ISO-ARRAYS AND CONDITIONAL COMMUNICATION ON P SYSTEM

K. Bhuvaneswari*
Department of Mathematics
Sathyabama Institute of Science and Teachnology
Chennai - 600 119
bhuvanamageshwaran@gmail.com

T. Kalyani
Department of Mathematics
St. Joseph's Institute of Technology
Chennai - 600 119
kalphd02@yahoo.com

Abstract. Among the theoretical applications of formal language theory, membrane computing is a new theoretical model. It is a cell-like structure in which regions are separated by membranes. It is introduced by G.H. Paun, called P system. This computational system can be used for generating string languages, two dimensional picture languages, tiling patterns and tessellations. In this paper we introduced triangular tile pasting P system with conditional communication and explained with an example. Comparison result between triangular tile pasting system and triangular tile pasting P system has been given. The results on generating powers of iso-array grammars and triangular tile pasting P system are also examined in this paper.
Keywords: iso-triangular tiles, tile pasting P System, iso-array grammars, permitting and forbidden conditions.

1. Introduction

Membrane computing is a new theoretical model to computer researchers and non-computer researches. G.H. Paun introduced a new computing model called P system in [7]. It has a membrane structure. In this structure membranes are working as living cells in biological structure model. P system is a distributed and parallel computing model. P system has merely a biological background and mathematical formalism. The evolution rules and evolving objects are presented in the regions of membranes. The rules are applied non-deterministically and the computation starts by applying the rules to the tiles presented in the region in parallel mechanism. P system with multi rewriting rules, tissue like P system and neural like P system are some types of P system. Square tiles [11] are considered in the tile pasting P system to generate some two dimensional picture languages. And the computational power of this P system is examined with the

∗. Corresponding author

computational powers of other existing P systems. By relating P system and array grammars, Ceterchi et al. [5] began a study on linking these two areas, which were not discussed in early research. Connecting P system and array re-writing grammars, array-rewriting P system is introduced to generate two dimensional picture languages. Iso-triangular tile pasting system is introduced by Kalyani et.al. [9] which is an interesting tile pasting system, generating iso-picture patterns and iso-picture languages. Connecting P system and iso-triangular tile pasting system, Triangular tile pasting P system is defined in [1].

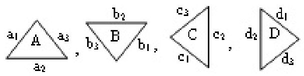
In recent years many different types of P system have been introduced, among the variants of P system, conditional communication [8] is a new study, in which the communication to the membranes are controlled by the tiles or tiling not by the pasting rules. To control the communication, permitting and forbidden conditions are defined in the tiles or sub tiles which exits in a given picture. Tissue-like P system is a type of P system to generate two dimensional pictures. In this P system membranes are considered as elementary membranes and it is differentiated from usual P system in such a way that it has no skin membranes and environment. But the computational result is collected only in the output membrane. It is not necessary that the membranes should connect or communicate to each other directly and the communication of membranes are permitted only if the membranes are designed in the system. The communication of membranes is represented graphically by using the synapses (syn). Ceterchi et.al. [6] defined a type of tissue-like P system in which instead of considering a hierarchical structure, membranes are placed at the nodes of a graph.

By linking the two models, conditional communication P system and Tissue-like P system, Robinson et.al. introduced tissue-like P system with conditional communication in membranes for the tiles by means of tile pasting. It is noticed that the evaluation rules are applied only in maximally parallel way to generate the pictures. Puzzle grammar and context-free grammars have been studied in [10]. By connecting P system and iso-array grammar rules, iso-array rewriting P system with context free iso-array rules is introduced in [2]. Contextual iso-array rewriting P system [3] is introduced to generate the triangular picture languages. Also we have examined the generating powers of the triangular tile pasting P system and array generating petrinets in [4]. Inspired by these studies, in this paper conditional communication on triangular tile pasting P system is introduced to generate the triangular picture languages. The computational powers of triangular tile pasting P system and iso-array grammars are examined here.

2. Preliminaries

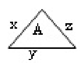
Tiling is an art and it is a well-known theory in the applications of pattern generation. In this section we recollect the notion of iso-triangular tiles and tile pasting system. A tile is a topological disc with closed boundary in the XOY plane, whose edges are gluable. A tiling is a family of countable tiles with no gaps or overlaps that covers the Euclidean plane.

