Journal of Computer and Mathematical Sciences, *Vol.10(3),454-460 March 2019* *ISSN 0976-5727 (Print)*
(An International Research Journal), *www.compmath-journal.org* *ISSN 2319-8133 (Online)*

Linear Equi-triangular Array P System

A. Dharani[1,*] and R. Stella Maragatham[2]

[1,2]Department of Mathematics,
Queen Mary's College, Chennai 600 004 INDIA.
email: dharaniclement@gmail.com, rstellamar@yahoo.com * Research Scholar.

(Received on: March 3, 2019)

ABSTRACT

Motivated by the studies of grammatical formalism for picture generation in two dimensional plane using polygons like triangle, square and hexagon, we have considered equilateral triangles. In this paper we propose a new kind of P system called linear equi-triangular array P system. We show that linear equi-triangular array P system with two membranes has more generative power than linear equi-triangular array P system with one membrane. We also introduce parallel linear equi-triangular array P system and prove that parallel linear equi-triangular array P system with two membranes is properly included in the family of linear equi-triangular array P system with six membranes.

Keywords: Linear array, equilateral triangle, membrane computing.

1. INTRODUCTION

In the theory of formal languages various syntactic techniques have been developed by researches to generate picture languages in two dimensional plane. Polygons like triangle, square and hexagon have been used widely. Two variants of equi-triangular array grammar has been recently introduced. Motivated by the structure of membranes and the working of living cells, P system in the area of membrane computing was introduced by Paun (2000)[5,6]. The idea of P system and grammars for the generation of arrays[1] and picture languages[7] have been studied.

In this paper we have studied the concepts of P system combined with one of the variants of equi-triangular array grammar, namely, linear equi-triangular array grammar.

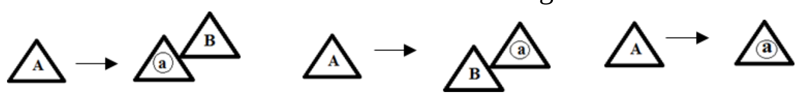
2. PRELIMINARIES

Definition 2.1

(a) Et-tile: An equilateral triangle with unit side length is named as et-tile.
(b) Equi-triangular array: An equi-triangular array is a portion in the two dimensional plane made of et-tiles joined together by a vertex and the mid-point of the edge.

Definition 2.2

A linear equi-triangular array grammar (LINet-AG) is a structure G = (N,T,P,S) where N is a finite nonempty set of et-tiles labelled by nonterminal symbols, T is a finite nonempty set of et-tiles labelled by terminal symbols and N∩T = ∅. An et-tile in N is labelled by the start symbol S. The set P consists of linear rules of the following forms:



where **A** is a nonterminal symbol and ***a*** is a terminal symbol. Note that the first and the third rules given above are in fact the rules of a regular et-array grammar. In the right side of the rules, vertex to edge catenation of the et-tiles is done. Similar rules corresponding to each side of the et-tile labelled A can be given. In an application of a rule of any of the forms mentioned above, the et-tile with the circled label replaces the et-tile with the label A. For convenience, sometimes we mention only the label of an et-tile instead of giving it as an equilateral triangle with a label. Also the use of the circle surrounding a label is only for the purpose of indicating how the rule is applied and is not retained after the application of the rule.

The linear et-array language of G, denoted by L(G) , consists of et-arrays such that all the et-tiles are labelled by terminal symbols. The family of linear et-array language is denoted by $\mathcal{L}$(et-LIN).

