Math. Appl. **3** (2014), 17–31
DOI: 10.13164/ma.2014.02

SHUFFLE ON ARRAY LANGUAGES GENERATED BY ARRAY GRAMMARS

D. K. SHEENA CHRISTY, VEDHANAYAGAM MASILAMANI,
DURAIRAJ GNANARAJ THOMAS, ATULYA K. NAGAR
AND ROBINSON THAMBURAJ

Abstract. Motivated by the studies done by G. Siromoney et al. (1973) and Alexandru Mateescu et al. (1998) we examine the language theoretic results related to shuffle on trajectories by making use of Siromoney array grammars such as $(R : R)AG$, $(R : CF)AG$, $(CF : R)AG$, $(CF : CF)AG$, $(CS : R)AG$, $(CS : CS)AG$ and $(CF : CS)AG$ which are more powerful than the Siromoney matrix grammars (1972) and are used to make digital pictures.

1. Introduction

There have been several studies on generation of languages of finite words [2]. One such tool for the generation of languages is found to be the shuffle on trajectories [5]. The shuffle on trajectories is based on the parallel composition. This operation is introduced using a uniform method based on the notion of trajectory.

A trajectory is a segment of a line in a two dimensional XY plane. The line can change its direction only at points with non negative integer coordinates. A trajectory defines how to skip from a word to another word during the shuffle operation. Languages consisting of trajectories are either regular or context free string languages considered in [5].

The generation of two-dimensional arrays by theoretical models [2, 4, 6, 8–10] describes a wide variety of interesting classes of pictures. To develop the study on parallel contextual array grammars, the shuffle operation on finite arrays with trajectories has been introduced [3]. Based on the studies of rewriting rules on various Siromoney matrix grammars [8], the shuffle on trajectories on finite and infinite rectangular array languages has been done in [1]. The closure properties for different classes of Siromoney matrix languages with respect to shuffle on trajectories and the comparison of its generative power with other array grammars have been made in [1].

In formal language theory new families of languages are introduced by changing the type of rewriting rules. Furthermore in order to obtain richer families restrictions are imposed on the use of production rules in well known families of

MSC (2010): primary 68Q42, 68Q45.
Keywords: shuffle on trajectories, array languages, array grammars.
The authors would like to thank the support rendered by the Department of Mathematics and Computer Science, Liverpool Hope University, UK.

grammars. Several such studies are available in the literature [7]. In [9] Siromoney et al. have given the generalized notion of rewriting rules in string grammars to array rewriting rules for matrix grammars. These rules are either regular, context free or context sensitive in nature but the use of production rules is restricted by the condition of row and column concatenation. As an application, Siromoney array grammars generate interesting kolam patterns [10,11] and it can be noted that Siromoney matrix languages are particular cases of Siromoney array languages [9].

In this paper we continue the study of the shuffle on trajectories as a tool for obtaining various finite array languages generated by Siromoney array grammars [9] such as $(R:R)AG$, $(R:CF)AG$, $(CF:R)AG$, $(CF:CF)AG$, $(CS:R)AG$, $(CS:CS)AG$ and $(CF:CS)AG$. We further examine the language theoretic results related to $L_1 ⧢_T L_2$ where L_1 and L_2 can be taken from different or same families of array languages and T is either a regular language or a context-free language.

2. Basic definitions

In this section we first review some of the basic definitions from [1,9].

Definition 2.1. Let Σ be a finite alphabet of symbols. A picture A over Σ is a rectangular $m \times n$ array of elements of the form

$$A = \begin{matrix} a_{m1} & \dots & a_{mn} \\ \vdots & \ddots & \vdots \\ a_{11} & \dots & a_{1n} \end{matrix} = [a_{ij}]_{m \times n}.$$

The set of all pictures or arrays over Σ is denoted by Σ^{**}. A picture language or an array language over Σ is a subset of Σ^{**}.

Definition 2.2. Let $A = [a_{ij}]_{m\times p}$ and $B = [a_{ij}]_{n\times q}$. The column concatenation $A ⦶ B$ of A and B is defined only when $m = n$ and is given by

$$A ⦶ B = \begin{matrix} a_{m1} & \dots & a_{mp} & b_{n1} & \dots & b_{nq} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{11} & \dots & a_{1p} & b_{11} & \dots & b_{1q} \end{matrix}.$$

Similarly, the row concatenation $A \ominus B$ of A and B is defined only when $p = q$ and is given by

$$A \ominus B = \begin{matrix} b_{n1} & \dots & b_{nq} \\ \vdots & \ddots & \vdots \\ b_{11} & \dots & b_{1q} \\ a_{m1} & \dots & a_{mp} \\ \vdots & \ddots & \vdots \\ a_{11} & \dots & a_{1p} \end{matrix}.$$

The empty array is denoted by Λ. We have $\Lambda ⦶ P = P ⦶ \Lambda = P$ and $\Lambda \ominus P = P \ominus A = P$ for any $P \in \Sigma^{**}$.

We use $\oplus$ to denote either $⦶$ or $\ominus$. Also when there is no ambiguity and the meaning is clear then the operator $\oplus$ is left out.

