



ISO-ARRAY REWRITING P SYSTEMS WITH CONTEXT-FREE ISO-ARRAY RULES

KASI BHUVANESWARI, THIYAGARAJAN KALYANI,
DURAIRAJ GNANARAJ THOMAS, ATULYA K. NAGAR
AND ROBINSON THAMBURAJ

Abstract. A new computing model called P system is a highly distributed and parallel theoretical model, which is proposed in the area of membrane computing. Ceterchi et al. initially proposed array rewriting P systems by extending the notion of string rewriting P systems to arrays (2003). A theoretical model for picture generation using context-free iso-array grammar rules and puzzle iso-array grammar rules are introduced by Kalyani et al. (2004, 2006). Also iso-array rewriting P systems for iso-picture languages have been studied by Annadurai et al. (2008). In this paper we consider the context-free iso-array rules and context-free puzzle iso-array rules in iso-array rewriting P systems and examine the generative powers of these P systems.

1. INTRODUCTION

Among various models employed for pattern representation and analysis, the array grammar has attracted more and more attention because it has several advantages over others [8, 9]. The new computing model, called P system was introduced by G. H. Păun [6, 7]. This model deals with distributed computing models inspired by the structure and the functioning of the living cell. The syntactic techniques have an important role in this computability model.

This model processes multi-sets of objects in the regions that are defined by a hierarchical arrangement of membranes in which the evolution rules can be associated with the regions of the membranes. The main area of study of the theoretical models includes digital pictures, image analysis and picture generations. Ceterchi et al. began a study on linking the two areas of membrane computing and picture grammars by relating P systems and array rewriting grammars generating picture languages and proposing array rewriting P systems [2]. Subsequently, a number of P systems with array objects and different kinds of rewriting were introduced in [5, 10, 11]. The two dimensional grammar models called puzzle grammars for generating connected arrays of unit cells are discussed in [12, 13]. The comparisons with pure two dimensional context-free grammars and extension of P2DCFG are discussed with array rewriting P system model in [14, 15].

MSC (2010): primary 68Q42, 68Q45.

Keywords: membrane computing, array grammars, iso-arrays, P systems.

The authors would like to thank the Liverpool Hope University for the support.

Iso-arrays are made of isosceles right angled triangular tiles and iso picture is a picture formed by catenating iso-arrays of the same size. Iso-arrays can be used in tiling system, pattern generation and kolam pattern. Wang tiling systems, context-free iso-array grammars and context-free puzzle iso-array grammars are used for picture generation [3, 4]. Iso-array rewriting P systems for picture languages are studied in [1]. This iso-array rewriting P system can be used as a basic framework for the picture extraction in contour image retrieval system in which the picture position and orientation can be analyzed.

In this paper, iso-array rewriting P systems with context-free iso-array grammar rules and context-free puzzle iso-array grammar rules are used to generate iso picture languages and examine the generative powers of the resulting iso-array rewriting P systems.

2. PRELIMINARIES

In this section we recall the notions of iso-triangular tiling systems, iso-arrays, iso picture languages and context-free iso-array, context-free puzzle iso-array grammars [3, 4].

Let $\Sigma = \left\{ \begin{array}{c} a_1 \triangleleft a_3 \\ a_2 \end{array} \right.$, $\begin{array}{c} b_2 \\ \triangleleft B \\ b_3 \end{array} \triangleright b_1$, $\begin{array}{c} c_3 \\ \triangleleft C \\ c_1 \end{array} \triangleright c_2$, $\begin{array}{c} d_1 \\ \triangleleft D \\ d_3 \end{array} \triangleright d_2 \right\}$ be a finite set of labeled isosceles

right angled triangular tiles of dimensions $1/\sqrt{2}, 1/\sqrt{2}$ and 1 unit are obtained by intersecting a unit square by its diagonals. 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 rules $\{(a_3, c_1)\}$ and with tile D by the rule $\{(a_1, d_3)\}$. In a similar way the glueable rules can be defined for the tiles B, C and D .

Definition 2.1. An iso-array is an isosceles right angled triangular arrangement of tiles A, B, C, D whose sides of equal lengths are denoted by S_1, S_3 and the side of a different length by S_2 and it consists of (for some integer $m > 0$) tiles A, B, C, D alongside S_2 . A U -iso-array of size m is formed exclusively by m A -tiles on side S_2 , and it is denoted by U_m . It will have m^2 tiles in total (including the m A -tiles on S_2). Similarly, D -iso-array, L -iso-array, and R -iso-array are formed exclusively by B -tile, C -tile, and D -tile on side S_2 , respectively.

Example 2.2. A U -iso-array of size 3 (U_3), D -iso-array of size 3 (D_3), L -iso-array of size 3 (L_3) and R -iso-array of size 3 (R_3) are shown in Figure 1.

Iso-arrays of the same size can be catenated using the following catenation operations.

- (1) Horizontal Catenation:

The horizontal catenation is defined between U -iso-array and D -iso-array of the same size, and it is denoted by the symbol \ominus .

- (2) Vertical catenation:

The vertical catenation is defined between L -iso-array and R -iso-array of the same size and it is denoted by the symbol \oplus .

- (3) Right Catenation:

The right catenation is defined between any two glueable iso-arrays of the

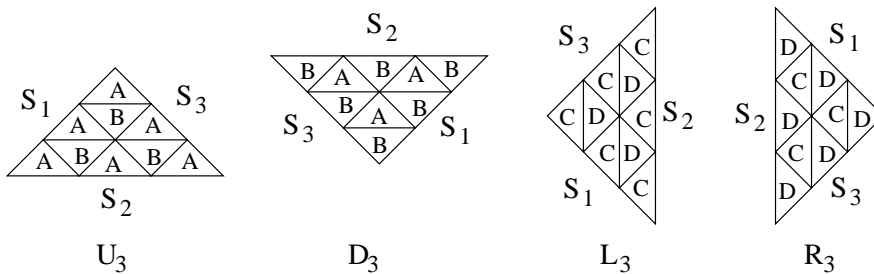


Figure 1. Iso-arrays of size 3.

same size and it is denoted by the symbol \oslash . This catenation includes the following:

- (a) $D \oslash U$ (b) $R \oslash U$ (c) $D \oslash L$ (d) $R \oslash L$.
- (4) Left Catenation:

The left catenation is defined between any two gluable iso-arrays of the same size and it is denoted by the symbol \oslash . This catenation includes the following:

- (a) $U \oslash D$ (b) $U \oslash L$ (c) $L \oslash R$ (d) $R \oslash D$.