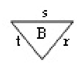
Definition 2.1. *Consider the labeled iso-triangular tiles*

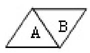


, whose horizontal (Vertical) and side edges are of length 1unit and $1/\sqrt{2}$ *unit respectively.*

Definition 2.2. *Generally a pasting rule is a pair* (x, y) *of labeled tiles with distinct edges (not necessarily distinct).*

For example, the triangular tile 

and tile 

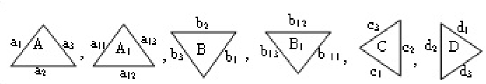
are joined by the edges (z, t)*, which means that the edge* z *of tile* A *is glued with the edge* t *of tile* B*, we get the pattern* 

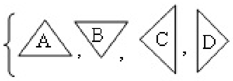
Note that the edges are of same length. The set of all edge labels is called an edge set denoted by E*.*

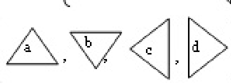
Tile pasting rules of the tiles A, B, C, D are given below:

1. Tile A can be glued with tile B by the pasting rules $\{(a_1, b_1), (a_2, b_2), (a_3, b_3)\}$ with tile C by the rule $\{(a_3, c_1)\}$ and with tile D by the rule $\{(a_1, d_3)\}$
2. Tile B can be glued with tile A by the pasting rules $\{(b_1, a_1), (b_2, a_2), (b_3, a_3)\}$ with tile C by the rule $\{(b_1, c_3)\}$ and with tile D by the rule $\{(b_3, d_1)\}$
3. Tile C can be glued with tile A by the pasting rule $\{(c_1, a_3)\}$ with tile B by $\{(c_3, b_1)\}$ and with the tile D by the pasting rules $\{(c_1, d_1), (c_2, d_2), (c_3, d_3)\}$
4. Tile D can be glued with tile A by the pasting rule $\{(d_3, a_1)\}$ with tile B by $\{(d_1, b_3)\}$ and with the tile C by the pasting rules $\{(d_1, c_1), (d_2, c_2), (d_3, c_3)\}$.

Notations:

1. Iso-triangular tiles = 

2. Non-terminal symbols $N =$ 

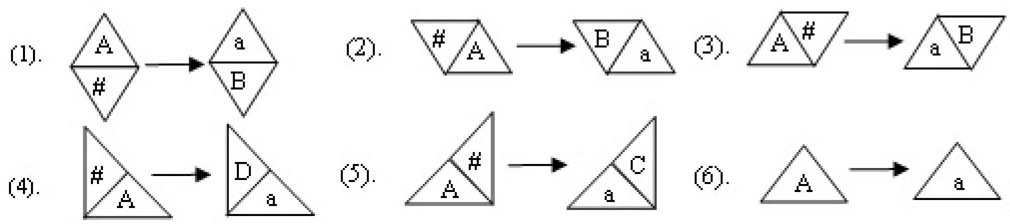
3. Terminal symbols $T =$ 

The edge set $E = \{a_1, a_2, a_3, a_{11}, a_{12}, a_{13}, b_1, b_2, b_3, b_{11}, b_{12}, b_{13}, c_1, c_2, c_3, d_1, d_2, d_3\}$. Note that in all further results the tiles are mentioned by their labels to avoid taking much space.

Definition 2.3. *A Triangular tile pasting system (TTPS) is* $S = (\Sigma, P, t_0)$*, where* Σ *is a finite set of labeled triangular tiles,* P *is a finite set of pasting rules and* t_0 *is the axiom of pattern.*

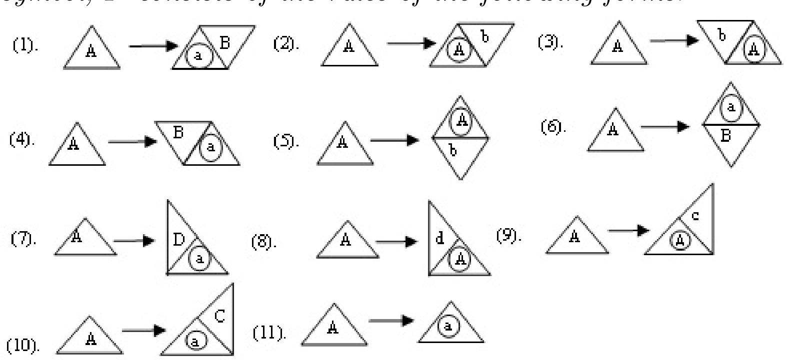
A pattern p_2 is generated from a pattern p_1 by applying the pasting rules in parallel manner to the edges of the pattern p_1, where pasting is possible. Note that the labels of pasted edges in a pattern are ignored once the tiles are pasted. The set of all patterns generated from the axiom t_0 constitutes the picture language $L(TTPS)$.