Definition 2.3. The column shuffle operation on two arrays P and Q denoted by $P \sqcup\!\sqcup^c Q$ is defined recursively as

$$\begin{aligned} P \sqcup\!\sqcup^c Q &= ((A \obar X) \sqcup\!\sqcup^c (B \obar Y)) \\ &= A \obar (X \sqcup\!\sqcup^c (B \obar Y)) \cup B \obar ((A \obar X) \sqcup\!\sqcup^c Y) \end{aligned}$$

where $P = A \obar X$ and $Q = B \obar Y$, $P, Q \in \Sigma^{**}$, A is the first column of P and B is the first column of Q. The operation is defined only when the number of rows in P and the number of rows in Q are equal. If A is empty then $X = P$. Similarly if B is empty then $Y = Q$. Also $P \sqcup\!\sqcup^c \Lambda = \Lambda \sqcup\!\sqcup^c P = P$.

Example 2.4. Let $P = A \obar X$ where $A = \begin{array}{c} a \\ b \end{array}$, $X = \begin{array}{c} b \\ a \end{array}$ and $Q = B \obar Y$ where $B = \begin{array}{c} c \\ d \end{array}$, $Y = \begin{array}{c} d \\ c \end{array}$. Now

$$X \sqcup\!\sqcup^c (B \obar Y) = \begin{array}{c} b \\ a \end{array} \sqcup\!\sqcup^c \begin{array}{cc} c & d \\ d & c \end{array} = \left\{ \begin{array}{ccc} b & c & d \\ a & d & c \end{array}, \begin{array}{ccc} c & b & d \\ d & a & c \end{array}, \begin{array}{ccc} c & d & b \\ d & c & a \end{array} \right\},$$

$$A \obar (X \sqcup\!\sqcup^c (B \obar Y) = \left\{ \begin{array}{cccc} a & b & c & d \\ b & a & d & c \end{array}, \begin{array}{cccc} a & c & b & d \\ b & d & a & c \end{array}, \begin{array}{cccc} a & c & d & b \\ b & d & c & a \end{array} \right\},$$

$$(A \obar X) \sqcup\!\sqcup^c Y = \begin{array}{cc} a & b \\ b & a \end{array} \sqcup\!\sqcup^c \begin{array}{c} d \\ c \end{array} = \left\{ \begin{array}{ccc} a & b & d \\ b & a & c \end{array}, \begin{array}{ccc} a & d & b \\ b & c & a \end{array}, \begin{array}{ccc} d & a & b \\ c & b & a \end{array} \right\},$$

$$B \obar ((A \obar X) \sqcup\!\sqcup^c Y) = \left\{ \begin{array}{cccc} c & a & b & d \\ d & b & a & c \end{array}, \begin{array}{cccc} c & a & d & b \\ d & b & c & a \end{array}, \begin{array}{cccc} c & d & a & b \\ d & c & b & a \end{array} \right\}.$$

Therefore

$$\begin{aligned} P \sqcup\!\sqcup^c Q = \Bigg\{ & \begin{array}{cccc} a & b & c & d \\ b & a & d & c \end{array}, \begin{array}{cccc} a & c & b & d \\ b & d & a & c \end{array}, \begin{array}{cccc} a & c & d & b \\ b & d & c & a \end{array}, \begin{array}{cccc} c & a & b & d \\ d & b & a & c \end{array}, \\ & \begin{array}{cccc} c & a & d & b \\ d & b & c & a \end{array}, \begin{array}{cccc} c & d & a & b \\ d & c & b & a \end{array} \Bigg\}. \end{aligned}$$

Definition 2.5. The row shuffle operation on two arrays P and Q denoted by $P \sqcup\!\sqcup^r Q$ is defined recursively as

$$\begin{aligned} P \sqcup\!\sqcup^r Q &= ((A \ominus X) \sqcup\!\sqcup^r (B \ominus Y)) \\ &= A \ominus (X \sqcup\!\sqcup^r (B \ominus Y)) \cup B \ominus ((A \ominus X) \sqcup\!\sqcup^r Y) \end{aligned}$$

where $P = A \ominus X$ and $Q = B \ominus Y$, $P, Q \in \Sigma^{**}$, A is the first row of P and B is the first row of Q. The operation is defined only when the number of columns in P and the number of columns in Q are equal. Also $P \sqcup\!\sqcup^r \Lambda = \Lambda \sqcup\!\sqcup^r P = P$.

Definition 2.6. Let $V_1 = \{r, u\}$, $V_2 = \{\ell, d\}$ be the sets of versors in the plane. ℓ, r, u and d stand for the left, right, up and down directions respectively. A trajectory is an element $t \in V_1^* \cup V_2^*$.

Let $|P|_c$ denote the number of columns in the array P, $|P|_r$ denote the number of rows in the array P. If w is a finite string, then $|w|_a$ denotes the number of occurrences of a in w.

