*Journal of Universal Computer Science, vol. 13, no. 11 (2007), 1707-1721*
*submitted: 10/9/06, accepted: 19/11/07, appeared: 28/11/07 © J.UCS*

# Spiking Neural P Systems with Astrocyte-Like Control[1]

**Gheorghe Păun**
(Institute of Mathematics of the Romanian Academy
PO Box 1-764, 014700 Bucharest, Romania
and
Research Group on Natural Computing, Department of Computer Science
Sevilla University, Avda Reina Mercedes s/n, 41012 Sevilla, Spain
Email: george.paun@imar.ro, gpaun@us.es)

**Abstract:** Spiking neural P systems are computing models inspired from the way the neurons communicate by means of spikes, electrical impulses of identical shapes. In this note we consider a further important ingredient related to brain functioning, the astrocyte cells which fed neurons with nutrients, implicitly controlling their functioning. Specifically, we introduce in our models only one feature of astrocytes, formulated as a control of spikes traffic along axons. A normal form is proved (for systems without forgetting rules) and decidability issues are discussed.

**Key Words:** Membrane computing, neural computing, spiking neural P system, astrocyte

**Category: ACM Computing Classification:** F.1.1 (Computation by Abstract Devices), F.4.2 (Grammars and Other Rewriting Systems), F.4.3 (Formal Languages), G.2.2 (Graph Theory)

## 1 Introduction

Spiking neural P systems (in short, SN P systems) were recently introduced in [Ionescu et al. 2006] as a rather particular class of P systems, with inspiration from the way neurons communicate by sending to each other *spikes*, electrical impulses of identical forms (we refer to [Gerstner and Kistler 2002] and [Maass and Bishop 1999] for details related to bio-neurology and neural computing based on spiking). In short, neurons are placed in the nodes of a graph, with the arcs representing the synapses; spikes are present in neurons, in the form of occurrences of the symbol $a$, representing the spike; rules for spiking and for forgetting spikes are also associated with the neurons. Using these rules in a synchronous mode (each neuron which can fire has to do it, hence the neurons evolve in parallel), with each neuron using in each time unit only one rule. An input can be provided to such a net, in the form of the sequence of spikes entering a specified neuron (such a sequence is called *spike train*) and an output can be defined in the form of the spike train produced by a specified output neuron.

[1] C. S. Calude, G. Stefanescu, and M. Zimand (eds.). *Combinatorics and Related Areas. A Collection of Papers in Honour of the 65th Birthday of Ioan Tomescu.*

Several variants can be considered – see the references given in the end of the paper, where we have collected most of the papers we know at this moment about SN P systems. For further developments, the reader is advised to consult the membrane computing web site at [WebPage].

Still, an important ingredient of neuro-biology is missing from the previous model: the *astrocytes*. They are cells which play an essential role in neurons functioning and interaction, by differently feeding them with nutrients, depending on their activity. More specifically, astrocytes are cells which sense at the same time the spike traffic along several neighboring axons, and fed the respective neurons (e.g., with calcium) depending on the spikes frequency. We do not enter here into biological details, but we refer to [Binder et al. 2007], where a first attempt was made to capture the "excitatory and inhibitory" role of astrocytes. See also [Volterra and Meldolesi 2005].

The present paper takes the idea from [Binder et al. 2007], but considers a much simplified version of SN P systems with astrocytes, namely, with only inhibitory astrocytes, working in a rather restricted manner: an astrocyte checking several axons leaves to pass only one spike along them, suppressing all others.

Actually, like in [Binder et al. 2007], the astrocytes are controlling the synapses, not the axons; in the end of the paper we will also briefly discuss the more realistic case of astrocytes associated with axons.

We deal here with two types of problems, deadlocks and normal forms. The former issue is related to the previous simplified definition of astrocyte functioning: when several astrocytes control the same synapses, it is possible that their action is contradictory (see Section 3 below). Deciding whether such a situation occurs during the functioning of a system is an important question – but, as expected, this is an undecidable property of SN P systems. However, on the good side (see Section 4), we can simplify the astrocytes without losing (computing) power: given an SN P system with astrocytes controlling several synapses, without forgetting rules, it is possible to construct an equivalent system with each astrocyte controlling only two synapses.

The present investigation is a preliminary one – many further research topics are mentioned along the paper and in the last section of it.

## 2 Definitions

We start by introducing the basic class of SN P systems, without astrocytes. The reader is assumed to have some familiarity with (basic elements of) language theory, e.g., from [Rozenberg and Salomaa 1997].

A *spiking neural P system* of degree $m \geq 1$ is a construct of the form

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

where:

1. $O = \{a\}$ is the singleton alphabet ($a$ is called *spike*);

2. $\sigma_1, \ldots, \sigma_m$ 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 by the neuron;

   b) $R_i$ is a finite set of *rules* of the following two forms:

      (1) $E/a^c \to a; d$, where $E$ is a regular expression with $a$ the only symbol used, $c \geq 1$, and $d \geq 0$;

      (2) $a^s \to \lambda$, for some $s \geq 1$, with the restriction that $a^s \in L(E)$ for no rule $E/a^c \to a; d$ of type (1) from $R_i$;

3. $syn \subseteq \{1, 2, \ldots, m\} \times \{1, 2, \ldots, m\}$ with $(i, i) \notin syn$ for $1 \leq i \leq m$ (*synapses*);

4. $in, out \in \{1, 2, \ldots, m\}$ indicate the *input* and the *output neuron*, respectively.