Definition 2.4. *A regular iso-array grammar (RIAG) is a structure $G = (N, T, P, S)$ where $N = \{A, B, C, D\}$, and $T = \{a, b, c, d\}$ are finite sets of symbols (isosceles right angled triangular tiles), $N \cap T = \phi$, Elements of N and T are called non-terminals and terminals respectively. $S \in N$, S is the start symbol of the axiom, P consists of rules of the following forms:*



Similar rules can be given for the other tiles B, C and D. The set of all languages generated by RIAG is RIAL. By this grammar rules one can rewrite a non-terminal iso-triangular tile on the left hand side by a terminal iso-triangular tile on the right hand side. The non-terminal iso-triangular tile associated with the terminal on the right hand side of the rule can be admitted to the empty place.

Definition 2.5. *Basic Puzzle Iso-Array Grammar is a structure $G = (N, T, P, S)$, where $N = \{A, B, C, D\}$ and $T = \{a, b, c, d\}$ are finite sets of symbols. The non terminals are in N and the terminal symbols are in T. $S \in N$ is the start symbol, P consists of the rules of the following forms:*

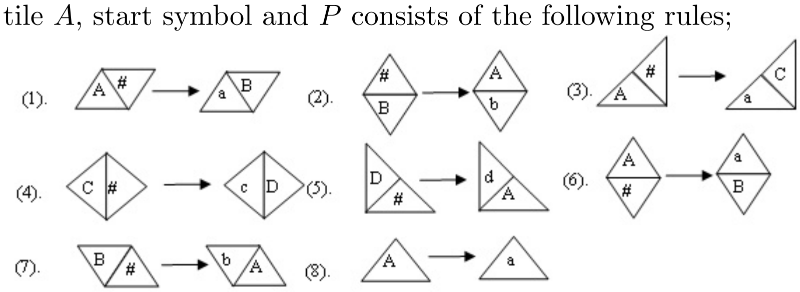


Similar rules can be given for the other tiles B, C, D. The language generated by BPIAG is $L(BPIAG)$. The set of all languages generated by BPIAG is BPIAL. In BPIAG the non-terminal iso-triangular tile of left hand side of a rule can be replaced by a circled iso-triangular tile of right hand side.

3. Triangular tile pasting P system

Definition 3.1. *A triangular tile pasting P system (TTPPS) is defined as* $\Pi = (\Sigma, \mu, F_1, F_2, \ldots, F_m, R_1, R_2, \ldots, R_m, i_0)$*, where* Σ *is a finite set of iso-triangular tiles.* μ *is a membrane structure. In* μ*, the membranes are labeled in a one-one manner with the labels* $1, 2, \ldots, m$*.* $F_1, F_2, \ldots, F_m$ *are finite sets of picture over the tiles of* Σ *associated with* m *regions of the membranes.* $R_1, R_2, \ldots, R_m$ *are finite sets of rules of the type rules* $(t_i, (x_i, y_i), 1 \leq i \leq n)$ *associated with* m *regions of* μ *and* i_0 *is the output membrane, which is an elementary membrane. An evaluation in TTPPS is defined in such a way that, to each picture pattern in each region of the system, a pasting rule could be applied. The picture pattern is moved (retained) in to another region (in the same region) by the target indication associated with the pasting rule.*
A computation is successful only if the computation is stopped, that is if there is no further rule can be applied. The result of halting picture pattern is composed only by the pasting rules and the pattern halts in the membrane labeled by i_0*; the pattern halts in the membrane labeled with* i_0*. The set of all picture pattern computed by a TTPPS is denoted by* $TTPPL(\Pi)$*. The set of all such languages* $TTPPL(\Pi)$ *is generated by the system* Π *is denoted by* $TTPPL_m$*.*

Example 3.1. A class of language of arrow heads can be generated by a RIAG with 4 distinct labeled iso-triangular tiles.
Consider the RIAG $G = (N, T, P, S)$ where $N = \{A, B, C, D\}$ and $T = \{a, b, c, d\}$ are finite sets of non-terminal and terminal symbols respectively. S is the labeled tile A, start symbol and P consists of the following rules;