Definition 2.3. An iso-picture is a picture formed by catenating iso-arrays of the same size. It is said to be of size (n, m) if there are n iso-arrays of size m catenated to form the iso picture. The number of tiles in any iso picture of size (n, m) is nm^2 . The set of all iso pictures over the alphabet Σ is denoted by Σ_I^{**} . An iso picture languages L over Σ is a subset of Σ_I^{**} .

Definition 2.4. Let p be an iso-picture of size (n, m) . We denote by $B_{n', m'}(p)$, the set of all sub iso-pictures of p of size (n', m') , where $n' \leq n, m' \leq m$. \hat{p} is an iso-picture obtained by surrounding p with a special blank symbols

$$\triangle_{\#A}, \nabla_{\#B}, \triangleleft_{\#C}, \triangleright_{\#D} \notin \Sigma.$$

Definition 2.5. A Context-Free Iso-Array Grammar (CFIAG) is a structure $G = (N, T, P, S)$ where

$$N = \{ \triangle_A, \nabla_B, \triangleleft_C, \triangleright_D \} \text{ and } T = \{ \triangle_a, \nabla_b, \triangleleft_c, \triangleright_d \}$$

are finite non-empty sets of symbols (isosceles right angled triangular tiles), $N \cap T = \phi$. Elements of N and T are called non-terminals and terminals, respectively. $S \in N$ is the start symbol or a axiom. P consists of rules of the form $\alpha \rightarrow \beta$, where α and β are finite connected array of one or more triangular tiles

over $N \cup T \cup \{ \triangle_{\#A}, \nabla_{\#B}, \triangleleft_{\#C}, \triangleright_{\#D} \}$ and satisfy the following conditions:

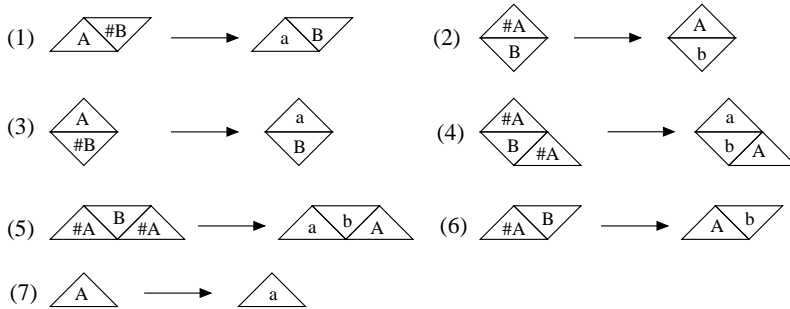
- (1) The shapes of α and β are identical.
- (2) α contains exactly one non-terminal and possibly one or more empty symbols (#).

- (3) Terminals in α are not rewritten.
 (4) Application of the rule $\alpha \rightarrow \beta$ preserves the connectedness of the host array (that is, the application of the rule to a connected array results in a connected array).

The rule $\alpha \rightarrow \beta$ is applicable to a finite connected array γ over

$N \cup T \cup \{ \triangle_{\#A}, \nabla_{\#B}, \triangleleft_{\#C}, \triangleright_{\#D} \}$, if α is a sub array of γ and in a direct derivation step, one of the occurrences of α is replaced by β , yielding a finite connected array δ . We denote the relation by $\gamma \Rightarrow_G \delta$. The relation \Rightarrow_G^* is the reflexive and transitive closure of \Rightarrow_G . The array language generated by G is defined by $\{ \delta \mid S \Rightarrow_G^* \delta, \delta \text{ is a finite connected array over } T \}$ and is denoted by $L(G)$. The class of iso picture languages generated by CFIAGs is denoted by $\mathcal{L}(CFIAG)$.

Example 2.6. Consider the context-free iso-array grammar (CFIAG) $G = (N, T, P, S)$ where $N = \{ \triangle_A, \nabla_B \}$, $T = \{ \triangle_a, \nabla_b \}$, $S = \triangle_A$ the axiom of pattern and P consists of the following rules:



This CFIAG generates a language $L \in \mathcal{L}(CFIAG)$.

The sample derivations of the iso picture pattern of size 4 is shown in Figure 2.

Hence, it is clear that $L(G) = \{ \triangle_a, \begin{array}{c} \triangle_a \\ \triangle_a \triangle_b \triangle_a \\ \dots \end{array} \}$.

Definition 2.7. A Context-Free Puzzle Iso-Array Grammar (CFPIAG) is a structure $G = (N, T, P, S)$, where N, T and S are defined as in Definition 2.5 and P is the set of rewriting rules of the form $X \rightarrow \alpha$, where X can be any of the tiles

$\{ \triangle_A, \nabla_B, \triangleleft_C, \triangleright_D \}$ and α is a finite connected array of one or more iso-triangular arrays, each cell containing either a non-terminal or a terminal symbol, with the symbol in one of the cells of α being circled.