3. LINEAR EQUI-TRIANGULAR ARRAY P SYSTEM

Definition 3.1

A linear equi-triangular array P system of degree $m \geq 1$, is defined as $\Pi = (V, T, \mu, F_1, F_2,\ldots, F_m, R_1, R_2,\ldots, R_m, i_0)$, where V is the finite set of et-tiles, $T \subset V$ is the finite set of et-tiles, called terminal et-tiles, μ is a membrane structure with *m* labelled regions, each having a distinct label taken from the set $\{1, \ldots, m\}$; $F_1, F_2,\ldots, F_m$ are finite sets of et- arrays over V , with F_i in the region *i* of μ; $R_1, R_2,\ldots, R_m$ are finite sets of linear equi-triangular array grammar rules, with R_i in the region *i* of the membrane structure μ. The rules have target indications $tar \in \{here, out, in\}$. If the target is *here*, then the evolved et-array stays in the same region; if the target is *in* then the evolved et-array enters the immediate inner region; if the target is *out*, the evolved et-array enters the immediate outer region; i_0 is the label of output region of μ. In a region, at a time only one rule, which is applicable to an et-array in that region, is applied. A computation in the system is said to be successful if it halts when no rule can be applied to the objects in the regions and the generated picture of an et-array reaches the output membrane and is collected in the language of the system Π, denoted by et-*AL*(Π).

We consider the evolution rules to be linear et-array grammar kind of rules mentioned in definition 2.2.

The family of all et-array languages generated by linear equi-triangular P systems of degree at most m, is denoted by $etAP_m$ (LINet-AG).

We illustrate with an example. Here again, for convenience, we mention only the labels of et-tiles.

Example 3.2

Consider a linear equi-triangular array P system Π_1, having 6 membranes and rules of linear et-array grammar kind of rules given by Π_1= ({A_1,A_2,A_3,A_4,A_5,A_6}, {a}, μ, F_1,F_2, F_3, F_4, F_5 , F_6,R_1, R_2,R_3, R_4,R_5, R_6,6) where the membrane structure is μ=$[_1[_2]_2[_3]_3[_4]_4[_5]_5[_6]_6]_1$. F_1 consists of et-array initially present in region 1 as in Figure 3.3(A) and F_2, F_3, F_4, F_5 and F_6 are all empty indicating that they have no initial et-arrays. The region 6 is the output region.

The set of rules are given by

R_1 = {(r_{11},in_2)(r_{12},in_2)} ; R_2 = { (r_{21},in_3)(r_{22},in_3)};R_3 = { (r_{31},in_4)(r_{32}, in_4)}

R_4 = {(r_{41}, in_5)(r_{42}, in_5)}; R_5 = {(r_{51}, in_6)(r_{52}, in_6)}; R_6 = { (r_{61},out_1)(r_{62},$here_6$)}