The rules of type (1) are *firing* (we also say *spiking*) *rules*, and they are applied as follows: if the neuron contains $k$ spikes, $a^k \in L(E)$ and $k \geq c$, then the rule $E/a^c \to a; d$ can be applied, and this means that $c$ spikes are consumed, only $k - c$ remain in the neuron, the neuron is fired, and it produces a spike after $d$ time units (a global clock is assumed, marking the time for the whole system, hence the functioning of the system is synchronized). If $d = 0$, then the spike is emitted immediately, if $d = 1$, then the spike is emitted in the next step, and so on. In the case $d \geq 1$, if the rule is used in step $t$, then in steps $t, t+1, t+2, \ldots, t+d-1$ the neuron is *closed*, and it cannot receive new spikes (if a neuron has a synapse to a closed neuron and tries to send a spike along it, then the spike is lost). In step $t + d$, the neuron spikes and becomes again open, hence can receive spikes (which can be used in step $t + d + 1$). A spike emitted by a neuron $\sigma_i$ replicates and goes to all neurons $\sigma_j$ such that $(i, j) \in syn$. If in a rule $E/a^c \to a; d$ we have $L(E) = \{a^c\}$, then we write it in the simpler form $a^c \to a; d$.

The rules of type (2) are *forgetting* rules, and they are applied as follows: if the neuron contains exactly $s$ spikes, then the rule $a^s \to \lambda$ can be used, and this means that all $s$ spikes are removed from the neuron.

In each time unit, in each neuron which can use a rule we have to use a rule, either a firing or a forgetting one. Because two firing rules $E_1/a^{c_1} \to a; d_1$ and $E_2/a^{c_2} \to a; d_2$ can have $L(E_1) \cap L(E_2) \neq \emptyset$, it is possible that two or more rules can be applied in a neuron, and then one of them is chosen non-deterministically.

Note however that we cannot interchange a firing rule with a forgetting rule, as all pairs of rules $E/a^c \to a; d$ and $a^s \to \lambda$ have disjoint domains, in the sense that $a^s \notin L(E)$. (Of course, this restriction can be removed, but this point is not important here.)

The initial configuration of the system is described by the numbers $n_1, n_2, \ldots, n_m$ of spikes present in each neuron. During a computation, the system is described both by the numbers of spikes present in each neuron and by the state of each neuron, in the open-closed sense. Specifically, if a neuron is closed, we have to specify the number of steps until it will become again open, i.e., the configuration is written in the form $\langle p_1/q_1, \ldots, p_m/q_m \rangle$; the neuron $\sigma_i$ contains $p_i \geq 0$ spikes and will be open after $q_i \geq 0$ steps ($q_i = 0$ means that the neuron is already open).

Using the rules as suggested above, we can define transitions among configurations. A transition between two configurations $C_1, C_2$ is denoted by $C_1 \Longrightarrow C_2$. Any sequence of transitions starting in the initial configuration is called a *computation*. A computation halts if it reaches a configuration where all neurons are open and no rule can be used. With any computation, halting or not, we associate a *spike train*, the sequence $t_1, t_2, \ldots$ of natural numbers $1 \leq t_1 < t_2 < \ldots$, indicating time instances when the output neuron sends a spike out of the system (we also say that the system itself spikes at that time).

In [Ionescu et al. 2006], with any spike train containing at least two spikes one associates a result, in the form of the number $t_2 - t_1$; we say that this number is computed by $\Pi$. The set of all numbers computed in this way by the system $\Pi$ is denoted by $N_2(\Pi)$.

This idea was extended in [Păun et al. 2006] to several other sets of numbers which can be associated with a spike train: taking into account the intervals between the first $k$ spikes, $k \geq 2$ (direct generalization of the previous idea), or between all intervals; only halting computations can be considered or arbitrary computations; an important difference is between the case when all intervals are considered and the case when the intervals are taken into account alternately (take the first interval, ignore the next one, take the third, and so on); the halting condition can be combined with the alternating style of defining the output. It is also possible to consider SN P systems working in the recognizing mode: we start the computation from an initial configuration, and we introduce in the input neuron two spikes, in steps $t_1$ and $t_2$; the number $t_2 - t_1$ is recognized by the system if the computation eventually halts. We can also use an SN P system as a computing device, passing from a number $n$ introduced in the system as above and producing the value $f(n)$ of a given function. Furthermore, the spike train itself can be considered as the result of a computation, codified as a string of bits: we write 1 for a step when the system outputs a spike and 0 otherwise. The halting computations will thus provide finite strings over the binary alphabet, the non-halting computations will produce infinite sequences of bits. If also an input is provided, then a transducer is obtained, translating input binary strings into binary strings. Details can be found in the references.

Some of the features of SN P systems as introduced above can be omitted without decreasing the computational power. In particular, the forgetting rules can be omitted – see [Ibarra et al. 2007] – that is why from now on we do no longer consider forgetting rules.