The picture language is shown with three members in Figure 1

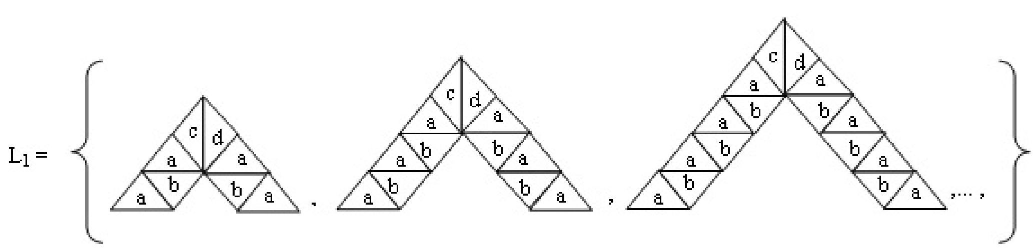
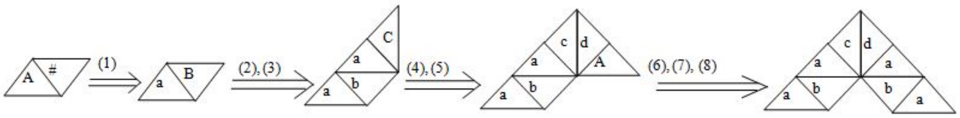


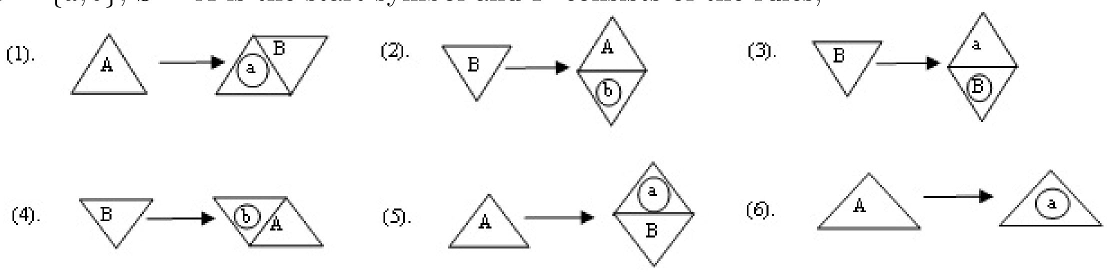
Figure 1: Language of arrow heads with the tiles "c" and "d" on the top

The first member of the picture language is shown with the derivation steps.



In the derivation of second member of the language the rules are taken in the sequence 1, 2, 1, 2, 3, 4, 5, 6, 7, 6, 7, 8. Note that only arrow heads can be generated by this RIAG.

Example 3.2. Consider the BPIAG $G = (N, T, P, S)$, where $N = \{A, B\}$, $T = \{a, b\}$, $S = A$ is the start symbol and P consists of the rules;



The picture language consists of arrow heads with a tile on the top is generated by BPIAG. The picture language is shown in figure 2.

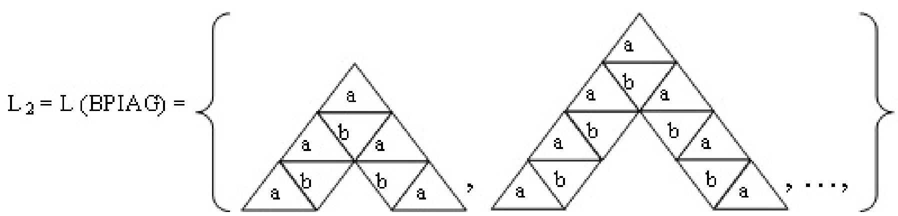
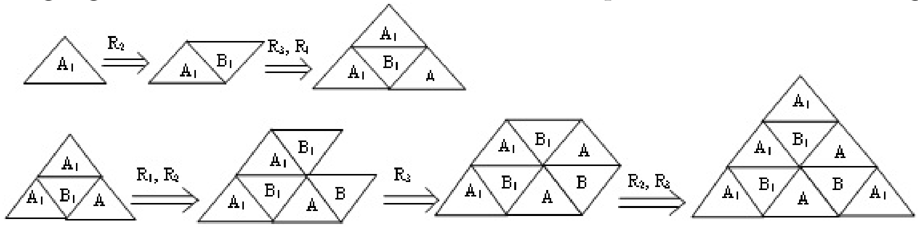


Figure 2: The language of arrow heads with tile "a" on the top

The first member of the languages can be derived with the sequence of rules 1, 2, 1, 3, 4, 5, 4, 6.

Example 3.3. The triangular tile pasting P system $\Pi_1 = (\Sigma, [_1[_2]_2[_3]_3]_1, F_1, F_2, F_3, R_1, R_2, R_3, 1)$ generates the family of two dimensional iso-triangular picture languages $\mathcal{L}_1$ and $\mathcal{L}_2$. Here $\Sigma = \{A, A_1, B, B_1\}$, $F_2 = A$, $F_1 = \phi$, $F_3 = \phi$,
$R_1 = \{(A, (a_3, b_3), in_2), (B_1, (b_{11}, a_1), here)\}$,
$R_2 = \{(B, (b_2, a_{12}), in_3), (B, (b_1, a_{11}), in_3), (A_1, (a_{13}, b_{13}), in_3)\}$,
$R_3 = \{(B, (b_2, a_2), in_2), (B_1, (b_{12}, a_{12}), out), (B, (b_1, a_1), out)\}$
and 1 is the output region.