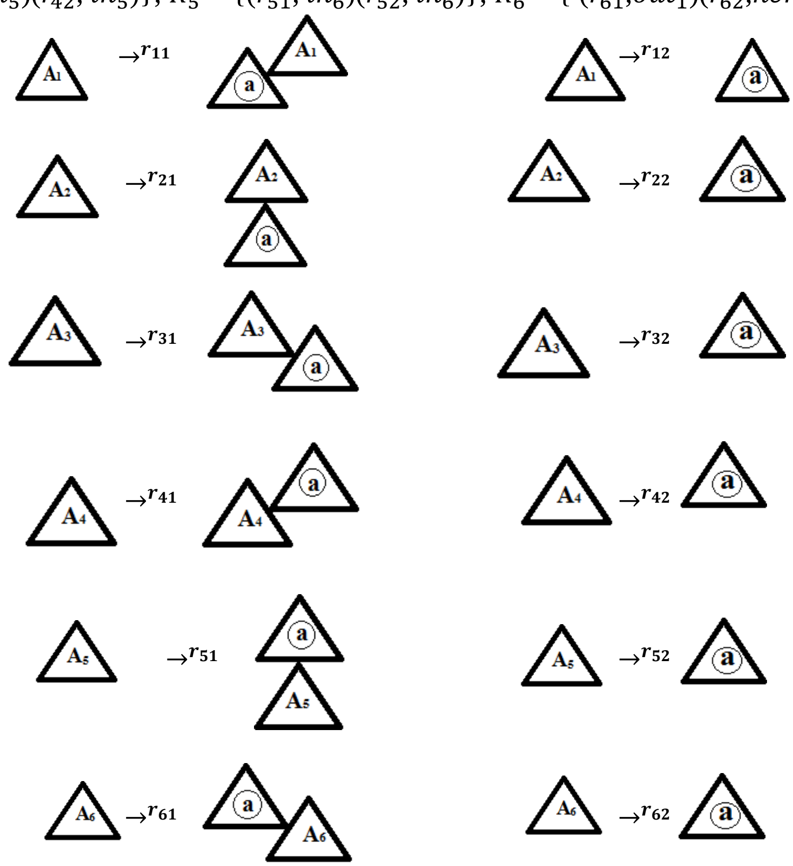
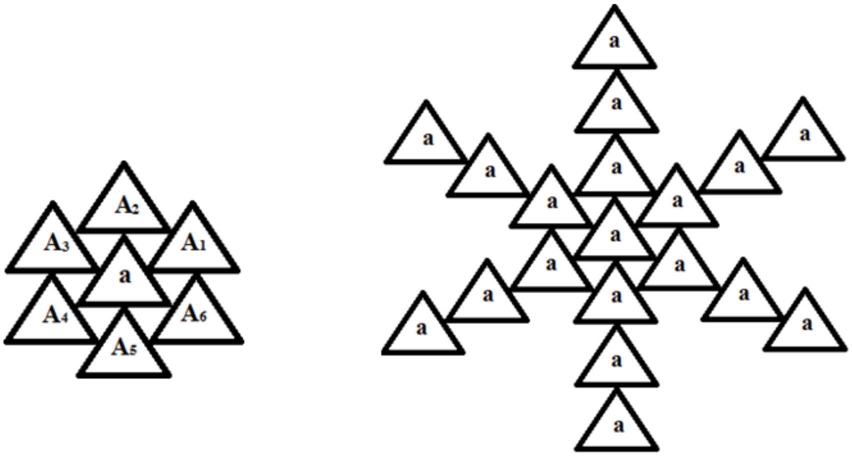


Figure 1



(A) (B)

Figure 2

The derivation starts with the initial array F_1. The generation of the picture is carried on by applying the rules r_{11} ,r_{21},r_{31},r_{41},r_{51},r_{61} and the termination of the picture is done using the rules r_{12} ,r_{22},r_{32},r_{42},r_{52},r_{62}. The generated language consists of et-arrays in the shape of a star. Thus this language is in the family $etAP_6$ (LINet-AG).

Theorem 3.3

$etAP_1$ (LINet-AG) $\subset$ $etAP_2$ (LINet-AG)

Proof

For proving proper inclusion, we consider a linear equi-triangular array P system Π_2, given by Π_2=({S,A,B},{a,b},μ,F_1,F_2,R_1,R_2,2) having rules that are linear et-array grammar kind of rules, where the membrane structure μ=$[_1[_2]_2]_1$ consists of two membranes with membrane labelled 2 inside the membrane labelled 1. F_1 consists of an et-array initially present in region 1 as in Figure 3.4(A) and F_2 is empty indicating that the region with label 2 has no initial et-array in it. The region 2 is the output region. The rule sets are given by:

R_1= ((r1, in) (r3, in)) R_2= ((r2, out) (r4, here))

