

GENERATION OF CYCLE PICTURE LANGUAGES USING SEQUENTIAL SPIKING NEURAL P SYSTEM

C. Y. PREETHI, HEPZIBAH A. CHRISTINAL¹, S. JEBASINGH, AND A. CHANDY

ABSTRACT. Spiking Neural P Systems (SN P Systems) is a bio-inspired computing model, abstracting the model of brain in processing information using spikes and neurons. The theoretical study of the model has proved that it can compute sets of positive numbers, Boolean functions and string languages. Cycle picture language is a set of pictures obtained using cycle grammar and chain code representation. In this paper we aim to compute the cycle picture languages using a variant of SN P system namely, Sequential SN P System using neurons and spiking rules. We compute the cycle picture language of sequence of chains.

1. INTRODUCTION

Membrane computing is a new branch of natural computing which was initiated by Gheorghe Paun in 1998 [7]. The principle of membrane computing is to abstract computing ideas from the structure and the functioning of a single cell and from complexes of cells, such as tissues and organs including the brain. The models obtained are distributed and parallel computing devices, usually called P systems. In the literature we find that there are three main classes of P systems investigated: cell-like P systems, tissue-like P systems and neural-like P systems. Spiking Neural P system (SN P System) is the special case of neural-like P system, where the computations are inspired by the neuro physiological

¹*corresponding author*

Key words and phrases. Membrane computing, Chain Code Picture, Cycle pictures, SN P System.

behaviour that sends electric impulses alongside axon to other neurons. SN P systems are proved to generate string languages and functions [3], [8], [6].

Picture languages are generalization of string languages. In the literature we find various methods to generate picture languages using grammars [2]. A string over the alphabet denotes a picture if the alphabets describes a unit line move in the two dimensional plane by walking in the direction of north, east, west and south respectively. Such a word is called a chain code, introduced by Freeman [1]. A set of pictures described using a chain code is called a chain code picture language. In [5], the authors have studied the generation of space filling curves using P system with parallel rewriting. In [4], the authors have studied the generation of cycle picture languages using cycle rewriting chain code P system. The aim of the present study is to compute cycle picture languages using Cell-like SN P System in sequential model and study the generative power in terms of the number of neurons.

In section 2, we present the preliminaries of chain code pictures, cycle picture languages and Cell-like SN P System. In section 3, we present a Cell-like SN P System to construct the sequence of even number of chain in sequential mode. The string language generated by the system is mapped to the picture description alphabets through the homomorphism. The picture words obtained through homomorphism describe the required chain code cycle picture language.

2. PRELIMINARIES

Chain codes is one of the approaches for solving the problem in picture processing. It gives the connection between pictures and strings describing pictures [1]. A string over the alphabet $\Sigma = \{n, e, w, s\}$ denotes the picture if the alphabets describes a unit line move in the two dimensional plane by walking in the direction of north, east, west and south respectively. This type of word was introduced by Freeman. A set of pictures described using a chain code is called chain code picture language. A picture word describes a cycle if it is a closed curve. A picture word denotes elementary cycle if it is a cycle and no proper sub word of w forms a cycle. A set of pictures is called picture language and a set L of picture words is called a picture description language.

A cell-like SN P system of degree $m \geq 1$, is a construct of the form

$$\prod = \{O, \mu, n_1, \dots, n_m, R_1, \dots, R_m, i_o\},$$

Where:

- $O = \{a\}$ is the singleton alphabet(the object a is called spike);
- μ is a hierarchical structure of neurons;
- $n_i, 1 \leq i \leq m$, is the number of spikes initially present in the compartment i of μ ;
- $R_i, 1 \leq i \leq m$, is the finite set of rules from the membrane i , of the following two forms:
 - (a). $E/a^c \rightarrow \mu$, where E is a regular expression over O , $c \geq 1$, and μ is a sequence of pairs of the form (a^p, tar) , $p \geq 1$, $tar \in \{here, out, in, in_{all}\} \cup \{in_j | 1 \leq j \leq m\}$;
 - (b) $a^s \rightarrow \lambda$, $forsomes \geq 1$, with the restriction that $a^s \notin L(E)$ for any rule $E/a^c \rightarrow u$ of type (1) from R_i ;
- $i_o \in \{0, 1, \dots, m\}$ indicates the output region of $\prod$.