The rules of R_1, R_2 and R_3 are applied with the target indications which belongs to the set $tar \in \{here, in, out\}$ and then the resultant iso-triangular picture pattern is collected in the output region one finally. Two members of the Picture language $\mathcal{L}_1$ is shown below with derivation steps and $\mathcal{L}_1$ is shown in figure 3.



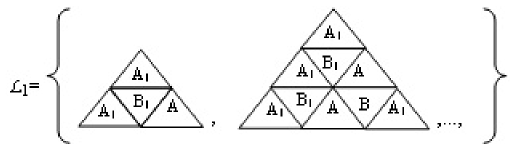


Figure 3: Language consists of iso-triangular arrays

On the other hand, the computation starts from the region R_2. The first member of the language "adjoin of iso-triangles" is collected in the region 1. Again the generation is started in the same region one the 2nd member of the language is generated and it is collected in the region one. The computation is continued in this way, the members of the language are collected in the region 1.

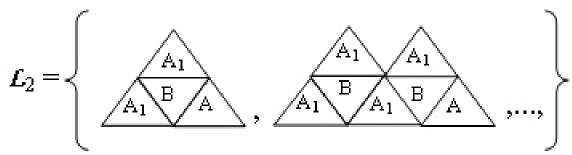


Figure 4: The family of languages of adjoin of iso-triangles

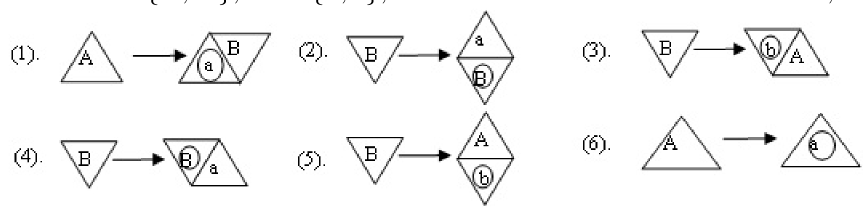
Theorem 3.1. $L(BPIAG) - L(RIAG) \neq \phi$

Proof. From the Definitions 2.4 and 2.5 all the rules of BPIAG contains the rules of RIAG. Hence it is clear that RIAG is the proper subset of BPIAG. In BPIAG the non-terminal symbol in the left hand side of the rules are replaced by the circled symbol (The circled symbol being either non-terminal or terminal). But in RIAG the non-terminal symbols of left hand side of the rules are replaced only by terminal symbols. The picture language L_2 shown in Example 3.2 cannot be generated by any RIAG. In L_2 the top of each member has iso-triangular tile a and in L_1 top of each member has iso-triangular tiles c and d. Here the shapes of L_1 and L_2 are same but they are not equal. □

Theorem 3.2. $TTPPL_3 - BPIAL \neq \phi$

Proof. The language $\mathcal{L}_1$ is generated by the TTPPS Π_1 explained in example 3.3 with the rules $R_1 = \{(A, (a_3, b_3), in_2), (B, (b_{11}, a_1), here)\}$, $R_2 = \{(B, (b_2, a_{12}),$

$in_3), (B, (b_1, a_{11}), in_3), (A_1, (a_{13}, b_{13}), in_3)\}$, R_3=$\{(B, (b_2, a_2), out), (B_1, (b_{12}, a_{12}),$ $out), (B, (b_1, a_1), out)\}$ (refer Example 3.3) cannot be generated by any BPIAG. Since in BPIAG, the non-terminal symbols of left hand side of each rule is replaced only by the circled symbol (circled symbol may be non-terminal or terminal). By the rules of BPIAG the resultant iso-array shown in Figure 3 cannot be generated. For example, consider the BPIAG $G_1 = (N, T, P, S)$, where $N = \{A, B\}$, $T = \{a, b\}$, $S = A$ and P consists of the rules;



The sequence of rules 1, 4, 5, 6 and the sequence of rules 1, 2, 3, 6 of BPIAG generate the languages L_3 and L_4 respectively shown in Figures 5 and 6. □