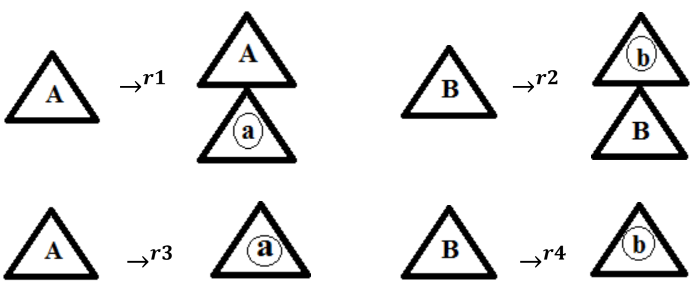
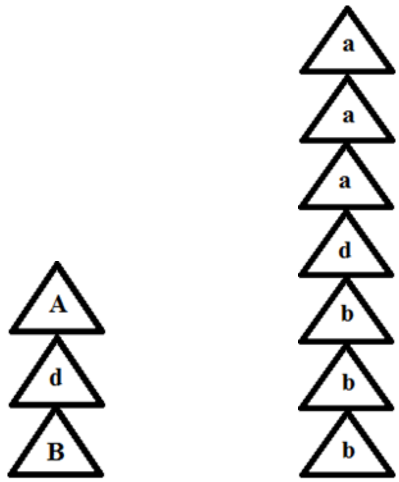


Figure 3



(A) (B)

Figure 4

The computation starts with the initial et-array in region 1.The application of the rule r1 will generate et-array in upper part. Then the et-array moves to the region 2 as the target indication is '*in'*. In region 2, the rule r2 is applied and the evolved et-array is sent to region 1 as the target indication is '*out'*. The process is repeated using the rules r1 and r2. In region 1 the growth is terminated using rule r3 and in region 2 the growth is terminated using rule r4. The terminated et-arrays are collected in the output region 2.

The et-arrays are generated in such a way that there are equal number of tiles to the upper and lower part of the tile 'd' as shown in Figure 4 (B). The generated language is in the family $etAP_2$ (LINet-AG). But this language does not belong to $etAP_1$ (LINet-AG), since the pattern of equal growth of et-arrays above and below the tile 'd', cannot be maintained.

Corollary 3.4

$etAP_2$ (LINet-AG) – $\mathcal{L}$ (et-Lin) $\neq \phi$ where ϕ is the empty set.

Proof

The result follows from the construction of the proof of theorem 3.3 because the et-array shown in Figure 4(B) which involves the number of triangles labelled 'a' and the number of triangles labelled 'b' are equal in number. But this feature cannot be handled by linear equi-triangular array grammar rules.

4. PARALLEL LINEAR EQUI-TRIANGULAR ARRAY P SYSTEM

Definition 4.1

A parallel linear et-array P system is a construct

$\Pi = (V, T, \mu, F_1, F_2, \ldots, F_m, R_1, R_2, \ldots, R_m, i_0)$

where every term in the construct is the same as linear et-array P system. The rules applied are linear et-array grammar kind of rules. The rules having the same target indication, '**here**', '**in**' or '**out**' are applied in a parallel manner. In a region, if the set of rules does not have the same target indications then the computation does not take place and the et-array remains in the same region.

The family of all array languages generated by a parallel linear et-array P system is denoted by $PetAP_m$ (LINet-AG).

Theorem 4.2

There exists an equi-triangular array language which is generated by a linear et-array P system involving six membranes while it is generated in a parallel linear array P system with only two membranes.

Proof

In example 3.2 we have constructed a linear equi-triangular array p system with 6 membranes generating a language consisting of et-arrays in the shape of a star with six arms as shown in figure 2(B).

To construct a parallel linear equi-triangular array P system Π_2= ({A_1,A_2,A_3,A_4,A_5,A_6}, {a}, μ, F_1,F_2, R_1 ,2) where the rules are ri1,ri2 where i= 1 to 6 , and F_1 are same as in example 3.2. F_2 is empty indicating the fact that there is no initial et-array in region 2. Membrane 2 is the output membrane.