In what follows, on the one hand, we simplify the model, by considering only rules without a delay (all spiking rules are of the form $E/a^c \to a; 0$, hence we write them in the form $E/a^c \to a$), on the other hand, we add a further component to the system,

$$astro \subseteq syn^{\leq k},$$

where $k \geq 2$ is a natural number, and $X^{\leq n}$ denotes the set of strings of length at most $n$ over $X$.

The idea is that an astrocyte $x \in astro$ controls all synapses specified in $x$ and in each step, if along these synapses at least one spike is transmitted, then exactly one spike is selected, non-deterministically, while all others are removed. Thus, if no spike goes along the synapses in $x$, then nothing happens; if only one spike is transmitted along these synapses, then again nothing is changed, the spike is left to go. If, however, at least two spikes are passing along synapses in $x$, then all of them but one are simply removed and the one which was (non-deterministically) chosen is sent to its destination. (The ordering of synapses present in a string in *astro* plays no role, hence all permutations of a string represent the same astrocyte.)

Note that the presence of an astrocyte $x \in syn$ imposes no restriction, hence such astrocytes are ignored. Then, the presence of astrocytes implies a restriction on the work of the system (our astrocytes are a particular case of those considered in [Binder et al. 2007], with an inhibitory role). Furthermore, the use of astrocytes adds a new degree of non-determinism to the functioning of the system, by the branching due to the non-deterministic choice of the surviving spike.

For a given SN P system $\Pi = (O, \sigma_1, \ldots, \sigma_n, syn, astro, in, out)$, we say that it is of astro-degree $k$ if $k = \max\{|x| \mid x \in astro\}$.

Graphically, an SN P system with astrocytes is represented as suggested in Figure 1: besides neurons placed in the nodes of a directed graph, and drawn as ovals with spikes and rules inside, we also consider square boxes with "arms" touching the synapses; such an arm indicates that the respective synapse is under the control of the astrocyte. (For an easier reference, we can associate labels to astrocytes, but we do not proceed in this way below.)

In Figure 1 we have an astrocyte of degree 3. If two or all three neurons $i_1, j_1, k_1$ send a spike along the synapses $(i_1, i_2), (j_1, j_2), (k_1, k_2)$, respectively,

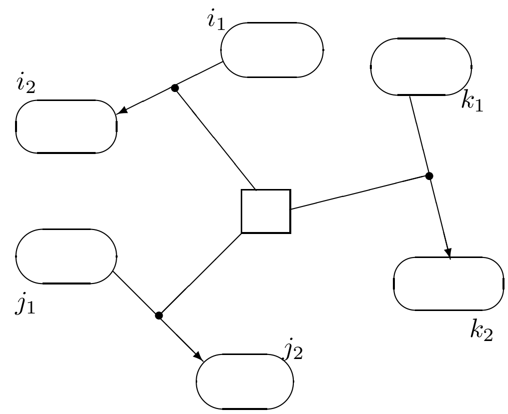


**Figure 1:** An SN P system with astrocytes.

then only one of these spikes is selected and transmitted to the destination and the other two spikes are lost.

Of course, we can also consider a generalization: having astrocytes with different "capacities", not always equal to one. Specifically, we can consider

$$astro \subseteq syn^{\leq k} \times \mathbf{N},$$

with the meaning that for an astrocyte $(x, r)$, if more than $r$ spikes are transmitted along the synapses from $x$, then $r$ of them are non-deterministically selected and let to move, all other spikes are removed. In this paper we do not investigate this case.

## 3 The Undecidability of Deadlock

The previous definition of astrocyte functioning does not capture an important situation, that of *deadlock*. Let us start by examining the case from Figure 2, where we have three astrocytes, all of them of degree 2, simultaneously controlling three synapses.

If all three neurons $i_1, j_1, k_1$ emit spikes, then, although the system is "alive", no continuation is possible: if one of the spikes is selected to survive (assume that this is the case for the spike along the synapse $(i_1, i_2)$), then the two astrocytes which involve the respective synapse (in our case, $(i_1, i_2)(j_1, j_2)$ and $(i_1, i_2)(k_1, k_2)$) will impose that the spikes which had to pass along the remaining synapses (here, $(j_1, j_2)$ and $(k_1, k_2)$) are removed, hence the third astrocyte $((j_1, j_2)(k_1, k_2))$ is not satisfied. (According to the definition in the previous section, if one or more spikes are sent along synapses in an astrocyte, then exactly one spike is selected and let to go.)

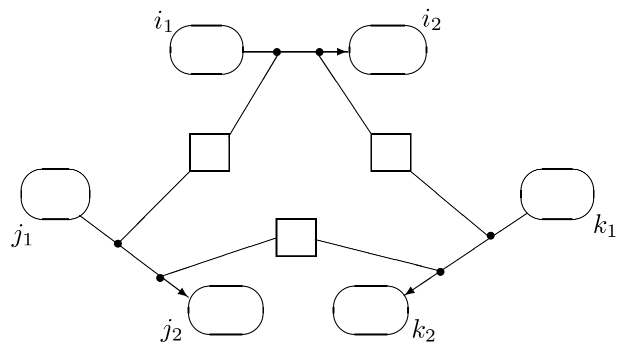


**Figure 2:** A simple case of deadlock.

No continuation is possible (according to the previous definition); we call such a situation *deadlock*. The computation is blocked, it aborts without producing any result.

