnosis.

The main result of this paper[1] is to prove that given a reasoning system based on propositional logic it is possible to find an SN P system with the same declarative semantics. We prove it in both cases: when the reasoning system involves negation and when it does not. A declarative semantics for a rule-based propositional system is usually given by selecting models which satisfy certain properties. This choice is often described by an operator mapping interpretations to interpretations. In this paper we consider the so-called *immediate consequence operator* due to van Emden and Kowalski [22]. It is well-known that for a rule based system without negation *KB*, such operator is order continuous and its least fix point coincides with the least model of *KB*. We adapt the definition of the immediate consequence operator to a restricted form of SN P system and we prove that a least fix point, and hence a least model is obtained for the given reasoning system. The monotonicity is lost if negation is allowed, but even in this case, the immediate consequence operator can be computed with membrane computing techniques.

The paper is organized as follows: firstly, we recall some aspects about SN P systems and the semantics of deductive databases. In Section 3 we prove that standard SN P systems can deal with the semantics of deductive databases if they do not involve negation. In Section 4 it is shown that endowing SN P systems with anti-spikes and annihilation, then such devices can deal with the semantics of deductive databases with negation. Finally, some conclusions and topics for further discussion are provided in the last section.

2. Preliminaries

We assume the reader to be familiar with basic elements about membrane computing and the semantics of rule-based systems. Next, we briefly recall some definitions. We refer to [23] for a comprehensive presentation of the former and [7,24,25] for the latter.

2.1. Spiking neural P systems

SN P systems were introduced in [26] with the aim of incorporating membrane computing ideas with spike-based neuron models. It is a class of distributed and parallel computing devices, inspired by the neurophysiological behavior of neurons sending electrical impulses (*spikes*) along axons to other neurons.

In SN P systems the cells (also called *neurons*) are placed in the nodes of a directed graph, called the *synapse graph*. The contents of each neuron consist of a number of copies of a single object type, called the *spike*. Every cell may also contain a number of *firing* and *forgetting* rules. Firing rules allow a neuron to send information to other neurons in the form of *spikes* which are accumulated at the target cell. The applicability of each rule is determined by checking the contents of the neuron against a regular set associated with the rule. In each time unit, if a neuron can use one of its rules, then one of such rules must be used. If two or more rules could be applied, then only one of them is non-deterministically chosen. Thus, the rules are used in the sequential manner in each neuron, but neurons function in parallel with each other. As usually happens in membrane computing, a global clock is assumed, marking the time for the whole system, and hence the functioning of the system is synchronized.

Formally, an SN P system of the degree $m \geq 1$ is a construct[2]

$$\Pi = (O, \sigma_1, \sigma_2, \ldots, \sigma_m, syn)$$

where:

1. $O = \{a\}$ is the singleton alphabet (a is called *spike*);
2. $\sigma_1, \sigma_2, \ldots, \sigma_m$ are *neurons*, of the form $\sigma_i = (n_i, R_i)$, $1 \leq i \leq m$, where:
   (a) $n_i \geq 0$ is the *initial number of spikes* contained in σ_i;
   (b) R_i is a finite set of *rules* of the following two forms:
      (1) *firing* rules $E/a^p \to a^q$, where E is a regular expression over a and $p, q \geq 1$ are integer numbers[3];
      (2) *forgetting* rules $a^s \to \lambda$, with s an integer number such that $s \geq 1$;
3. $syn \subseteq \{1, 2, \ldots, m\} \times \{1, 2, \ldots, m\}$, with $(i, i) \notin syn$ for $1 \leq i \leq m$, is the directed graph of *synapses* between neurons.

Rules of type (1) are firing rules, and they are applied as follows. If the neuron σ_i contains k spikes, $k \geq p$, and a^k belongs to the language $L(E)$ associated to the regular expression E, then the rule $E/a^p \to a^q$ can be applied. The application of this rule means removing p spikes (thus only $k - p$ remain in σ_i), the neuron is fired, and it produces q spikes which are sent immediately to all neurons σ_j such that $(i, j) \in syn$. The rules of type (2) are forgetting rules and they are applied as follows: if the neuron σ_i contains exactly s spikes, then the rule $a^s \to \lambda$ from R_i can be used, meaning that all s spikes are removed from σ_i. If a rule $E/a^p \to a^q$ of type (1) has $E = a^p$, then we will write it in the simplified form $a^p \to a^q$. In each time unit, if a neuron σ_i can use one of its rules, then a rule from R_i must be used. Let us remark that it is possible that two or more rules can be applied in a neuron, and in that case only one of them is non-deterministically chosen regardless its type.. The jth configuration of the system is described by a vector $\mathbb{C}_j = (t_1, \ldots, t_m)$ where t_k represents the number of spikes at the neuron σ_k in such configuration. The initial configuration is $\mathbb{C}_0 = (n_1, n_2, \ldots, n_m)$. Using the rules as described above, one can define transitions among configurations. Any sequence of transitions starting in the initial configuration is called a computation. A computation halts if it reaches a configuration where no rule can be used. Generally, a computation may not halt. If it halts, the last configuration is called a *halting* configuration.

A useful extension to the model presented above was introduced by adding anti-spikes as a formalization of the idea of inhibiting in some way the communication between neurons (see [8]). This extension leads to the definition of *SN P systems with anti-spikes*. In this extension a further object, $\overline{a}$, is added to the alphabet O, and the spiking and forgetting rules are of the forms $E/b^p \to b'^q$ and $b^p \to \lambda$ where E is a regular expression over a or over $\overline{a}$, while $b, b' \in \{a, \overline{a}\}$ and $p, q \geq 1$. As above, if $L(E) = b^p$, then we write the first rule as $b^p \to b'^q$. The rules are used as in a usual SN P system, with the additional fact that a and $\overline{a}$ cannot stay

[1] A preliminary version of this paper appeared in the Proceedings of the 14th Brainstorming Week on Membrane Computing (BWMC2016).

[2] We provide a definition without delays, input or output neurons because these features are not used in this paper.

[3] In the general case, the restriction $p \geq q$ is imposed. In this paper, we will also consider rules $E/a \to a^2$. A more complex design of our solutions satisfying $p \geq q$ is possible, but we prefer to include rules $E/a \to a^2$ for the sake of a simpler design.

Please cite this article as: D. Díaz-Pernil, M.A. Gutiérrez-Naranjo, Semantics of deductive databases with spiking neural P systems, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.07.007