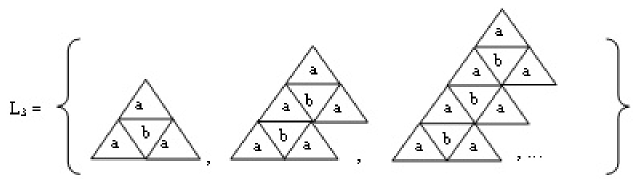


Figure 5: Language consists of step model of iso-triangles

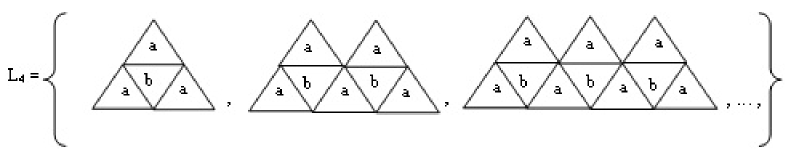


Figure 6: Language consists of adjoin of iso-triangles

Theorem 3.3. $TTPPL_3 - RIAL \neq \phi$

Proof. The family of languages $\mathcal{L}_3$ is generated by TTPPS $\Pi_2 = (\Sigma, [_1[_2]_2[_3]_3]_1,$ F_1, $F_2, F_3, R_1, R_2, R_3, 3)$, where $\Sigma = \{A, B, B_1\}$, $F_1 = A$, $F_2 = \phi$, $F_3 = \phi$, $R_1 =$ $\{(A, (a_3, b_3), here), (B, (b_2, a_2), here), (A, (a_3, c_1), in_2)\}$, $R_2 = \{(C, (c_2, d_2), here),$ $(D, (d_3, a_1), in_3)\}$, $R_3 = \{(A, (a_2, b_{12}), here)$, $(B_1, (b_{11}, a_1), here)\}$ and 3 is the output region. The family of language $\mathcal{L}_3$ generated by the above rules are collected in the region three of the membrane 3. It is shown in the Figure 7. This family of language cannot be generated by any RIAG. (refer Example 3.3). □

Theorem 3.4. $TTPPL_3 \cap L(TTPS) \neq \phi$

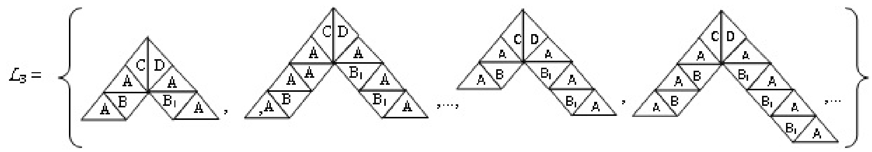


Figure 7: Language consists of arrow heads and arrow heads with long tail on the right

Proof. The triangular tile pasting system $G = (\Sigma, P, t_0)$ generates the picture language L_5, where $\Sigma = \{A, B, B_1\}$, $t_0 = A$ and $P = \{(a_3, b_3), (b_2, a_2), (a_3, c_1), (c_2, d_2), (d_3, a_1), (a_2, b_{12}), (b_{11}, a_1)\}$. The computation starts with the rule (a_3, b_3) with the tile A. Then the rules given in P are applied one by one. The resultant language is $\mathcal{L}_3$. It is shown in Figure 7.
On the other hand if the computation starts by the rule (a_2, b_{12}), the language L_5 can be generated and which is given in the Figure 8. Thus the TTPS generate the families of languages $\mathcal{L}_3$ and L_5.

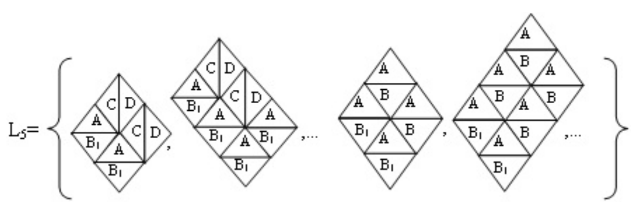


Figure 8: A language of overlaps of Rhombuses

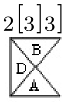
TTPPS defined in Theorem 3.3 generates the picture language $\mathcal{L}_3$ shown in the Figure 7 (refer Theorem 3.3). Hence the intersection of families of languages generated by TTPPS and TTPS is non-empty. □

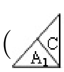
4. Conditional communication on triangular tile pasting P system