Definition 2.7. Let Σ be a finite alphabet, $t \in V_1^*$, $v \in \{r, u\}$ and $P, Q \in \Sigma^{**}$. The column shuffle of P with Q on the trajectory vt, denoted by $P \sqcup\!\sqcup^c_{vt} Q$ is recursively defined as follows. If $P = A\varobar X$ and $Q = B\varobar Y$ where $A, B, X, Y \in \Sigma^{**}$, A and B are the first columns of P and Q respectively, then

$$P \sqcup\!\sqcup^{c}_{vt} Q = ((A \varobar X) \sqcup\!\sqcup^{c}_{vt} (B \varobar Y)) = \begin{cases} A \varobar (X \sqcup\!\sqcup^{c}_{t} (B \varobar Y)), & \text{if } v = r \\ B \varobar ((A \varobar X) \sqcup\!\sqcup^{c}_{t} Y), & \text{if } v = u \end{cases}.$$

If $P = \Lambda$, then

$$\Lambda \sqcup\!\sqcup^{c}_{vt} (B \varobar Y) = \begin{cases} \phi & \text{if } v = r \\ B \varobar (\Lambda \sqcup\!\sqcup^{c}_{t} Y) & \text{if } v = u \end{cases}.$$

If $Q = \Lambda$, then

$$(A \varobar X) \sqcup\!\sqcup^{c}_{vt} \Lambda = \begin{cases} A \varobar (X \sqcup\!\sqcup^{c}_{t} \Lambda) & \text{if } v = r \\ \phi & \text{if } v = u \end{cases} \quad \text{and} \quad \Lambda \sqcup\!\sqcup^{c}_{vt} \Lambda = \begin{cases} \Lambda & \text{if } t = \lambda \\ \phi & \text{otherwise.} \end{cases}.$$

The row shuffle of P with Q on the trajectory vt, $v \in \{\ell, d\}$, $t \in V_2^*$ is defined in a similar way with r, u replaced by ℓ, d and $\varobar$ catenation is replaced by $\ominus$ catenation. Also if $|P|_c \neq |t|_r$ or $|Q|_c \neq |t|_u$ then $P \sqcup\!\sqcup^{c}_{t} Q = \phi$. Similarly if $|P|_r \neq |t|_\ell$ or $|Q|_r \neq |t|_d$ then $P \sqcup\!\sqcup^{r}_{t} Q = \phi$.

If T is a set of trajectories, i.e., $T \subseteq V_1^* \cup V_2^*$, then $P \sqcup\!\sqcup^{c}_{T} Q = \bigcup_{t \in T \cap V_1^*} P \sqcup\!\sqcup^{c}_{t} Q$,

$$P \sqcup\!\sqcup^{r}_{T} Q = \bigcup_{t \in T \cap V_2^*} P \sqcup\!\sqcup^{r}_{t} Q \text{ and } P \sqcup\!\sqcup_{T} Q = \left(\bigcup_{t \in T \cap V_1^*} P \sqcup\!\sqcup^{c}_{t} Q \right) \cup \left(\bigcup_{t \in T \cap V_2^*} P \sqcup\!\sqcup^{r}_{t} Q \right).$$

The above operation is extended to array languages over Σ. If $L_1, L_2 \subseteq \Sigma^{**}$ then

$$L_1 \sqcup\!\sqcup_{T} L_2 = \bigcup_{\substack{P \in L_1, \\ Q \in L_2}} P \sqcup\!\sqcup_{T} Q = \bigcup_{\substack{P \in L_1, \\ Q \in L_2}} \left(\left(\bigcup_{t \in T \cap V_1^*} P \sqcup\!\sqcup^{c}_{t} Q \right) \cup \left(\bigcup_{t \in T \cap V_2^*} P \sqcup\!\sqcup^{r}_{t} Q \right) \right).$$

Example 2.8. Let P, Q and $R \in \Sigma^{**}$. If

$$P = \begin{matrix} a & a & a \\ a & a & a \\ a & a & a \end{matrix}, \quad Q = \begin{matrix} b & b & b \\ b & b & b \\ b & b & b \end{matrix} \quad \text{and} \quad R = \begin{matrix} c & c & c \\ c & c & c \\ c & c & c \end{matrix},$$

then

$$P \sqcup\!\sqcup^{c}_{t} Q = \begin{matrix} a & b & b & a & b & a \\ a & b & b & a & b & a \\ a & b & b & a & b & a \end{matrix}, \quad \text{where} \quad t = ru^2rur \text{ and } P \sqcup\!\sqcup^{r}_{t} R = \begin{matrix} a & a & a \\ c & c & c \\ a & a & a \\ c & c & c \\ c & c & c \\ a & a & a \end{matrix}$$

where $t = \ell d^2 \ell d \ell$.

Definition 2.9. Let $G = (V, I, P, S)$ be an array (rewriting) grammar (AG), where $V = V_1 \cup V_2$, V_1 a finite set of non-terminals, V_2 a finite set of intermediates, I, a finite set of terminals, $P = P_1 \cup P_2 \cup P_3$, P_1 the finite set of non-terminal rules, P_2 the finite set of intermediate rules and P_3 the finite set of terminal rules. $S \in V_1$

is the start symbol, P_1, is a finite set of ordered pairs (u, v), $u, v \in (V_1 \cup V_2)^+$ or $u, v \in (V_1 \cup V_2)_+$.

P_1 is context-sensitive (CS) if there is a (u, v) in P_1 such that $u = u_1 s_1 v_1$ and $v = u_1 \alpha v_1$ where $s_1 \in V_1$, u_1, v_1, α are all in $(V_1 \cup V_2)^+$ or all in $(V_1 \cup V_2)_+$. P_1 is called context-free (CF) if every (u, v) in P_1 is such that $u \in V_1$ and v in $(V_1 \cup V_2)^+$ or $(V_1 \cup V_2)_+$ and regular (R) if $u \in V_1$ and v of the form $U \oplus V$, U in V_1 and V in V_2 or U in V_2 and V in V_1.