The rule $E/a^c \rightarrow u$ is called firing rule , where $\mu = (a_1^p, tar_1) \dots (a_k^p, tar_k)$ with $P_i \geq 1$ and $tar_i \in \{here, out, in, in_{all}\} \cup \{in(j) | 1 \leq j \leq m\}, 1 \leq i \leq k$. The rule is applied as follows: if compartment $i \in 1, 2, \dots, m$ where the rule resides contains k spikes and $a^k \in L(E)$, $k \geq c$, then the rule is enabled and it can be applied. Using such a rule means consuming c spikes and producing $p_1 + \dots + p_k$ spikes of u : the target indications $here, out, in, in_{all}$,means that the spikes with target $here$ remain same region where the rule is applied, the spikes with target out are moved to the region immediately outside membrane i , while the spikes with the target in are sent to one of the immediately inner membranes, it moves sequentially of all the rule used. If a rule $E/a^c \rightarrow \mu$ has $E = a^c$, then we will write it in the simplified form $a^c \rightarrow \mu$.

The rules of the form $a^{(s)} \rightarrow \lambda$ are forgetting rules. If compartment i contains exactly s spikes, then the rules can be applied and it removes s spikes immediately from the compartment. Thus the rules are used in sequential manner in each compartment. The computation in Cell-like SN P system is described by the number of spikes present in each compartment. Thus, the initial configuration is $C_0 = \langle n_1, n_2, \dots, n_m \rangle$. Using the rules as described above. A transition between two configurations C_1 and C_2 is denoted by $C_1 \rightarrow C_2$. Any sequence of transitions starting from the initial configuration is called a

computation. A computation halts if it reaches a configuration when no rule is used. The language generated by cell like SNP system is denoted by $L(\pi) = \{(a^2)^k(aa^3a^2)^k/k \geq 1\}$. $LSNP_m\{forg, here, in, in_{all}\}$ denotes the family of all set of languages generated by the system with at most m membranes, using forgetting rules and target indications of the types given.

The output string language $L(\pi)$ obtained from the SNP system is mapped with the chain code alphabets $\Sigma = \{n, e, w, s\}$ by a homomorphism $\phi : L(\pi) \rightarrow \Sigma$. The chain code interpretation of the string language $\phi(L(\pi))$ gives the desired cycle picture language representation.

3. GENERATION OF SEQUENCE OF EVEN NUMBER OF CHAINS USING CELL-LIKE SNP SYSTEM

In the following we present a Cell-Like SNP system which generates cycle picture language, sequence of even number of chains, in sequential mode.

$\Pi = (O, \mu, n_0, n_1, n_2, R_0, R_1, R_2, i_0)$ Where

- $O = \{a\}$ is a spike
- $\mu = [[]_1 []_2]_0$
- $n_0 = \{\phi\}; n_1 = \{a^2\}; n_2 = \{\phi\}$
- Rules are as follows:
 - $R_0 = \begin{cases} a(a^{2k})/a^{2k} \rightarrow (a^{2k}, out)(a^{6k+1}, in_2) \\ a^{6k+1}/a \rightarrow (a, out) \\ a^{6k}/a^3 \rightarrow (a^3, out) \\ a^{6k-3}/a^2 \rightarrow (a^2, out) \end{cases}$
 - $R_1 = \begin{cases} (a^2)^k/a^2 \rightarrow (a^2, out)(a^2, here) \\ a^s \rightarrow \lambda, s = 2k - 1 \\ a \rightarrow (a, out) \end{cases}$
 - $R_2 = \begin{cases} a^{6k+1}/a \rightarrow (a, out) \\ a^{6k}/a^3 \rightarrow (a^3, out) \\ a^{6k-3}/a^2 \rightarrow (a^2, out) \end{cases}$
- $i_0 = []_0$

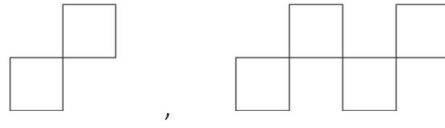


FIGURE 1. Sequence of even number of Chains

3.1. Overview of the computation. The computation of the system is sequential at the level of membrane and also with respect to the implementation of the firing and forgetting rules present in the region. The system has 3 membranes n_0, n_1 and n_2 each with firing rules and forgetting rules defined in the set R_0, R_1 and R_2 . Initially the membrane region n_1 has 2 spikes and the other regions do not have any spikes. The region n_1 with spike is activated and the rules from the set R_1 are applied non-deterministically to the spikes.