Definition 4.1. *Conditional communication on triangular tile pasting P system (CCTTPPS) is defined in a way such that* $\Pi = (V, \mu, (t_0)_i, syn, (R_{(i,j)}, P_{(i,j)}, F_{(i,j)}), i_0)$ *where* V *is the set of all labeled iso-triangular tiles,* μ *is the membrane structure contains finite no of membranes, which are labeled from the set* $\{1, 2, 3, \ldots n\}$*,* t_0 *is the axiom of picture pattern, present in the membrane* i $(1 \leq i \leq m)$*,* $Syn = \{(i, j)/i, j \in k \ (1 \leq k \leq m)\}$ *is the set of links of the membranes,* $R_{(i,j)}$ *is the finite set of evolution rules over the set of iso-triangular tiles* V *present in the regions of the membrane system. An evolution rule is a pair* $\{(\alpha_i, \beta_i)/\alpha_i, \beta_i \in E\}$*,* $P_{(i,j)}$ *is the set of permitting conditions and* $F_{(i,j)}$ *is the set of forbidden conditions associated with region* (i, j) *where* $1 \leq i, j \leq m$*. The conditions can be defined as follows;*

1. **Empty** *There is no restriction or condition given on generating the pattern and the pattern transferred from one membrane to another membrane freely. We denote the permitting and forbidden conditions are empty. An empty permitting condition is denoted by* $(true, in_j)$ *and an empty forbidden condition by* $(false, in_j)$*.*
2. **Labeled Tile Checking** $P_{(i,j)}$ *present in each region is a set of pair* (α, in_j) *for* $\alpha \in V$ *and* in_j *is the target indication which transfer the pattern to the membrane* j *and* $F_{(i,j)}$ *present in each region of membrane is a pair* $(\beta, not\ in_j)$ *for* $\beta \in V$*. A pattern can be sent from a membrane to another membrane if the pair* $(\alpha, in_j) \in P_{(i,j)}$ *and if the pair* $(\beta, not\ in_j) \in F_{(i,j)}$*,* β *does not belongs to* $P_{(i,j)}$*.*
3. **Sub Pattern Checking** *Each* P_i *is a set of pair* (u, in_j) *for* $u \in V^+$ *and each* F_i *is a set of pair* $(v, not\ in_j)$ *for* $v \in V$*. A pattern* p *can go to another membrane only if there is a pair* $(u, in_j) \in P_i$ *with* $u \in sub\ pattern(p)$ *and for each pair* $(v, not\ in_j) \in F_i$ *with* $v \notin sub\ pattern(p)$*.*
The evolution method is explained below.
The tile pasting rules present in each region is applied non-deterministically. The pattern obtained in this way is checked through by the conditions in $P_{(i,j)}$ *and* $F_{(i,j)}$ *presented in the corresponding region. If the condition* $P_{(i,j)}$ *is satisfied in the region then pattern sent to the* j^{th} *membrane. If the pattern fulfills the forbidden condition* $F_{(i,j)}$ *then the pattern retained in the same region. If the pattern is not satisfied by both the conditions then the picture pattern retained in the same region of the membrane. The computational process continued in this manner and then the resultant picture pattern is collected in the output membrane. If the permitting condition* $P_{(i,j)}$ *is never satisfied by the pattern, then the computation is an unsuccessful one.*

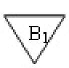
The set of all such picture patterns generated by this way is denoted by $L(CCTTPPS)$*. The family of all such languages* $L(CCTTPPS)$ *generated by the system* $CCTTPPS(\Pi)$ *is denoted by* $L_m(CCTTPPS)$*.*

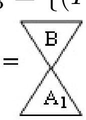
Example 4.1. Consider $\Pi = (V, [_1[_2]_2[_3]_3]_1, t_0, syn, (R_i, P_i, F_i), i_0 = 1)$,
where $V = \{A, A_1, B, B_1, C, D\}$, $t_0 =$ 

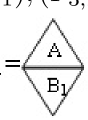
$F_1 = \phi$, $F_2 = ($ 

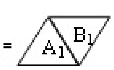
$, out)$, $F_3 = \phi$. $F_3 = \phi$, $F_3 = \phi$

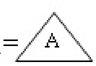
$R_1 = \{(a_3, b_{13}), (a_3, b_3), (b_{12}, a_2)$, $P_1 = \{(P_{1,11}, in_2), (P_{1,12}, in_2)\}$,
$R_2 = \{(b_1, a_1), (a_{13}, c_1), (a_{13}, b_{13})$, $P_2 = \{(P_{2,11}, in_3), (P_{2,12}, in_3)\}$,
$R_3 = \{(b_{11}, a_{11})$, $P_3 = \{(P_{3,11}, in_1), (P_{3,12}, in_2)\}$,