R_1 = {(r_{11},here) (r_{21},here) (r_{31},here) (r_{41},here) (r_{51},here) (r_{61},here) (r_{12},in) (r_{22},in) (r_{32},in) (r_{42},in) (r_{52},in) (r_{62},in)}

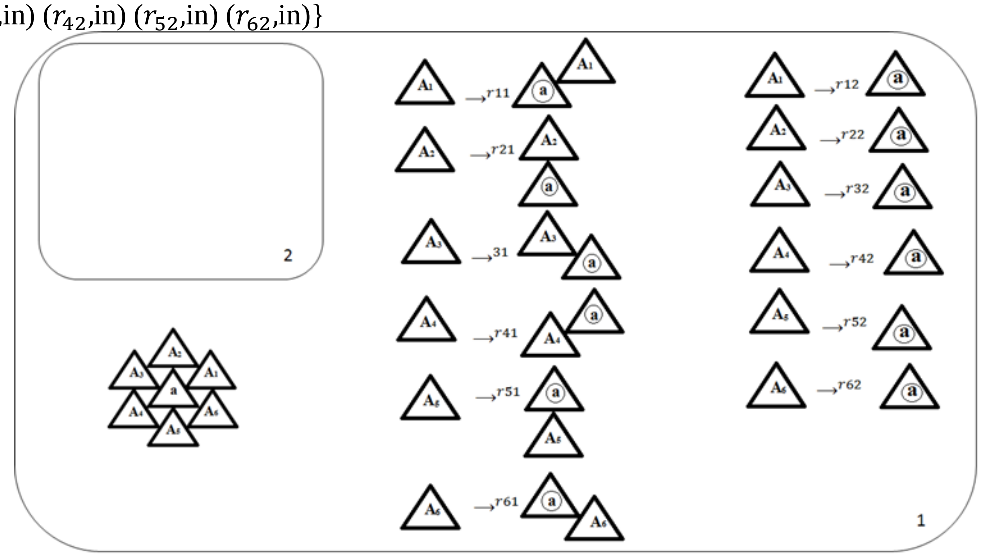


Figure 5

For the rules $r_{11}, r_{21}, r_{31}, r_{41}, r_{51}, r_{61}$, when the target indication is 'here', the initial et-array grows by one tile in all the six directions as rules applied are parallel in nature. The process of applying the non-terminal rules with target indication 'here' is repeated to grow the star shaped et-array to bigger size. We apply the rules $r_{12}, r_{22}, r_{32}, r_{42}, r_{52}, r_{62}$ with target indication 'in' to terminate the et-array and it is sent to region 2 in which the output is collected.

Hence the language generated by this grammar $PetAP_2$ (LINet-AG) consists of et-arrays in the shape of a star with six arms.

Thus we have proved that there exists an equi-triangular array language which is generated by a linear et-array P system involving six membranes could also be generated by a parallel linear et-array P system with only two membranes.

5. CONCLUSION

We have introduced a new model named linear equi-triangular array P system and studied its generative power. The study of P system for one more variant of equi-triangular array grammar named basic puzzle equi-triangular array P system is for future work.

REFERENCES

1. R. Ceterchi, M. Mutyam, Gh. Paun , K.G. Subramanian, Array-Rewriting P system, *Natural Computing*, 2, 229-249 (2003)
2. D. Giammarresi, A. Restivo: Two-dimensional languages. In: Rozenberg G., Salomaa, *A.(eds.) Handbook of Formal Languages*, Springer, Heidelberg, vol. 3, 215-267, (1997)
3. P.T. Hadingham, Formal systems in Artificial Intelligence: an illustration using semigroup, automata and language theory, *Artificial Intelligence Review*, 4, 3-19, (1990).
4. P. Isawasan, A. Dharani, R. Stella Maragatham, K.G. Subramanian, Extensions of Equi-Triangular Regular Array Grammars, Proceedings of 8th IEEE International Conference on Control Systems, *Computing and Engineering (ICCSCE2018)*, Penang, Malaysia, 115-118,(2018).
5. Gh. Paun, Membrane Computing: An Introduction. Springer-Verlag Berlin, Heidelbrg, (2000).
6. Gh. Paun, Computing with Membranes, *J. Comp. System Sci*., 61, 108 – 143, (2000).
7. K. G. Subramanian, P Systems and Picture Languages, *Lecture Notes in Computer Science*, 4664, 99-109 (2007).