As before, the derivations begin with S written in a unit cell in the two dimensional plane with all other cells containing the blank symbols

$\{ \triangle_{\#A}, \nabla_{\#B}, \triangleleft_{\#C}, \triangleright_{\#D} \}$. In a direct derivation step, a non-terminal X in

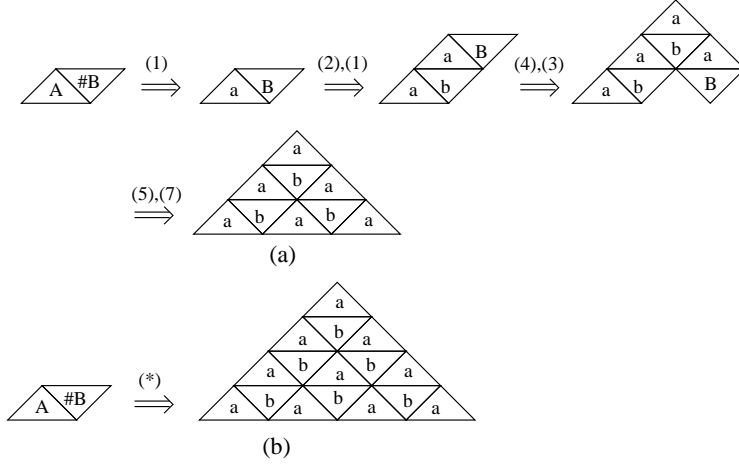


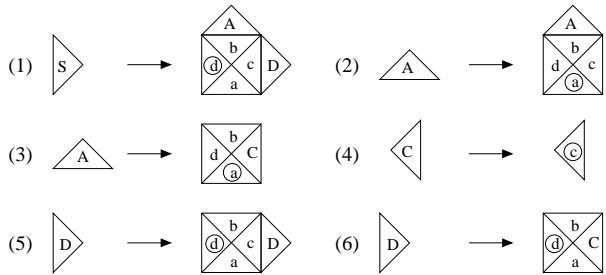
Figure 2. (a) Derivation of iso-array of size 3 (b) Derivation of iso-array of size 4.

a cell is replaced by the right side α of the rule $X \rightarrow \alpha$. In the replacement, the circled symbol occupies the cell with symbol X and the remaining symbols of α occupying their respective relative positions with respect to the circled symbol of α . Again the rewriting by $X \rightarrow \alpha$ is possible only when the cells to be filled in by the non-circled symbols of α contain the blank symbols $\{\triangleleft \#A, \triangleright \#B, \triangleleft \#C, \triangleright \#D\}$.

Example 2.8. A context-free puzzle iso-array grammar, generating digitized staircase patterns is $G = (N, T, P, S)$ where

$$N = \{\triangleleft S, \triangleleft A, \triangleright B, \triangleright D\}, T = \{\triangleleft a, \triangleright b, \triangleleft c, \triangleright d\}, S = \triangleleft S.$$

P consists of the following rules:



A picture pattern generated by G is shown in Figure 3 with a sample derivation.

3. ISO-ARRAY REWRITING P SYSTEMS WITH CONTEXT-FREE ISO-ARRAY RULES

In this section we consider the iso-array rewriting P systems with context-free iso-array rules in the regions associated with the membranes.

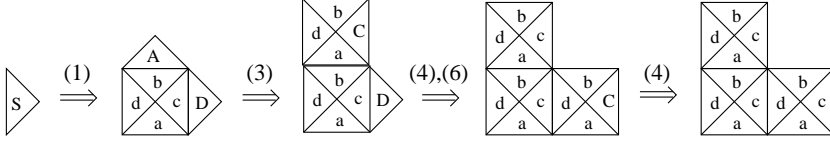


Figure 3. Staircase pattern.

Definition 3.1. An iso-array rewriting P system of degree $m \geq 1$, using rules of context-free iso-array grammars (context-free puzzle iso-array grammars) can be defined as

$$\Pi = (V, T, \{ \triangle \#A, \nabla \#B, \triangleleft \#C, \triangleright \#D \}, \mu, F_1, F_2, \dots, F_m, R_1, R_2, \dots, R_m, i_0),$$

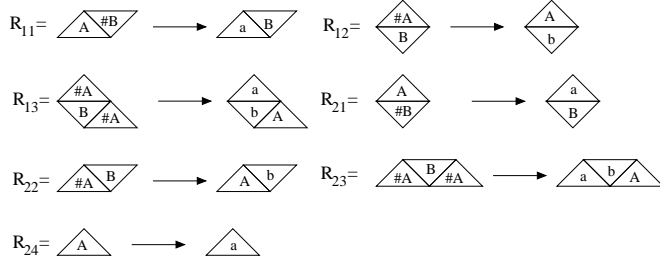
where V is the finite set of total alphabets, $T \subset V$ is the finite set of terminal iso-arrays, $\{ \triangle \#A, \nabla \#B, \triangleleft \#C, \triangleright \#D \}$ are blank symbols, μ is the membrane structure associated with m regions, F_1, F_2, \dots, F_m are finite sets of iso-arrays over V associated with m regions of μ . R_1, R_2, \dots, R_m are finite sets of iso-array rewriting rules over V associated with m regions of the membrane system μ . The rules of context-free iso-array grammars are associated with the target indications $tar \in \{here, out, in\}$. The target indication “here” means that the resultant picture remains in the same region, “out” means the resultant picture is sent out of the current region, the target indication “in” means the resultant picture sent into the inner region, i_0 is the label of output membrane of μ . If there is no internal membrane, then the indication “in” could not be applied.

The process or computation is said to be successful if the generated picture reaches the output membrane by applying the rules of CFIAG (CFPIAG). The result of a halting computation consists of iso-triangular arrays over V which is placed in the region of the output membrane. The set of all such computed or generated pictures by the system Π is denoted by $IAL(\Pi)$. The family of all iso-array languages $IAL(\Pi)$ generated by Π with m membranes is denoted by $IARP_m$ (CFIAG) ($IARP_m$ (CFPIAG)).