P_2 is a set or ordered pairs (u, v), u and v in $(V_2 \cup \{x_1, x_2, \ldots, x_p\})^+$ or u and v in $(V_2 \cup \{x_1, x_2, \ldots, x_p\})_+$; $x_1, x_2, \ldots, x_p$ in I^{++} have the same number of rows in the first case and same number of columns in the second case; i.e., the finite set of intermediate rules involve only intermediates and a finite number of fixed arrays in I^{++}. Further P_2 is such that each intermediate in V_2 generates either a language (called intermediate matrix language) whose terminals are a finite number of arrays with the same number of rows or the transpose of such a language. P_2 is called CS, CF or R depending on whether the intermediate matrix languages generated are CS, CF or R.

P_3 the finite set of terminal rules is ordered pairs (u, v), $u \in (V_1 \cup V_2)$ and v in I^{++}.

Remark 2.10. The transpose of a language L is

$$L^T = \left\{ \begin{array}{ccc} a_{11} & \ldots & a_{m1} \\ \ldots & \ldots & \ldots \\ a_{1n} & \ldots & a_{mn} \end{array} \middle| \begin{array}{ccc} a_{11} & \ldots & a_{1n} \\ \ldots & \ldots & \ldots \\ a_{m1} & \ldots & a_{mn} \end{array} \in L \right\}.$$

Remark 2.11. I^{**} denotes the set of all horizontal sequences of rectangular arrays from I and $I^{++} = I^{**} - \{\epsilon\}$, where ϵ is the empty array. I_{**} denotes the set of all vertical sequences of rectangular arrays from I and $I_{++} = I_{**} - \{\epsilon\}$. Also $(x)^{i+1} = (x)^i \bigcirc\!\!\!\!\mid\; x$, $(x)_{i+1} = (x)_i \ominus x$ where $x \in I^{++}$.

Definition 2.12. An array grammar (AG) is called $(CS : CS)AG$ if the non-terminal rules are CS and at least one intermediate language is CS.

An array grammar is $(CS : CF)AG$ if the non-terminal rules are CS and none of the intermediate language is CS. A grammar is called $(CS : R)AG$ if the non-terminal rules are CS and all the intermediate languages are regular. Similarly for all the other six, viz., $(CF : CS)AG$, $(CF : CF)AG$, $(CF : R)AG$, $(R : CS)AG$, $(R : CF)AG$, $(R : R)AG$.

Definition 2.13. If A is an intermediate, the intermediate matrix language generated by A is $M_A = \{X/A \Rightarrow^* X \in \{x_1, x_2, \ldots, x_p\}^+,\ x_1, x_2, \ldots, x_p \in I^{++}$ and $x_1, x_2, \ldots, x_p$ have same number of rows$\}$ or $M_A = \{X/A \Rightarrow^* X \in \{x_1, x_2, \ldots, x_p\}_+,\ x_1, x_2, \ldots, x_p \in I^{++}$ and $x_1, x_2, \ldots, x_p$ have same number of columns$\}$.

Definition 2.14. $M = \{X/S \Rightarrow_G^* X, X \text{ in } I^{++}\}$ is a (context sensitive : context sensitive) array language ($(CS : CS)AL$) if there exists a $(CS : CS)AG$ G such that $M = M(G)$. Similarly for the remaining eight families.

Notation. We write $\alpha \Rightarrow_P \beta$ when a rule in P_1 or P_2 or P_3 is used in the derivation. The reflexive, transitive closure of $\Rightarrow$ is denoted by $\Rightarrow^*$.

3. Results

In this section we give the existince results for the array languages such as $(R : R)AL$, $(CF : CF)AL$, $(CF : R)AL$, $(CF : CS)AL$ and $(CS : CS)AL$.

Theorem 3.1. *There exist two languages L_1 and L_2 in $(R : R)AL$ and $(CF : CF)AL$ and T is a regular language such that $L = L_1 \sqcup\!\sqcup_T L_2$ is a $(CF : CF)AL$.*

Proof. Let $L_1 = \{L$ token of all sizes with fixed proportion$\}$ be a $(R : R)AL$ generated by $L_1 = L(G_1)$, $G_1 = (V, I, P_1, S_1)$ where $V = V_1 \cup V_2$, $V_1 = \{S_1\}$, $V_2 = \{A_1, B_1\}$, $I = \{X, \cdot\}$, $P_1 = P_{11} \cup P_{12} \cup P_{13}$, with $P_{11} = \{S_1 \to (A_1 \ominus S_1) \oplus B_1\}$, P_{12} be the set of intermediate rules given by L_{A_1} and L_{B_1},

$$P_{13} = \left\{ S_1 \to \begin{matrix} X & \cdot \\ X & X \end{matrix} \right\}, \quad L_{A_1} = \{X \ \ (\cdot)^n | n \geq 1\}, \quad L_{B_1} = \left\{ \begin{matrix} (\cdot)_n \\ X \end{matrix} \ | n \geq 1 \right\}.$$