A similar situation appears in Figure 2 if exactly two neurons spike.

Of course, deciding whether or not a given SN P system enters a deadlock during its computations is an important problem, but, as expected (due to the computational universality of these systems), the problem has a negative answer.

**Theorem 1.** *It is undecidable whether an arbitrary SN P system with (at least three) astrocytes reaches a deadlock.*

*Proof.* Let us consider an arbitrary recursively enumerable set of natural numbers, $Q$ (given, for instance, by means of a register machine which generates its elements). Like in [Ionescu et al. 2006] we construct an SN P system $\Pi$ such that $Q = N_2(\Pi)$. The output neuron of $\Pi$ spikes twice if and only if $Q$ is non-empty, which is undecidable. We construct the SN P systems $\Pi'$ as indicated in Figure 3 – instead of a formal definition of this system we choose to present it in the more suggestive form of a graphical representation.

In alternate time units, neurons $\sigma_2$ and $\sigma_3$ exchange spikes, hence in alternate time units (never at the same time), a spike goes along synapses $(2, 1)$ and $(3, 4)$. Only one spike in the "triangle" of synapses $(2, 1), (3, 4), (5, 6)$ does not lead to deadlock.

If the system $\Pi$ spikes twice (which is equivalent with having the set $Q$ non-empty), then neuron $\sigma_5$ accumulates two spikes and fires, sending a spike along the synapse $(5, 6)$. This spike is simultaneous with one of the spikes passing along synapses $(2, 1), (3, 4)$, hence at that time we have two spikes in this "triangle" of synapses and, like in Figure 2, we have a deadlock.

Thus, the deadlock appears if and only if the set $Q$ is non-empty, and this is an undecidable property. □

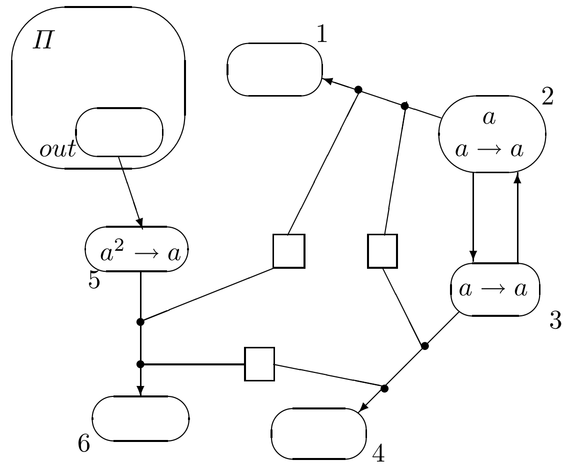


**Figure 3:** The SN P system from the proof of Theorem 1.

Of course, a deadlock can also be interpreted as halting, or we can avoid it by changing the definition, but we do not consider such cases here.

## 4 A Normal Form

Let us consider now the degree of astrocytes. Can it be decreased – if possible, to the minimal value, two? The answer is affirmative.

**Theorem 2.** *Starting from an SN P system $\Pi$ with astrocytes of an arbitrary degree, a system $\Pi'$ can be constructed, with astrocytes of degree 2, such that $N_2(\Pi) = N_2(\Pi')$.*

*Proof.* The idea of the proof is suggested in Figures 4 and 5, where the case of an astrocyte of degree 4 is handled. This astrocyte controls the synapses with *starting neurons* $\sigma_1, \sigma_3, \sigma_5, \sigma_7$ and *target neurons* $\sigma_2, \sigma_4, \sigma_6, \sigma_8$. The subsystem from Figure 4 is replaced with the subsystem from Figure 5.

The astrocyte of degree 4 is replaced by astrocytes of degree 2, controlling synapses between intermediate neurons; for each path from a start neuron to a target neuron we have an astrocyte, hence in total we have as many astrocytes as many pairs of synapses. In the general case, when starting from an astrocyte of degree $n$, we will have $(n-1)n/2$ astrocytes of degree 2, acting in different steps, hence we have $(n-1)n/2$ intermediate steps instead of the unique step of passing from start neurons to target neurons. It is clear that these $(n-1)n/2$ astrocytes behave exactly as the one from Figure 4, leaving only one spike to pass from start neurons to target neurons.

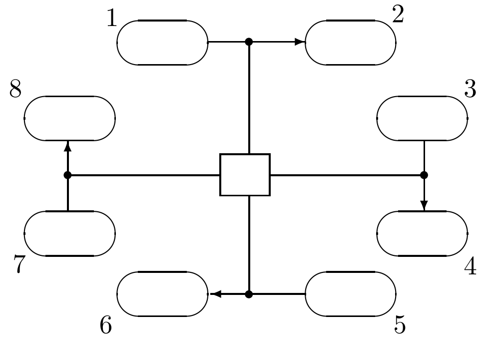


**Figure 4:** An astrocyte of degree four.

Of course, this basic idea raises several problems.