Example 3.2. Consider an iso-array rewriting P system with context-free iso-array grammar rules

$$\Pi_1 = (\{ \triangle A, \nabla B, \triangle a, \nabla b \}, \{ \triangle a, \nabla b \}, \{ \triangle A, \nabla B \}, [2[1]1]_2, \triangle A, \phi, R_1, R_2, 2)$$

where $R_1 = \{(R_{11}, here), (R_{12}, here), (R_{13}, out)\}$, $R_2 = \{(R_{21}, here), (R_{22}, here), (R_{23}, in), (R_{24}, here)\}$ and the rules of CFIAG are given below:



The derivation of the sample picture begins with the rule R_{11} of R_1 and then R_{12} . The rules R_{11}, R_{12}, R_{11} are applied. Then the rule R_{13} is applied with the target indication “out”. Then the resultant picture pattern is sent out to the region 2. In this region the rules R_{21}, R_{22}, R_{21} are applied and then R_{23} is applied. The target indication “in” helps to send the picture into the region 1 again. In this region R_{11}, R_{13} are applied. The resultant picture is once again sent out to the region 2 with the help of the target indication “out”. In region 2, the rule R_{24} is applied with the target indication “here” and the picture pattern digitized pyramid (iso-array of size 4) (see Figure 2(b)) is collected in the output membrane 2 as there is no non-terminal in the picture.

On the other hand, in the region 1, the rules $(R_{11}, here), (R_{13}, out)$ are applied. Because of the rule R_{13} the resultant picture pattern is sent out to the region 2. In this region the rule $(R_{21}, here)$ is applied and then the rule (R_{23}, in) is also applied. The target indication “here” causes the picture pattern to remain in the same region 2 and hence again the rules $(R_{21}, here), (R_{23}, in)$ are applied repeatedly. Then the rule R_{24} is applied. The resultant picture pattern is collected in region 2. The picture pattern with a sample derivation is shown in Figure 4.

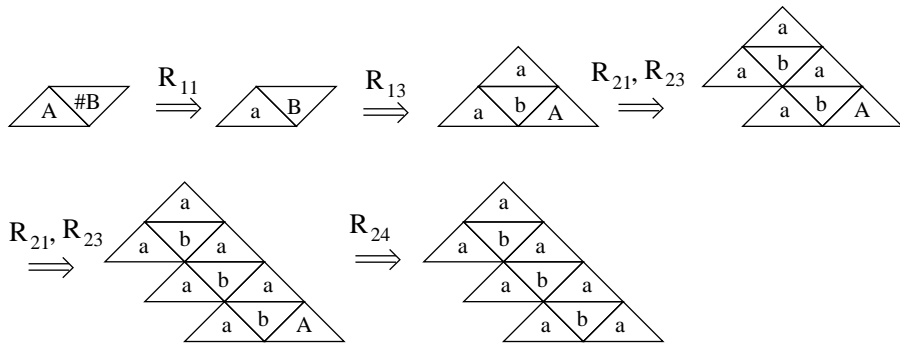


Figure 4. Overlapping triangles.

Hence it is clear that

$$L_1 = IAL(\Pi_1) = \left\{ \triangle_a, \begin{array}{c} \triangle_a \\ \triangle_a \triangle_b \triangle_a \end{array}, \dots, \begin{array}{c} \triangle_a \\ \triangle_a \triangle_b \triangle_a \\ \triangle_a \triangle_b \triangle_a \end{array}, \dots \right\}.$$

Theorem 3.3. *The two classes of languages $IARP_2(CFIAG)$ and $\mathcal{L}(CFIAG)$ have a non-empty intersection, i.e., $IARP_2(CFIAG) - \mathcal{L}(CFIAG) \neq \emptyset$.*

Proof. It is clear that $IARP_1(CFIAG) = \mathcal{L}(CFIAG)$, see [1]. Hence it is true that $\mathcal{L}(CFIAG) \subseteq IARP_2(CFIAG)$. Now we consider an iso-array language $L_1 \in IARP_2(CFIAG)$ given in Example 3.2. It is clear that $L_1 \notin \mathcal{L}(CFIAG)$. \square

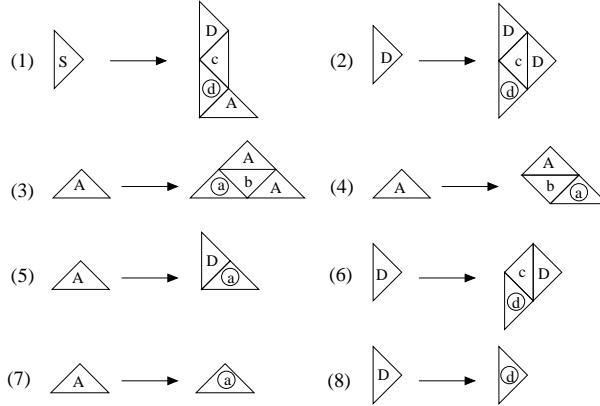
Theorem 3.4. (i) *The two classes $IARP_1(CFPIAG)$ and $\mathcal{L}(CFPIAG)$ coincide, i.e., $IARP_1(CFPIAG) = \mathcal{L}(CFPIAG)$.*

(ii) *The classes $IARP_3(CFPIAG)$ and $\mathcal{L}(CFPIAG)$ have non-empty intersection, i.e., $IARP_3(CFPIAG) - \mathcal{L}(CFPIAG) \neq \emptyset$.*

Proof. (i) The result is obvious from the Definition 2.7.

(ii) Consider the context-free puzzle iso-array grammar (CFPIAG),

$G = (N, T, P, S)$ where $N = \{ \triangleleft S, \triangle A, \triangle B, \triangle C, \triangleright D \}$,
 $T = \{ \triangle a, \triangle b, \triangle c, \triangle d \}$, the start symbol $\triangleleft S$ and P contains the following CFPIAG rules:



A sample derivation of a picture pattern generated by the CFPIAG is given in Figure 5.