The language thus generated is

$$L_1 = \left\{ \begin{matrix} X & \cdot \\ X & X \end{matrix}, \begin{matrix} X & \cdot & \cdot \\ X & \cdot & \cdot \\ X & X & X \end{matrix}, \dots \right\}.$$

Let us consider a $(CF : CF)AL$, L_2, which is the language

$$\left\{ \begin{matrix} \cdot & X & X & X \\ \cdot & X & \cdot & X \\ X & \cdot & X & \cdot \\ \cdot & X & \cdot & X \\ X & X & \cdot & X \end{matrix}, \begin{matrix} \cdot & \cdot & X & X & X & X \\ \cdot & \cdot & X & \cdot & X & X \\ \cdot & X & X & X & \cdot & X \\ \cdot & X & \cdot & \cdot & X & \cdot \\ X & \cdot & X & \cdot & X & X \\ \cdot & X & X & \cdot & \cdot & X \\ X & X & X & \cdot & \cdot & X \end{matrix}, \dots \right\},$$

L_2 is generated by the following $(CF : CF)AG$, $G_2 = (V_2, I, P_2, S_2)$ with $V_2 = V' \cup V''$, $V' = \{S_3, S_4, S_2\}$, $V'' = \{A_2, B_2\}$, $I = \{X, \cdot\}$ and $S_2 \in V'$ and $P_2 = P' \cup P'' \cup P'''$ where $P' = \{S_2 \to ((A_2 \ominus S_3) \oplus B_2) \oplus ((S_4 \ominus A_2) \oplus B_2)$, $S_3 \to (A_2 \ominus S_3) \oplus B_2$, $S_4 \to (S_4 \ominus A_2) \oplus B_2\}$,

$$P''' = \left\{ S_3 \to \begin{matrix} X \\ \cdot \\ X \end{matrix}, \ S_4 \to \begin{matrix} X \\ \cdot \\ X \end{matrix} \right\}$$

and

$$P'' : L_{A_2} = \left\{ \begin{pmatrix} \cdot \\ \cdot \end{pmatrix}^n / n \geq 1 \right\} \quad \text{and} \quad L_{B_2} = \left\{ \begin{matrix} (X)_n \\ \cdot \\ (X)_n \end{matrix} \ / n \geq 1 \right\}.$$

Now the shuffling of L_1 and L_2 over the trajectory $T_C = \{r^n u^m / n, m \geq 1\}$, which results in a $(CF : CF)AL$ with the language being of the form

$$L = L_1 \sqcup\!\sqcup_{T_C} L_2 = \left\{ \begin{matrix} X & \cdot & \cdot & \cdot & \cdot & \cdot & X & X & X \\ X & \cdot & \cdot & \cdot & \cdot & \cdot & X & \cdot & X \\ X & \cdot & \cdot & \cdot & \cdot & X & \cdot & X & \cdot \\ X & \cdot & \cdot & \cdot & \cdot & \cdot & X & \cdot & X \\ X & X & X & X & X & X & X & \cdot & X \end{matrix} . \ \cdots \right\}$$

is a $(CF:CF)AL$ generated by the following $(CF:CF)AG$, $G_3=(V_3,I,P_3,S_5)$ where $V_3=V_{31}\cup V_{32}$ with $V_{31}=\{S_1,S_2,S_3,S_4,S_5\}$, $V_{32}=\{A_1,B_1,A_2,B_2\}$ and $P_3=P_{31}\cup P_{32}\cup P_{33}$ where $P_{31}=\{S_5\to S_1 \obar S_2,\ S_1\to(A_1\ominus S_1)\obar B_1,\ S_2\to S_3\obar S_4,\ S_3\to(A_2\ominus S_3)\obar B_2,\ S_4\to(S_4\ominus A_2)\obar B_2\}$,

$$P_{32}=\left\{\begin{array}{ll} L_{A_1}=\{X\ \ (\cdot)^n/n\geq 1\}; & L_{B_1}=\left\{\begin{array}{c}(\cdot)_n\\ X\end{array}\Big/ n\geq 1\right\}\\ L_{A_2}=\left\{\begin{pmatrix}\cdot\\ \cdot\end{pmatrix}^n\Big/ n\geq 1\right\}; & L_{B_2}=\left\{\begin{array}{c}(X)_n\\ \cdot\\ (X)_n\end{array}\Big/ n\geq 1\right\}\end{array}\right\}\text{ and}$$

$$P_{33}=\left\{S_1\to\begin{array}{ccc}X&\cdot&\cdot\\ X&\cdot&\cdot\\ X&X&X\end{array},\ S_3\to\begin{array}{c}X\\ \cdot\\ X\end{array},\ S_4\to\begin{array}{c}X\\ \cdot\\ X\end{array}\right\}.$$

□

Theorem 3.2. *There exist two languages L_1 and L_2 in $(CF:R)AL$ and T a CFL such that $L=L_1\sqcup\!\sqcup_T L_2$ is a $(CF:R)AL$.*