First, after using a rule in the starting neurons, for $(n-1)n/2$ steps these neurons should remain idle, otherwise we have synchronizing problems. This issue is handled as follows. First, if a neuron of $\Pi$ contains initially $n$ spikes, then we introduce $2n+1$ spikes in the corresponding neurons of $\Pi'$. Thus, all neurons contain initially an odd number of spikes. Then, each rule $E/a^c \to a$ from $\Pi$ is replaced in $\Pi'$ by the rule $E'/a^{2c+1} \to a$, where $E'$ denotes a regular expression such that $L(E') = \{a^{2s+1} \mid a^s \in L(E)\}$ (clearly, if $L$ is a regular language, then also $\{a^{2s+1} \mid a^s \in L\}$ is a regular language). This means that after using a rule, the number of spikes remaining in the neuron is even, hence no further rule can be enabled.

From the start neurons only one spike is emitted towards the target neurons. In order not to change the parity of neurons in the target neurons, this spike is replicated before entering the target neuron, by using the *doubling neurons* from Figure 5. This means that from the start neurons to the target neurons we need $(n-1)n/2+2$ steps.

Several different astrocytes can have different degrees. In order to synchronize the system $\Pi'$, we ensure that for all paths from start to target neurons we need the same number of steps, namely the number imposed by the maximum degree of an astrocyte. To this aim, we introduce the *synchronizing neurons*, which only move step by step the spikes towards their destination.

Similar synchronizing (delaying) neurons are introduced also along synapses which do not appear in astrocytes, so that instead of passing in one step from a neuron to another one, we pass in a number of steps which is the same for all pairs of neurons from $\Pi$. Let us denote this number by $\alpha$.

After sequences of $\alpha$ steps, the system $\Pi'$ should be able to use again rules corresponding to rules of $\Pi$. This means that neurons of $\Pi'$ which correspond to neurons of $\Pi$ should contain again odd numbers of spikes. To this aim, we

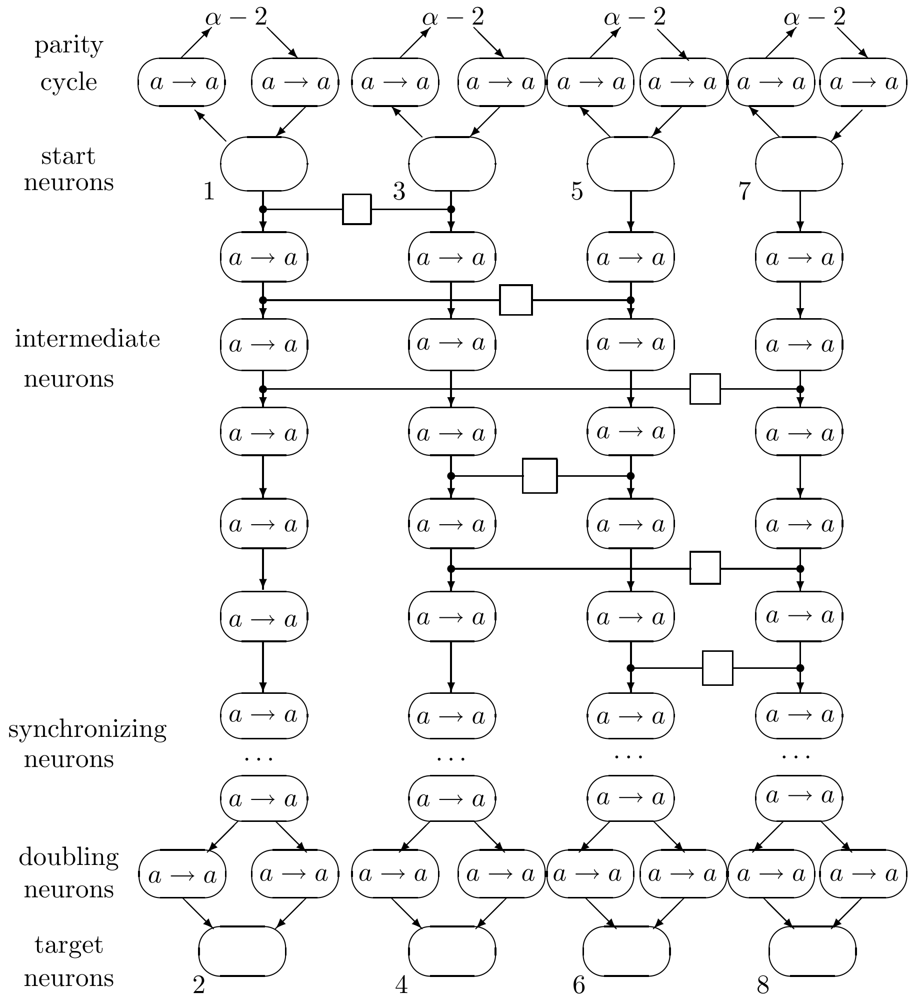


**Figure 5:** The idea of the proof of Theorem 2.

use the *parity cycles* from Figure 5: if a rule is used in a start neuron, hence the parity of its spikes turned from odd to even, then a spike enters the parity cycle and it returns to the start neuron after $\alpha$ steps. In this way, the neuron will again have an odd number of spikes. (Note that if a neuron does not use a rule, its contents remains odd, and no rule can be used before the $\alpha$ steps are performed and possibly new spikes are received form other neurons; because always the spikes come in pairs, due to the doubling neurons, the parity remains odd, hence it is possible now to use rules.)