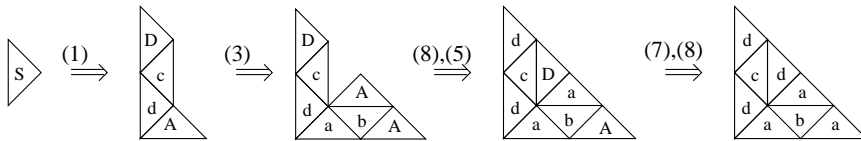


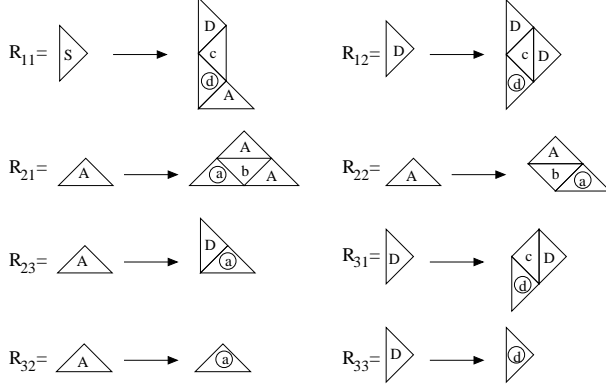
Figure 5. Picture pattern digitized right angled triangle.

Now consider an iso-array rewriting P system with CFPIAG rules

$$\Pi_2 = (\{ \triangle A, \triangle B, \triangle C, \triangleright D, \triangle a, \triangle b, \triangle c, \triangleright d \},$$

$$\left\{ \triangle_a, \nabla_b, \triangle_c, \triangle_d \right\}, [1[2[3]3]2]_1, \triangle_S, \phi, \phi, R_1, R_2, R_3, 3),$$

where $R_1 = \{(R_{11}, in), (R_{12}, in)\}$,
 $R_2 = \{(R_{21}, here), (R_{22}, out), (R_{23}, in), (R_{23}, out)\}$,
 $R_3 = \{(R_{31}, here), (R_{32}, here), (R_{33}, here)\}$ and the rules are given below:



In region one, (R_{11}, in) is applied. The target indication “in” sends the generated pattern into region two. Then the rule $(R_{21}, here)$ generates the new picture pattern and the pattern remains in the same region. Then the rules (R_{23}, in) is applied and the resultant iso-array is sent into membrane 3. In this region the rules R_{32}, R_{33} are applied with the target “here”. Then the resultant iso-array is collected in the same region as shown in Figure 6.

Again applying the rules $R_{11}, R_{21}, R_{21}, (R_{22}, out), R_{12}, R_{23}, R_{31}, R_{32}, R_{33}$ to

the picture \triangle_S with the target indications, the digitized right angled triangle is collected in region three. The sample derivation of the picture pattern is given in Figure 7.

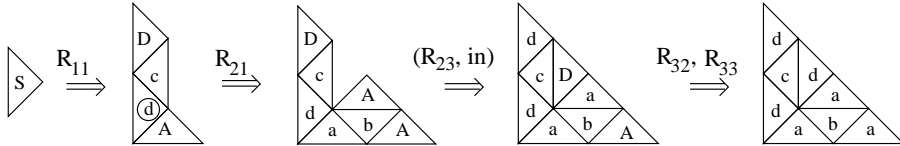


Figure 6. The derivation of digitized right angled triangle with 2 a’s on the base.

Again, in another instance in region one, (R_{11}, in) is applied and the resultant picture is sent into region two. In this region the rules $(R_{21}, here), (R_{23}, out)$ are applied. The target out sends the resultant iso-array out to skin membrane one. In this region, (R_{12}, in) is applied. Then the picture is entered into region two, in which the rules $(R_{21}, here)$ and (R_{23}, in) are applied. The resultant iso-array is sent into region three where the rules $(R_{32}, here), (R_{33}, here)$ are applied and

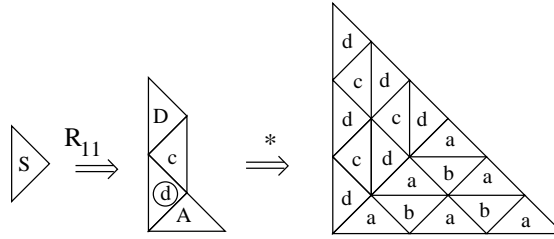


Figure 7. The derivation of digitized right angled triangle with 3a's on the base.

the resultant picture pattern is collected in region three. The sample derivation is given in Figure 8.

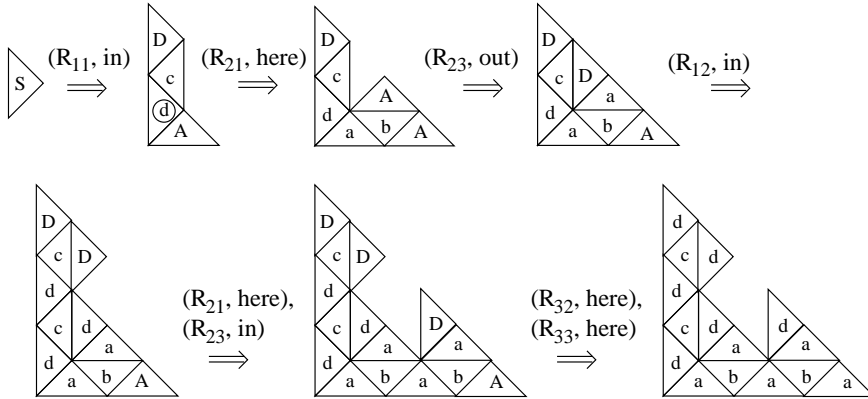


Figure 8. An iso-array.

Let $L_2 = IAL(\Pi_2)$. This language cannot be generated by any $CFPIAG$. \square

Theorem 3.5. (i) *The two classes $IARP_1(CFPIAG)$ and $IARP_1(CFIAG)$ coincide, i.e., $IARP_1(CFPIAG) = IARP_1(CFIAG)$.*

(ii) *The two classes $IARP_2(CFPIAG)$ and $IARP_2(CFIAG)$ have a non-empty intersection, i.e., $IARP_2(CFPIAG) - IARP_2(CFIAG) \neq \emptyset$.*

Proof. (i) It is true that $\mathcal{L}(CFPIAG) = \mathcal{L}(CFIAG)$, see [3, 4].