The neuron n_1 can either apply the firing rule or the forgetting rule non-deterministically. Suppose that the system non-deterministically selects the forgetting rule and applies to the spike a^2 then one spike will be lost and another spike will remain in the system. In the next step the neuron will apply the firing rule $a \rightarrow (a, out)$ and send the remaining one spike to the neuron n_0 . Now the neuron n_0 will be activated and n_1 will stop applying the rules. But in n_0 , no firing rule is applicable and computation stops without emitting any spikes.

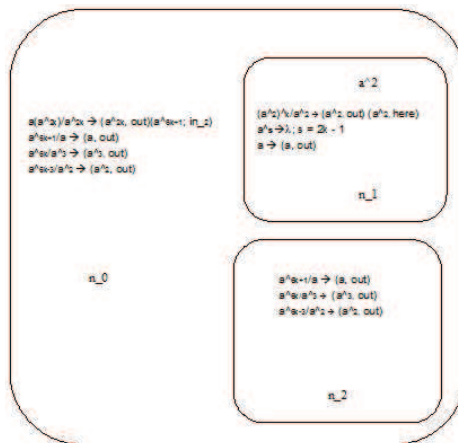


FIGURE 2. The cell-like membrane structure with firing and forgetting rules

Assume that the system non-deterministically selects the firing rule

$$(a^2)^k / a^2 \rightarrow (a^2, out)(a^2, here)$$

and applies to the spike a^2 . In that case it consumes a^2 spike and sends a^2 spike to n_0 , produces a^2 spike in n_1 . This process of sending a^2 spike to n_0 and producing a^2 spike in n_1 is repeated till the system chooses to apply the forgetting rule. On repeating the firing rule k times, n_0 and n_1 will have $(a^2)^k$ spikes. When the system apply the forgetting rule $a^s \rightarrow \lambda, s = 2k - 1$, the a^{2k-1} will be forgotten from n_1 and the region will have only one spike remaining. This result in the application of the rule $a \rightarrow (a, out)$ and the a spike is consumed and one spike is emitted to n_0 . After $k + 1$ steps the neuron n_1 will not have any spikes and n_0 will have $a(a^2)^k$ spikes, which triggers the neuron n_0 to apply the rules.

In n_0 the presence of $a(a^2)^k$ spikes makes the system to non-deterministically apply the rule $a(a^{2k})/a^{2k} \rightarrow (a^{2k}, out)(a^{6k+1}, in_2)$. The application of the rule consumes $(a^2)^k$ spikes, emits $(a^2)^k$ spikes to the environment produces a^{6k+1} spikes in to the region n_2 . The $(a^2)^k$ spikes emitted from n_0 forms part of the output string of the computation. Now the neuron n_0 has only spike and n_2 has a^{6k+1} spikes. The presence of a^{6k+1} spikes activates the neuron n_2 to apply the firing rules.

In n_2 the presence of a^{6k+1} spikes makes the system to non-deterministically apply the rule $a^{6k+1}/a \rightarrow (a, out)$. The application of the rule consumes a spike, emits a spike to the region n_0 and the region n_2 has spikes. The presence of a^{6k} spikes makes the system to non-deterministically apply the rule $a^{6k}/a^3 \rightarrow (a^3, out)$, resulting in the consumption of a^3 spikes and emitting a^3 spikes to the region n_0 and the region n_2 has a^{6k-3} spikes remaining. The presence of a^{6k-3} spikes makes the system to non-deterministically apply the rule $a^{6k-3}/a^2 \rightarrow (a^2, out)$, resulting in the consumption of a^2 spikes and emitting a^2 spikes to the region n_0 and the region n_2 has a^{6k-5} spikes remaining. This process will be repeated until the region n_2 has only one spike and the region n_0 will have a^{6k+1} spikes. The presence of a^{6k+1} spikes activates the neuron n_0 again to apply the firing rules.

In n_0 the presence of a^{6k+1} spikes makes the system to non-deterministically apply the rule $a^{6k}/a^3 \rightarrow (a^3, out)$, resulting in the consumption of a spike and

emitting one spike to the environment and the region n_0 has a^{6k} spikes remaining. The single spike emitted from n_0 is placed along with the output string of the computation $(a^2)^k a$.