All new neurons of $\Pi'$ (different from those in $\Pi$) are empty in the beginning and contain the rule $a \to a$. Of course, all these neurons have distinct labels.

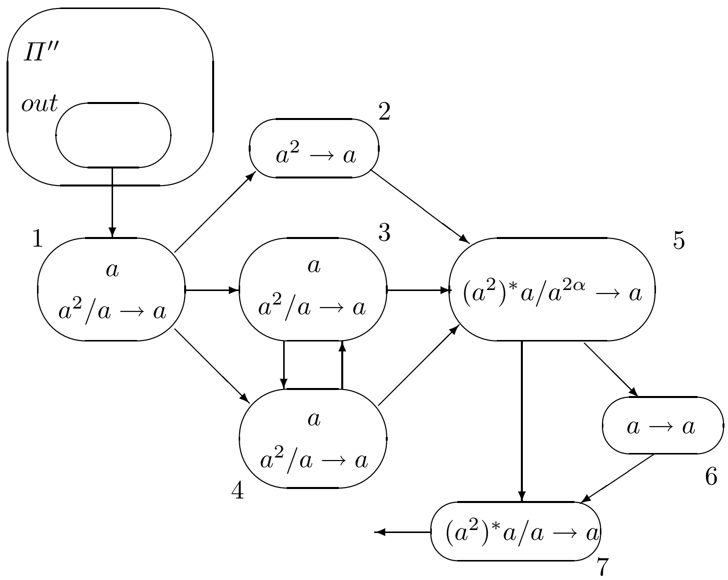


**Figure 6:** The output module of the system $\Pi'$.

It is clear that if the system $\Pi$ spikes at an interval of $n$ steps (thus computing the number $n$), then the system $\Pi'$ spikes at an interval of $n\alpha$ steps. In order to ensure that $\Pi'$ generates the same number as $\Pi$ we proceed as suggested in Figure 6. Specifically, besides the neurons discussed above, indicated in the figure by the subsystem denoted by $\Pi''$, we add the neurons $\sigma_i, 1 \leq i \leq 7$, with $\sigma_7$ being the output neuron of the system $\Pi'$. The regular expression from rules in neurons $\sigma_5$ and $\sigma_7$ describes the set of odd numbers. These new neurons work as follows. Before having a spike sent out from neuron $\sigma_{out}$ of $\Pi''$ (it corresponds to the output neuron of the system $\Pi$), no rule can be used in the new neurons. When a spike arrives in neuron $\sigma_1$, the rule of this neuron can be used. One spike is consumed, hence the rule cannot be used again, and one spike is sent to each of $\sigma_2, \sigma_3, \sigma_4$. Neuron $\sigma_2$ cannot fire with only one spike, but neurons $\sigma_3, \sigma_4$ start to fire. They send to each other one spike, hence their work continues step by step. In each step, these neurons send a pair of spikes to neuron $\sigma_5$, which accumulates these spikes without using them (its rule needs an odd number of spikes in order to fire). When the output neuron of $\Pi$ spikes again (hence after $n\alpha$ steps, for $n \in N_2(\Pi)$), one further spike is received in $\sigma_1$, this neuron spikes again and again neurons $\sigma_2, \sigma_3, \sigma_4$ receive one spike. Neurons $\sigma_3, \sigma_4$ will have now three spikes each, hence from now on they cannot fire. In turn, neuron $\sigma_2$ has now two spikes and fires, sending one spike to neuron $\sigma_5$. In this way, $\sigma_5$ accumulates $2n\alpha + 1$ spikes, and it can start now to fire. The first spike sent to $\sigma_7$ is immediately sent to the environment. In each step, the spike emitted by $\sigma_5$ also reaches $\sigma_6$ and from here it is sent to $\sigma_7$, which thus receives two spikes. Consequently, after the second step, neuron $\sigma_7$ accumulates an even number of spikes and it cannot spike. When the spikes from $\sigma_5$ are exhausted, and only one remains (note that in each step one consumes $2\alpha$ spikes, hence the process lasts $n$ steps), no spike is sent from $\sigma_5$ to $\sigma_7$ (and to $\sigma_6$), hence $\sigma_7$ receives only one spike, from $\sigma_6$. Thus, the number of spikes from $\sigma_7$ is odd again and $\sigma_7$ spikes for the second time. This happens at $n$ steps after the first spike sent out by $\sigma_7$, hence we have the same output as that computed by $\Pi$.

We leave the missing details (for instance, the obvious way to generalize this construction for systems with astrocytes of arbitrary degrees, not only four, as in Figure 4) to the reader. □

The previous construction can be carried out also for SN P systems working in the accepting mode: the only differences is that the input should be provided in such a way to have an odd number of spikes in each neuron, and that the construction from Figure 6 is no longer necessary (the input is accepted if and only if the system halts). The details are left to the reader.