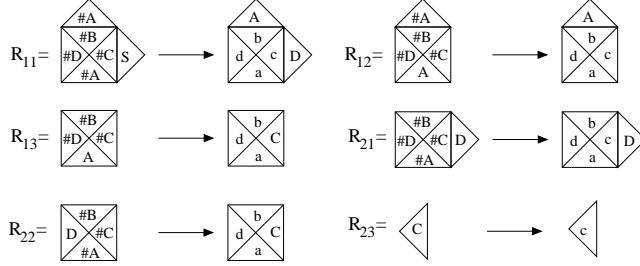
(ii) The iso-array language L_3 consisting of iso-arrays describing the token L with equal arms is in $IARP_2(CFIAG)$. In fact, it is generated by

$$\Pi_3 = (\{ \triangle A, \triangle B, \triangle C, \triangle D, \triangle S, \triangle a, \triangle b, \triangle c, \triangle d, \triangle \#A, \triangle \#B, \triangle \#C, \triangle \#D \},$$

$$\{ \triangle a, \triangle b, \triangle c, \triangle d \},$$

$$[2[1]1]_2, \triangleleft S, \phi, R_1, R_2, 2),$$

where $R_1 = \{(R_{11}, \text{here}), (R_{12}, \text{in}), (R_{13}, \text{in})\}$,
 $R_2 = \{(R_{21}, \text{out}), (R_{22}, \text{here}), (R_{23}, \text{here})\}$ with



The working rules of the system is as follows: In the skin membrane, the rules R_{11} , R_{12} are applied and the resultant iso-array is sent into region two, where the rule R_{21} is applied and the resulting iso-array is sent out to region one of the skin membrane. In this region, the rule R_{13} is applied and the resultant iso-array is sent into region two, in which the rules R_{22} and R_{23} are applied and the final iso-array obtained is the token L , which can be collected in the region of the output membrane two.

The derivation of a member of the language is shown in Figure 9.

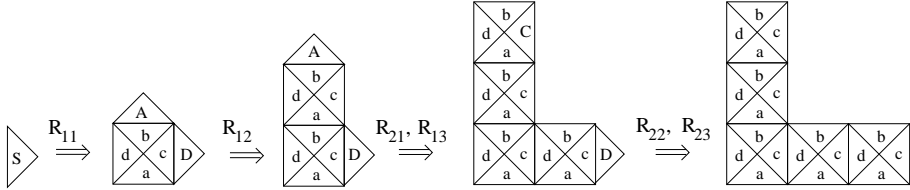


Figure 9. Derivation step of digitized L token.

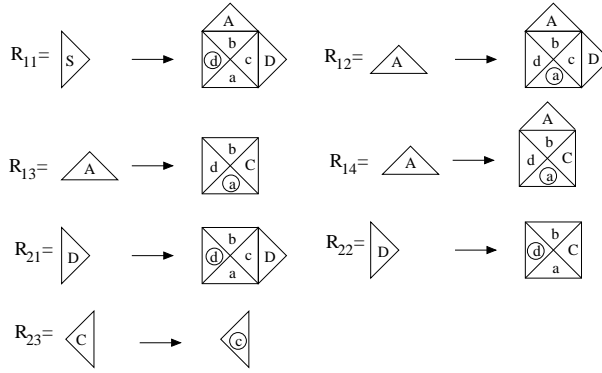
Also the language L_4 consisting of the patterns staircases and L tokens (iso-arrays) is in $IARP_2(CFPIAG)$ is generated by Π_4 .

$$\Pi_4 = \{ \triangleleft A, \triangleleft B, \triangleleft C, \triangleleft D, \triangleleft S, \triangleleft a, \triangleleft b, \triangleleft c, \triangleleft d \},$$

$$\{ \triangleleft a, \triangleleft b, \triangleleft c, \triangleleft d, \triangleleft \#A, \triangleleft \#B, \triangleleft \#C, \triangleleft \#D \},$$

$$[2[1]1]_2, \triangleleft S, \phi, R_1, R_2, 2),$$

where $R_1 = \{(R_{11}, \text{here}), (R_{12}, \text{out}), (R_{13}, \text{out}), (R_{14}, \text{out})\}$,
 $R_2 = \{(R_{21}, \text{in}), (R_{22}, \text{here}), (R_{23}, \text{here})\}$



The derivation of a member of the language L_4 is shown in Figure 10.

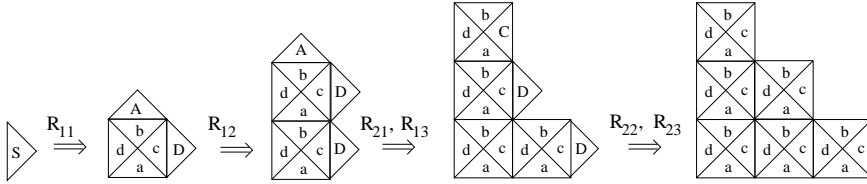


Figure 10. A sample derivation of staircase pattern.

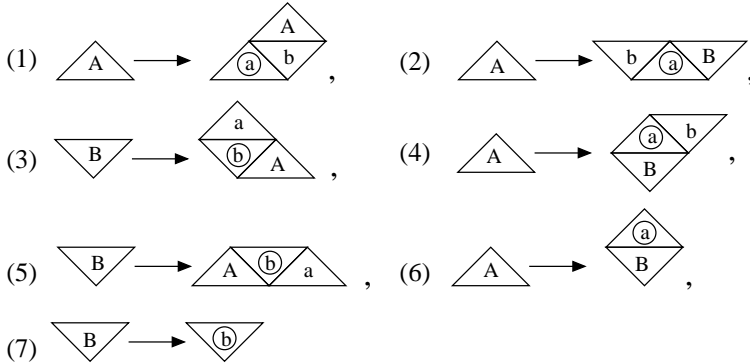
It is clear that $L_4 \notin IARP_2(CFPIAG)$. □

Theorem 3.6. (i) *The two classes $IARP_5(CFPIAG)$ and $\mathcal{L}(CFPIAG)$ have non-empty intersection, i.e., $IARP_5(CFPIAG) - \mathcal{L}(CFPIAG) \neq \emptyset$.*

(ii) *The two classes $IARP_5(CFPIAG)$ and $IARP_3(CFPIAG)$ have non-empty intersection, i.e., $IARP_5(CFPIAG) - IARP_3(CFPIAG) \neq \emptyset$.*

Proof. (i) It is clear from Theorem 3.4.