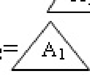
$P_{1,11} =$ 

, $P_{1,12} =$ 

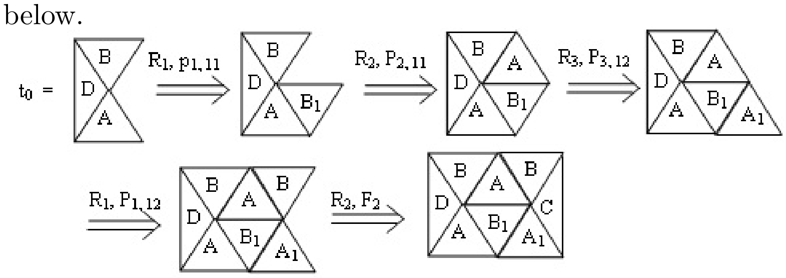
, $P_{2,11} =$ 

, $P_{2,12} =$ 

$P_{3,11} =$ 

, $P_{3,12} =$ 

The first member of the picture language is shown with the derivation steps below.



The language is shown with three members in Figure 9.

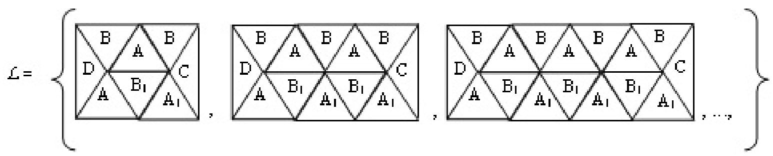


Figure 9: Language of rectangles with size of $4(n+1)$ iso-triangular tiles

Each member of the language consists of $4(n + 1)$ iso-triangular tiles and n is the number of the member of the languages.

Theorem 4.1. *$L(TTPS)$ and $\mathcal{L}(CCTTPPS)$ are not disjoint*

It is true from the Definitions 2.3 and 4.1

Theorem 4.2. $\mathcal{L}(TTPPS) - \mathcal{L}(CCTTPPS) \neq \phi$

It is true from the Definition 3.1 and 4.1

5. Conclusion

In this paper we have defined the triangular tile pasting P system with conditional communication and explained with suitable example. The generating powers of regular iso array grammar and basic puzzle iso-array grammar have been studied. The computational powers of the triangular tile pasting system , the triangular tile pasting P system and conditional communicational triangular tile pasting P system are examined.

References

[1] K. Bhuvaneswari, T. Kalyani, *Triangular tile pasting P system for pattern generation,* Conference Proceeding of ICAMTCS-2013, 26-30 (2013).

[2] K. Bhuvaneswari, T. Kalyani, D. Gnanaraj Thomas, A.K. Nagar, T. Robinson, *Iso-array rewriting P systems with context-free rules*, International Journal of Mathematics for Applications, 3, 1 (2014), 1-16.

[3] K. Bhuvaneswari, T. Kalyani, D.G. Thomas, *Contextual iso-triangular array P system and iso-triangular picture languages*, Asian Conference on Membrane Computing ACMC, 1-5, 2014.

[4] K. Bhuvaneswari, T. Kalyani, D. Lalitha, *Triangular Tile Pasting P system and Array Generating Petrinets*, International Journal of Pure and Applied Mathematics, 107 (2016), 111-128.

[5] R. Ceterchi, M. Mutyam, GH. Paun, K.G. Subramanian, *Array-rewriting P systems*, Natural Computing, 2 (2003), 229-249.

[6] R. Ceterchi, R. Gramatovici, N. Jonaska, K.G. Subramanian, *Tissue-like P Systems with Active Membranes for Picture Generation*, Fundamenta Informaticae, (2003), 311-328.

[7] G.H. Paun, *Membrane Computing: An Introduction,* Springer-Verlag, Berlin, Heidelbrg, 2002.

[8] S. Jebasingh, T. Robinson, Atulya K. Nagar, *A variant of Tile Pasting System for Tiling Patterns*, In: Proc. 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications, BIC-TA2010 (UK), 2 (2010), 1568-1576.

[9] Kalyani, K. Sasikala, V.R. Dare, T. Robinson, *Triangular pasting system*, in: K.G. Subramanian, K. Rangarajan, M. Mukund (Eds.), Formal Models, Languages and Applications, Series in Machine Perception and Artificial Intelligence, 66 (2006), 195-211.

[10] M. Nivat, A. Saoudi, K.G. Subramanian, R. Siromoney, V.R. Dare, *Puzzle grammars and context-free array grammars*, Int. Journal of Pattern Recognition and Artificial Intelligence, 5 (1991), 663-676.

[11] K.G. Subramanian, T. Robinson, Atulya K. Nagar, *Tile Pasting P System Model for Pattern Generation*, in Third Asian International Conference on Modeling & Simulation, (2009), 359-362.

Accepted: 1.11.2017