## 5 Astrocytes on Axons

As we have pointed out before, the definition of astrocyte control we work with, following the one from [Binder et al. 2007], is not very close to the biological reality, where the astrocytes sense especially the flow of spikes along axons. This can lead to situations as that from Figure 7, where two cases are represented where the interpretation of the astrocyte control looks strange from a biological point of view (the same spike sent along the unique axon of the neuron has different fates in the synapses).

However, without changing the definition, starting from an SN P system $\Pi$ with astrocyte control (defined as above, for synapses), we can construct an equivalent system $\Pi'$ where the astrocytes act on axons. The idea is to introduce intermediate neurons on each synapse, in such a way to have only one synapse outgoing to each neuron; such a synapse can then be interpreted as an axon. For the cases from Figure 7 the idea is suggested in Figure 8. Of course, in order to preserve the synchronization and the equivalence of the two systems we have to use the techniques from the proof of Theorem 2 (parity cycle, synchronizing neurons, doubling neurons, output module, etc.) – the details are left to the reader.

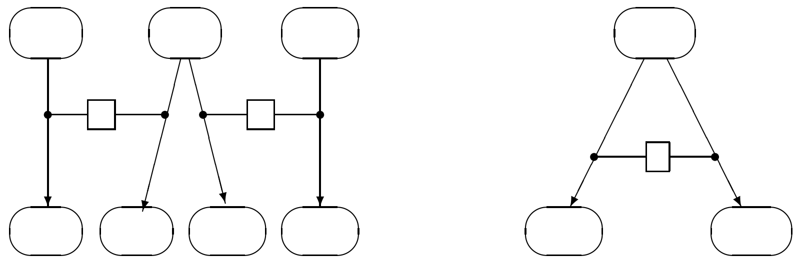

Figure 7: Two situations which show that the above definition should be changed.

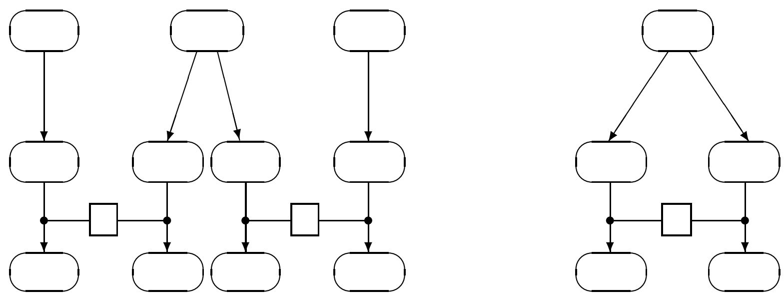

**Figure 8:** Passing to astrocytes controlling axons.

Of course, this only makes the model similar to the reality, but does not solve the question of having a definition which corresponds to reality; this remains as a topic for further research.

## 6 Final Discussion

We have considered here a rather restricted case of the SN P systems with inhibitory and excitatory astrocytes as introduced in [Binder et al. 2007], considering only two questions: the decidability of deadlock and a simplification of astrocytes (to degree 2). There are many issues which remain to be considered, starting with the basic problem of finding a more realistic model, closer to the neuro-biological reality. In particular, the case of astrocytes controlling axons, not synapses should be considered.

Several more technical problems were formulated along the paper and many others are also natural.

For instance, we have considered here SN P systems with standard rules (the spiking rules always produce only one spike); what about using extended rules, i.e., rules of the form $E/a^c \to a^p; d$, producing $p \geq 1$ spikes when fired? Similarly, a further difficulty is introduced by considering systems with delay.

In the previous considerations, the systems were synchronized; how astrocytes should be introduced in asynchronous SN P systems, like those investigated in [Cavaliere et al. 2007]? A similar question appear with respect to SN P systems with an exhaustive use of rules, as in [Ionescu et al. 2007].

## References

[Alhazov et al. 2007] Alhazov, A., Freund, R., Oswald, M., Slavkovik, M.: "Extended variants of spiking neural P systems generating strings and vectors of non-negative integers"; In [Hoogeboom et al. 2007], 123-134.

[Binder et al. 2007] Binder, A., Freund, R., Oswald, M., Vock, L.: "Extended spiking neural P systems with excitatory and inhibitory astrocytes"; In [Gutiérrez-Naranjo et al. 2007], 63-72.

[Cavaliere et al. 2007] Cavaliere, M., Egecioglu, O., Ibarra, O.H., Ionescu, M., Păun, Gh., Woodworth, S.: "Asynchronous spiking neural P systems; decidability and undecidability"; Proc. DNA13, Memphis (2007).

[Chen et al. 2006a] Chen, H., Ionescu, M., Ishdorj, T.-O.: "On the efficiency of spiking neural P systems"; In [Gutiérrez-Naranjo et al. 2006], Vol. I, 195-206, and Proc. 8th Intern. Conf. on Electronics, Information, and Communication, Ulanbator, Mongolia (2006), 49-52.

[Chen et al. 2006b] Chen, H., Ionescu, M., Păun, A., Păun, Gh., Popa, B.: "On trace languages generated by spiking neural P systems"; In [Gutiérrez-Naranjo et al. 2006], Vol. I, 207-224, and Proc. DCFS2006, Las Cruces, NM (2006).