Proof. Let $L_1=\{$ shape I of all sizes with fixed proportion $\}$, i.e.,

$$L_1=\left\{\begin{array}{ccc}X&X&X\\ \cdot&X&\cdot\\ X&X&X\end{array},\quad \begin{array}{ccccc}X&X&X&X&X\\ \cdot&\cdot&X&\cdot&\cdot\\ \cdot&\cdot&X&\cdot&\cdot\\ \cdot&\cdot&X&\cdot&\cdot\\ X&X&X&X&X\end{array},\ \cdots\right\}$$

be a $(CF:R)AL$ generated by the following $(CF:R)AG$, $G_1=(V',I,P',S)$ where $V'=V_1\cup V_2$, $V_1=\{S,S_1\}$, $V_2=\{A,B,C,D\}$, $I=\{X,\cdot\}$ and $P'=P_1\cup P_2\cup P_3$ where $P_1=\{S\to(B\ominus S_1\ominus B)\obar A\obar(B\ominus S_1\ominus B),\ S_1\to(S_1\ominus C)\obar D\}$,

$$P_3=\left\{S_1\to\begin{array}{cc}\cdot&\cdot\\ \cdot&\cdot\\ \cdot&\cdot\end{array},S\to\begin{array}{ccc}X&X&X\\ \cdot&X&\cdot\\ X&X&X\end{array}\right\},\quad P_2=\left\{\begin{array}{l}L_A=\{(X)_{2n+1}/n\geq 1\}\\ L_B=\{(X)^n/n\geq 1\}\\ L_C=\left\{\begin{pmatrix}\cdot\\ \cdot\end{pmatrix}^n/n\geq 1\right\}\\ L_D=\{(\cdot)_{2n+1}/n\geq 1\}\end{array}\right\}.$$

Let us consider another $(CF:R)AL$, $L_2=\{$ the token L of odd sizes and fixed proportion $\}$, generated by a $(CF:R)AG$, $G_2=(V',I,P',S')$ where $V'=V_1\cup V_2$, $I=\{X,\cdot\}$, $P'=P_1\cup P_2\cup P_3$ with $P_1=\{S'\to A\ominus((A\ominus S')\obar B)\obar B\}$,

$$P_3=\left\{S'\to\begin{array}{ccc}X&\cdot&\cdot\\ X&\cdot&\cdot\\ X&X&X\end{array}\right\},\quad P_2=\left\{\begin{array}{l}L_A=\{X\ \ (\cdot)^n/n\geq 1\}\\ L_B=\left\{\begin{array}{c}(\cdot)_n\\ X\end{array}\Big/ n\geq 1\right\}\end{array}\right\}.$$

Let us consider the trajectory T as CFL, $T_C=\{r^n u^n/n\geq 1\}$.

Now the shuffling of L_1 and L_2 over the trajectory T is a $(CF:R)AL$ given by

$$L=L_1\sqcup\!\sqcup_{T_C}L_2=\left\{\begin{array}{cccccc}X&X&X&X&\cdot&\cdot\\ \cdot&X&\cdot&X&\cdot&\cdot\\ X&X&X&X&X&X\end{array},\right.$$

$$\left.\begin{array}{cccccccccc} X & X & X & X & X & X & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & X & \cdot & \cdot & X & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & X & \cdot & \cdot & X & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & X & \cdot & \cdot & X & \cdot & \cdot & \cdot & \cdot \\ X & X & X & X & X & X & X & X & X & X \end{array}, \cdots \right\}$$

which can be generated by the following $(CF:R)AG$, $G'' = (V'', I, P'', S'')$ where $V'' = V_1 \cup V_2$, $V_1 = \{S_1, S_2, S_3, S''\}$, $V_2 = \{A_1, B_1, C_1, D_1, E_1, F_1\}$, $I = \{X, \cdot\}$, $P'' = P_1 \cup P_2 \cup P_3$ where

$$\begin{aligned} P_1 = \{ & S'' \to S_2 \obar S_3, S_2 \to (B_1 \ominus S_1 \ominus B_1) \obar A_1 \obar (B_1 \ominus S_1 \ominus B_1), \\ & S_3 \to A_1 \obar (E_1 \ominus (S_3 \obar F_1) \ominus B_1), \\ & S_1 \to (S_1 \ominus C_1) \obar D_1, S_3 \to (S_3 \obar F_1) \ominus C_1\}, \end{aligned}$$

$$P_2 : L_{A_1} = \{(X)_{2n+1}/n \geq 1\}, L_{B_1} = \{(X)^n/n \geq 1\},$$

$$L_{C_1} = \left\{ \begin{pmatrix} \cdot \\ \cdot \end{pmatrix}^n \Big/ n \geq 1 \right\}, \quad L_{D_1} = \{(\cdot)_{2n+1}/n \geq 1\},$$

$$L_{E_1} = \{(\cdot \quad \cdot)^{2n}/n \geq 1\}, \quad L_{F_1} = \{ \begin{pmatrix} \cdot \\ \cdot \\ \cdot \end{pmatrix}^n \Big/ n \geq 1\} \quad \text{and}$$

$$P_3 = \left\{ S'' \to \begin{array}{cccccc} X & X & X & X & \cdot & \cdot \\ \cdot & X & \cdot & X & \cdot & \cdot \\ X & X & X & X & X & X \end{array}, \quad S_1 \to \begin{array}{cc} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{array}, \quad S_3 \to \begin{array}{ccc} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{array} \right\}.$$