Now the region n_0 has a^{6k} spikes which makes the system to non-deterministically apply the rule $a^{6k}/a^3 \rightarrow (a^3, out)$, resulting in the consumption of a^3 spikes and emitting a^3 spike to the environment and the region n_0 has a^{6k-3} spikes remaining. The a^3 spikes emitted from n_0 are placed along with the output string of the computation $(a^2)^k aa^3$.

Now the region n_0 has a^{6k-3} spikes which makes the system to non-deterministically apply the rule $a^{6k-3}/a^2 \rightarrow (a^2, out)$, resulting in the consumption of a^2 spikes and emitting a^2 spikes to the environment and the region n_0 has a^{6k-5} spikes remaining. The a^2 spikes emitted from n_0 are placed along with the output string of the computation $(a^2)^k aa^3 a^2$. This process is repeated until the region is left with 2 spikes and no firing rule is applicable and computation stops. At that time the sequence of spikes emitted from the output neuron n_0 will be in the form $(a^2)^k (aa^3 a^2)^k, k \geq 1$.

Finally we define a homomorphism ϕ between the output string language $(a^2)^k (aa^3 a^2)^k, k \geq 1$ to the set $\Sigma = \{n, e, w, s\}$ to obtain a chain code interpretation of the output. Let $\phi(a^{2k}) = e^{2k}$, $\phi(a) = n$, $\phi(a^3) = ws^2$, $\phi(a^2) = wn$. Hence the string language obtained from the sequential cell-like SN P system is given by $L(\Pi) = \{e^{2k}(nws^2wn)^k, k \geq 1\}$. The chain code interpretation of the string language gives the cycle picture language namely, sequence of even number of chains.

4. CONCLUSION

In this paper, we have studied the generation of sequence of even number of chains using the bio-inspired computational model, sequential cell-like spiking neural P system with 3 neurons. The study links SN P system with cycle picture languages and indicates that SN P system is a powerful mathematical model to study picture languages, fractals and other space filling curves, which has not been previously reported in the literature. The study gives the scope that the model need to be further investigated in the generation of cycle picture languages such as kites, Von Koch quadric 8 segment-like curves and space filling curves.

REFERENCES

- [1] F. HERBERT: *Computer processing of line-drawing images*, ACM Computing Surveys (CSUR), **6**(1) (1974), 57-97.
- [2] G. SIROMONEY, R. SIROMONEY: *Rosenfeld's cycle grammars and kolam*, International Workshop on Graph Grammars and Their Application to Computer Science. Springer, Berlin, Heidelberg, 1986.
- [3] M. IONESCU, P. GHEORGHE, Y. TAKASHI: *Spiking neural P systems*, Fundamenta informaticae, **71**(2-3) (2006), 279-308.
- [4] S. JEBASINGH, G. JOHNSY, P. S. DIVYA: *P system using cycle grammar*, International journal of Recent Technology in Engineering, **8**(2) (2019), 327-330.
- [5] K. G. SUBRAMANIAN, I. VENKAT, L. PAN: *P systems generating chain code picture languages*, Proceedings of Asian Conference on Membrane Computing, 2012.
- [6] L. PAN, P. GHEORGHE, M. J. PEREZ-JIMENEZ: *Spiking neural P systems with neuron division and budding*, Science China Information Sciences, **54**(8) (2011), 1596.
- [7] P. GHEORGHE: *Computing with membranes*, Journal of Computer and System Sciences, **61**(1) (2000), 108-143.
- [8] T. SONG, L. PAN: *Spiking neural P systems with request rules*, Neurocomputing, **193** (2016), 193-200.

DEPARTMENT OF MATHEMATICS
KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES
COIMBATORE, TAMIL NADU
E-mail address: preeti.ceon@gmail.com

DEPARTMENT OF MATHEMATICS
KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES
COIMBATORE, TAMIL NADU
E-mail address: christyhep@gmail.com

DEPARTMENT OF MATHEMATICS
KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES
COIMBATORE, TAMIL NADU
E-mail address: jebasinghs@karunya.edu

DEPARTMENT OF ELECTRONICS AND COMMUNICATION
KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES
COIMBATORE, TAMIL NADU
E-mail address: abrahamdchandy@gmail.com