[Chen et al. 2006c] Chen, H., Ishdorj, T.-O., Păun, Gh., Pérez-Jiménez, M.J.: "Spiking neural P systems with extended rules": In [Gutiérrez-Naranjo et al. 2006], Vol. I, 241-265.

[Chen et al. 2006d] Chen, H., Ishdorj, T.-O., Păun, Gh., Pérez-Jiménez, M.J.: "Handling languages with spiking neural P systems with extended rules"; Romanian J. Information Sci. and Technology, 9, 3 (2006), 151-162.

[Chen et al. 2007a] Chen, H., Freund, R., Ionescu, M., Păun, Gh., Pérez-Jiménez, M.J.: "On string languages generated by spiking neural P systems"; Fundamenta Informaticae, 75, 1-4 (2007), 141-162.

[Chen et al. 2007b] Chen, H., Ishdorj, T.-O., Păun, Gh.: "Computing along the axon"; Progress in Natural Computing, 17, 4 (2007), 418-423..

[Gerstner and Kistler 2002] Gerstner, W., Kistler, W.: "Spiking Neuron Models. Single Neurons, Populations, Plasticity"; Cambridge Univ. Press (2002).

[Gutiérrez-Naranjo et al. 2006] Gutiérrez-Naranjo, M.A., et al. (Eds.): Proceedings of Fourth Brainstorming Week on Membrane Computing, Febr. 2006, Fenix Editora, Sevilla (2006).

[Gutiérrez-Naranjo et al. 2007] Gutiérrez-Naranjo, M.A., et al., (Eds.): "Proceedings of Fifth Brainstorming Week on Membrane Computing"; Febr. 2007, Fenix Editora, Sevilla (2007).

[Hoogeboom et al. 2007] Hoogeboom, H.J., Păun, Gh., Rozenberg, G., Salomaa, A. (Eds.): "Membrane Computing, International Workshop, WMC7, Leiden, The Netherlands, 2006, Selected and Invited Papers"; LNCS 4361, Springer, Berlin (2007).

[Ibarra and Woodworth 2006] Ibarra, O.H., Woodworth, S.: "Characterizations of some restricted spiking neural P systems"; In [Hoogeboom et al. 2007], 424-442.

[Ibarra et al. 2006] Ibarra, O.H., Woodworth, S., Yu, F., Păun, A.: "On spiking neural P systems and partially blind counter machines"; Proceedings of Fifth Unconventional Computation Conference, UC2006, York, UK, September 2006, LNCS 4135, Springer (2006), 113-129..

[Ibarra et al. 2007] Ibarra, O.H., Păun, A., Păun, Gh., Rodríguez-Patón, A., Sosik, P., Woodworth, S.: "Normal forms for spiking neural P systems"; Theoretical Computer Sci. 372, 2-3 (2007), 196-217.

[Ionescu et al. 2006] Ionescu, M., Păun, Gh., Yokomori, T.: "Spiking neural P systems"; Fundamenta Informaticae, 71, 2-3 (2006), 279-308.

[Ionescu et al. 2007] Ionescu, M., Păun, Gh., Yokomori, T.: "Spiking neural P systems with exhaustive use of rules"; Intern. J. Unconventional Computing, 3, 2 (2007), 135-154.

[Leporati et al. 2007a] Leporati, A., Zandron, C., Ferretti, C., Mauri, G.: "On the computational power of spiking neural P systems"; [Gutiérrez-Naranjo et al. 2007], 227-246.

[Leporati et al. 2007b] Leporati, A., Zandron, C., Ferretti, C., Mauri, G.: "Solving numerical NP-complete problems with spiking neural P systems"; In Pre-proc. WMC8 (Elefterakis, G., Kefalas, P., Păun, Gh., eds.), Thessaloniki (2007), 405-424.

[Maass and Bishop 1999] Maass, W., Bishop, C., (Eds.): Pulsed Neural Networks, MIT Press (1999).

[Păun 2002] Păun, Gh.: "Membrane Computing. An Introduction"; Springer, Berlin (2002).

[Păun 2007] Păun, Gh.: "Twenty six research topics about spiking neural P systems"; In [Gutiérrez-Naranjo et al. 2007], 263-280.

[Păun and Păun 2006] Păun, A., Păun, Gh.: "Small universal spiking neural P systems"; In [Gutiérrez-Naranjo et al. 2006], Vol. II, 213-234, and BioSystems, in press.

[Păun et al. 2005] Păun, Gh., Pérez-Jiménez, M.J., Rozenberg, G.: "Infinite spike trains in spiking neural P systems"; Submitted 2005.

[Păun et al. 2006] Păun, Gh., Pérez-Jiménez, M.J., Rozenberg, G.: "Spike trains in spiking neural P systems"; Intern. J. Found. Computer Sci., 17, 4 (2006), 975-1002.

[Rozenberg and Salomaa 1997] Rozenberg, G., Salomaa, A. (Eds.): "Handbook of Formal Languages" (3 volumes); Springer-Verlag, Berlin (1997).

[Volterra and Meldolesi 2005] Volterra, A., Meldolesi, J.: "Astrocytes, from brain glue to communication: the revolution continues"; Nat. rev. Neurosci. 6 (2005), 626-640.

[WebPage] The P Systems Web Page: `http://psystems.disco.unimib.it`.