□

Theorem 3.3. *There exist two languages L_1 and L_2 in $(CF:CS)AL$ and T a regular language such that $L = L_1 \sqcup\!\sqcup_T L_2$, is a $(CF:CS)AL$.*

Proof. Let us consider a $(CF:CS)AL$ L_1, a set of matrices of the form

$$L_1 = \left\{ \begin{array}{cccc} X & \cdot & \cdot & X \\ X & \cdot & \cdot & X \\ \cdot & \cdot & \cdot & \cdot \\ X & X & X & X \\ X & \cdot & \cdot & X \\ \cdot & X & X & \cdot \\ X & \cdot & \cdot & X \\ X & X & X & X \end{array}, \begin{array}{cccccc} X & \cdot & \cdot & \cdot & \cdot & X \\ X & \cdot & \cdot & \cdot & \cdot & X \\ X & \cdot & \cdot & \cdot & \cdot & X \\ \cdot & X & \cdot & \cdot & X & \cdot \\ X & X & \cdot & \cdot & X & X \\ X & \cdot & \cdot & \cdot & \cdot & X \\ X & X & X & X & X & X \\ \cdot & X & \cdot & \cdot & X & \cdot \\ X & \cdot & X & X & \cdot & X \\ X & X & \cdot & \cdot & X & X \\ X & X & X & X & X & X \end{array}, \cdots \right\}$$

which is generated by the $(CF:CS)AG$, $G_1 = (V', I, P', S')$ where $V' = V_1 \cup V_2$, $V_1 = \{S', S_1\}$, $V_2 = \{A, B\}$, $I = \{X, \cdot\}$ and $P' = P_1 \cup P_2 \cup P_3$ where

$$P_1 = \{S' \to B \obar (A \ominus S_1) \obar B, S_1 \to B \obar (A \ominus S_1) \obar B\},$$

$$P_3 = \left\{ S_1 \to \begin{matrix} X & X \\ \cdot & \cdot \\ X & X \\ \cdot & \cdot \\ X & X \end{matrix} \right\}, \quad P_2 = \left\{ \begin{matrix} L_A = \left\{ \begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{pmatrix}^n \Big/ n \geq 1 \right\} \\ L_B = \left\{ \begin{matrix} (X)_n \\ \cdot \\ (X)_n \\ \cdot \\ (X)_n \\ \cdot \end{matrix} \Big/ n \geq 1 \right\} \end{matrix} \right\}.$$

Let us consider another $(CF : CS)AL$ L_2, a set of matrix of the form

$$L_2 = \left\{ \begin{matrix} A & X & X & A \\ A & X & X & A \\ X & X & X & X \\ B & A & A & B \\ B & X & X & B \\ X & B & B & X \\ C & X & X & C \\ C & C & C & C \end{matrix}, \; \begin{matrix} A & X & X & X & X & A \\ A & X & X & X & X & A \\ X & A & X & X & A & X \\ B & A & X & X & A & B \\ B & X & X & X & X & B \\ B & B & A & A & B & B \\ X & B & X & X & B & X \\ C & X & B & B & X & C \\ C & C & X & X & C & C \\ C & C & C & C & C & C \end{matrix}, \; \dots \right\}$$

and L_2 is generated by $(CF : CS)AG$, $G_2 = (V'', I', P'', S'')$ where $V'' = V_1 \cup V_2$ with $V_1 = \{S'', S_2\}$, $V_2 = \{C', D'\}$, $I' = \{A, B, C, X\}$ and $S'' \in V_1$.

The production rule P'' consists of $P'' = P_1 \cup P_2 \cup P_3$, where $P_1 = \{S'' \to C' \bigoplus (D' \ominus S_2) \bigoplus C', S_2 \to C' \bigoplus (D' \ominus S_2) \bigoplus C'\}$,

$$P_3 = \left\{ S_2 \to \begin{matrix} A & A \\ X & X \\ B & B \\ X & X \\ X & X \end{matrix} \right\}, \quad P_2 = \left\{ \begin{matrix} L_{C'} = \left\{ \begin{matrix} (A)_n \\ X \\ (B)_n \\ X \\ (C)_n \end{matrix} \Big/ n \geq 1 \right\} \\ L_{D'} = \left\{ \begin{pmatrix} X & X \\ X & X \\ X & X \end{pmatrix}^n \Big/ n \geq 1 \right\} \end{matrix} \right\}.$$

Consider a regular trajectory $T_C = \{(ru)^n / n \geq 1\}$. Then

$$L = L_1 ⧢_{T_C} L_2 = \left\{ \begin{matrix} X & A & \cdot & X & \cdot & X & X & A \\ X & A & \cdot & X & \cdot & X & X & A \\ \cdot & X & \cdot & X & \cdot & X & \cdot & X \\ X & B & X & A & X & A & X & B \\ X & B & \cdot & X & \cdot & X & X & B \\ \cdot & X & X & B & X & B & \cdot & X \\ X & C & \cdot & X & \cdot & X & X & C \\ X & C & X & C & X & C & X & C \end{matrix}, \right.$$