(ii) Consider the context-free puzzle iso-array grammar $G = (N, T, P, S)$ where $N = \{ \triangle A, \triangle B \}$, $T = \{ \triangle a, \triangle b \}$, P consists of the rules



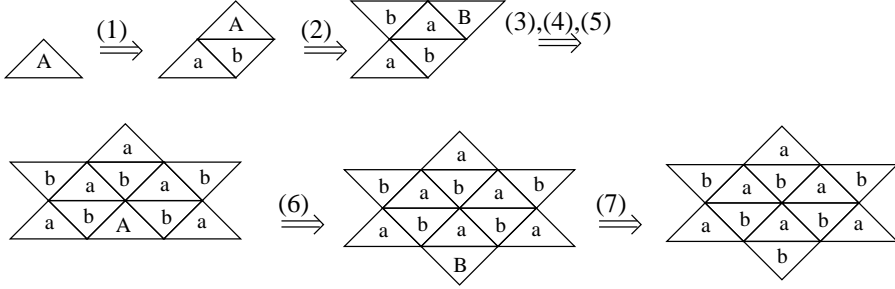


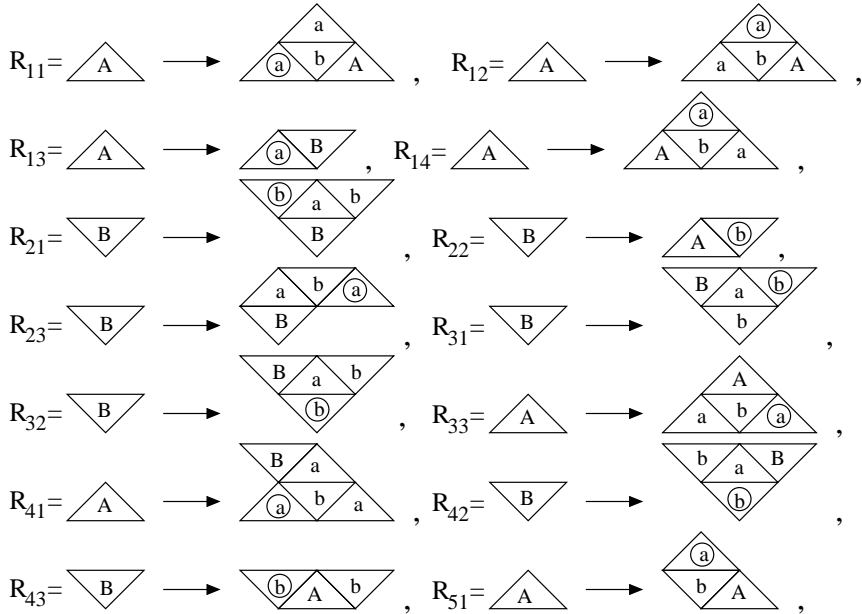
Figure 11. A sample derivation of star pattern.

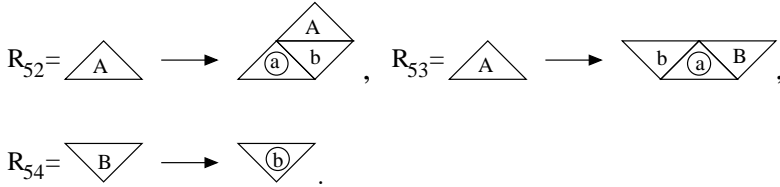
and $\triangleup A$ is the start symbol. The rules of CFPIAG generate star-like pictures (see Figure 11).

Now consider the language $L_5 = IAL(\Pi_5) \in IARP_5(CFPIAG)$ where

$$\Pi_5 = (\{ \triangleup A, \triangleleft B, \triangleup a, \triangleleft b \}, \{ \triangleleft a, \triangleleft b \}, [5[4[3[2[1]1]2]3]4]5, \triangleup A, \phi, \phi, \phi, \phi, R_1, R_2, R_3, R_4, R_5, 5)$$

with $R_1 = \{(R_{11}, here), (R_{12}, here), (R_{13}, out), (R_{14}, out)\}$,
 $R_2 = \{(R_{21}, here), (R_{22}, in), (R_{23}, out), R_{22}, out)\}$,
 $R_3 = \{(R_{31}, here), (R_{32}, in), (R_{33}, out)\}$,
 $R_4 = \{(R_{41}, here), (R_{42}, here), (R_{43}, out)\}$,
 $R_5 = \{(R_{51}, here), (R_{52}, here), (R_{53}, here), (R_{54}, here)\}$ with





The derivation can be started with R_{11} with the target “here” and then R_{12} , R_{13} are applied with their target indications. In region 1 the application of the rule R_{13} with the target out sends the resultant iso-array into region two, where the rules R_{21} , R_{22} are applied. The target in with R_{22} sends the generated iso-array into region one, in which again the rules R_{12} , R_{14} are applied with the target indications. The resultant iso-array is sent out to region two in which the rule R_{23} is applied with the target indication out. In region three the rules R_{31} and R_{32} are applied. Then the resultant picture is sent into region two where the rule R_{22} is applied with the target indication out. The picture pattern once again entered into region three, in which R_{33} is applied with the target indication out. In region four the rules (R_{41} , here), (R_{42} , here), (R_{43} , out) are applied. Then the resultant picture pattern is sent out to region five where the rule R_{51} is applied twice with the target indication here and then R_{52} , R_{53} , R_{54} are applied. The resultant picture pattern “Star with in a star” is collected in region 5. The resultant picture pattern can be shown in Figure 12. It is clear that L_5 does not belong to $L(CFPIAG)$ or $IARP_3(CFPIAG)$.

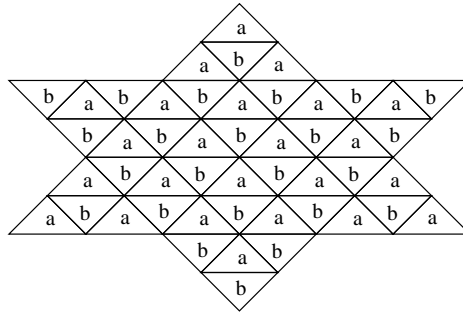


Figure 12. The pattern star within a star.

□

4. CONCLUSION

We have extended the study of iso-array rewriting P system models with the rules of context-free iso-array grammars and context-free puzzle iso-array grammars and compared the generative powers of the iso-array rewriting P systems with the existing models. This study can be continued to generate more interesting pictures like kolam patterns, tiling patterns, etc.

REFERENCES

- [1] S. Annadurai, D. G. Thomas, V. R. Dare and T. Kalyani, *Rewriting P systems generating iso-picture langauges*, 12th International Workshop, IWCIA 2008, Buffalo, NY, USA, April 7–9, 2008, Lecture Notes in Computer Science **4958**, Springer, Berlin, Heidelberg, 2008, 352–362. DOI: 10.1007/978-3-540-78275-9_31
- [2] R. Ceterchi, M. Mutyam, Gh. Păun and K. G. Subramanian, *Array-rewriting P systems*, Natural Computing **2** (2003), 229–249. DOI: 10.1023/A:1025497107681
- [3] T. Kalyani, V. R. Dare and D. G. Thomas, *Local and recognizable iso picture languages*, 11th International Conference, ICONIP 2004, Calcutta, India, November 22–25, 2004, Lecture Notes in Computer Science **3316**, Springer, Berlin, Heidelberg, 2004, 738–743. DOI: 10.1007/978-3-540-30499-9_113
- [4] T. Kalyani, V. R. Dare and D. G. Thomas, *Iso-array acceptors and learning*, 8th International Colloquium, ICGI 2006, Tokyo, Japan, September 20–22, 2006, Lecture Notes in Computer Science **4201**, Springer, Berlin, Heidelberg, 2006, 327–339. DOI: 10.1007/11872436_27
- [5] M. Nivat, A. Saoudi, K. G. Subramanian, R. Siromoney, and V. R. Dare, *Puzzle grammars and context-free array grammars*, Int. J. Patt. Recogn. Artif. Intell. **5** (1991), 663–676. DOI: 10.1142/S0218001491000399
- [6] Gh. Păun, *Computing with membranes*, J. Comput. Syst. Sci. **61** (2000), 108–143. DOI: 10.1006/jcss.1999.1693
- [7] Gh. Păun, *Membrane Computing: An Introduction*, Springer-Verlag, Berlin, Heidelberg, 2002.
- [8] A. Rosenfeld and R. Siromoney, *Picture languages – a survey*, Languages of Design **1** (1993), 229–244.
- [9] R. Siromoney, K. G. Subramanian, V. R. Dare and D. G. Thomas, *Some results on Picture languages*, Pattern Recogn. **32** (1999), 295–304. DOI: 10.1016/S0031-3203(98)00072-7
- [10] K. G. Subramanian, R. Siromoney, V. R. Dare and A. Saoudi, *Basic Puzzle Languages*, Int. J. Patt. Recogn. Artif. Intell. **9** (1995), 763–775. DOI: 10.1142/S0218001495000316
- [11] K. G. Subramanian, R. Saravanan and P. Helen Chandra, *Cooperating basic puzzle grammars*, Lecture Notes in Computer Science **4040**, Springer, Berlin, Heidelberg, 2006, 354–360. DOI: 10.1007/11774938_28
- [12] K. G. Subramanian, R. Saravanan and T. Robinson, *P systems for array generation and application to kolam patterns*, Forma **22** (2007), 47–54.
- [13] K. G. Subramanian, R. Saravanan, M. Geethalakshmi, P. Helen Chandra and M. Margenstern, *P systems with array objects and array rewriting rules*, Prog. Nat. Sci. **17** (2007), 479–485. DOI: 10.1080/10020070708541026
- [14] K. G. Subramanian, M. Geethlakshmi, A. K. Nagar and S. K. Lee, *Array P system model with pure context-free rules*, in: 11th International Conference on Computer Modeling and Simulation, UKSIM '09, 2009, 274–278. DOI: 10.1109/UKSIM.2009.55
- [15] K. G. Subramanian, A. K. Nagar and M. Geethalakshmi, *Pure 2D picture grammars (P2DPG) and P2DPG with regular control*, in: Brimkov et al. (eds.), Combinatorial Image Analysis, Lecture Notes in Computer Sciences **4959**, Springer, Berlin, Heidelberg, 2008, 330–341. DOI: 10.1007/978-3-540-78275-9_29

Kasi Bhuvaneshwari, Department of Mathematics, Sathyabama University, Chennai – 600 119, India

e-mail: bhuvanameshwaran@gmail.com

Thiyagarajan Kalyani, Department of Mathematics, St. Joseph's Institute of Technology, Chennai – 600 119, India

e-mail: kalphd02@yahoo.com

Durairaj Gnanaraj Thomas, Department of Mathematics, Madras Christian College, Tambaram, Chennai – 600 059, India

e-mail: dgthomasmcc@yahoo.com

Atulya K. Nagar, Department of Mathematics and Computer Science, Liverpool Hope University, Hope Park, Liverpool L16 9JD, UK

e-mail: nagara@hope.ac.uk

Robinson Thamburaj, Department of Mathematics, Madras Christian College, Tambaram, Chennai – 600 059, India

e-mail: robin.mcc@gmail.com