$$\left.\begin{array}{cccccccccc} X & A & \cdot & X & \cdot & X & \cdot & X & X & A \\ X & A & \cdot & X & \cdot & X & \cdot & X & X & A \\ X & A & \cdot & X & \cdot & X & \cdot & X & X & A \\ \cdot & X & X & A & \cdot & X & X & A & \cdot & X \\ X & B & X & A & \cdot & X & X & A & X & B \\ X & B & \cdot & X & \cdot & X & \cdot & X & X & B, \\ X & B & X & B & X & A & X & B & X & B \\ \cdot & X & X & B & \cdot & X & X & B & \cdot & X \\ X & C & \cdot & X & X & B & \cdot & X & X & C \\ X & C & X & C & \cdot & X & X & C & X & C \\ X & C & X & C & X & C & X & C & X & C \end{array}\quad \cdots \right\}$$

is also a $(CF : CS)AL$ being generated by the following $(CF : CS)AG$, $G = (V, I, P, S)$ where $V = V_1 \cup V_2$, $V_1 = \{S\}$, $V_2 = \{R, J\}$, $I = \{A, B, C, \cdot, X\}$ and $P = P_1 \cup P_2 \cup P_3$ where $P_1 = \{S \to R \bigoplus (J \ominus S) \bigoplus R\}$,

$$P_2 = \left\{ \begin{array}{l} L_J = \left\{ \begin{pmatrix} \cdot & X \\ \cdot & X \\ \cdot & X \end{pmatrix}^n \Big/ n \geq 1 \right\} \\ L_R = \left\{ \begin{array}{c} (X \;\; A)_n \\ \cdot \;\; X \\ (X \;\; B)_n \\ \cdot \;\; X \\ (X \;\; C)_n \end{array} \Big/ n \geq 1 \right\} \end{array} \right\},$$

$$P_3 = \left\{ S \to \begin{array}{cccccccc} X & A & \cdot & X & \cdot & X & X & A \\ X & A & \cdot & X & \cdot & X & X & A \\ \cdot & X & \cdot & X & \cdot & X & \cdot & X \\ X & B & X & A & X & A & X & B \\ X & B & \cdot & X & \cdot & X & X & B \\ \cdot & X & X & B & X & B & \cdot & X \\ X & C & \cdot & X & \cdot & X & X & C \\ X & C & X & C & X & C & X & C \end{array} \right\}.$$

□

Theorem 3.4. *There exist two languages L_1 and L_2 in $(CS : CS)AL$ and a context free trajectory T such that $L = L_1 \sqcup\sqcup_T L_2$ is a $(CS : CS)AL$.*

Proof. Let us consider a $(CS : CS)AL$ L_1 consisting of pictures of the form

$$\left\{ \begin{array}{ccc} \cdot & X & \cdot \\ X & X & \cdot \\ \cdot & \cdot & \cdot \end{array}, \begin{array}{ccc} \cdot & X & \cdot \\ \cdot & X & \cdot \\ X & X & \cdot \\ X & X & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{array}, \begin{array}{cccccc} \cdot & \cdot & X & X & \cdot & \cdot \\ X & X & X & X & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{array}, \begin{array}{ccccccccc} \cdot & \cdot & \cdot & X & X & X & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & X & X & X & \cdot & \cdot & \cdot \\ X & X & X & X & X & X & \cdot & \cdot & \cdot \\ X & X & X & X & X & X & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{array} \right\}$$

generated by the following $(CS : CS)AG$, $G_1 = (V_1, I, P_1, S)$ where $V_1 = V_{11} \cup V_{12}$, $V_{11} = \{S\}$, $V_{12} = \{A_1, B_1, C_1\}$, $I = \{\cdot, X\}$, $P_1 = P_{11} \cup P_{12} \cup P_{13}$, $P_{11} = \{S \to$

$A_1 \oplus S \oplus B_1 \oplus C_1, (C_1 \oplus B_1) \to (B_1 \oplus C_1), S \to \Lambda\}$,

$$P_{12} = \left\{ L_{A_1} = \left\{ \begin{matrix} (\cdot)_n \\ (X)_n / n \geq 1 \\ (\cdot)_n \end{matrix} \right\}; L_{B_1} = \left\{ \begin{matrix} (X)_{2n} \\ (\cdot)_n \end{matrix} / n \geq 1 \right\}; L_{C_1} = \{(\cdot)_{3n} / n \geq 1\} \right\},$$

$P_{13} = \phi$. Let us consider another $(CS : CS)AL$, L_2 consisting of a set of pictures of the form

$$\left\{ \begin{matrix} \cdot & X & \cdot \\ \cdot & X & X, \\ \cdot & \cdot & \cdot \end{matrix} \quad \begin{matrix} \cdot & \cdot & X & X & \cdot & \cdot \\ \cdot & \cdot & X & X & X & X, \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{matrix} \quad \begin{matrix} \cdot & X & \cdot \\ \cdot & X & \cdot \\ \cdot & X & X \\ \cdot & X & X, \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{matrix} \quad \begin{matrix} \cdot & \cdot & \cdot & X & X & X & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & X & X & X & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & X & X & X & X & X & X \\ \cdot & \cdot & \cdot & X & X & X & X & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{matrix} \right\}$$

generated by the following $(CS : CS)AG$, $G_2 = (V_2, I, P_2, S)$ where $V_2 = V_{21} \cup V_{22}$, $V_{21} = \{S'\}$, $V_{22} = \{A_2, B_2, C_2\}$, $I = \{\cdot, X\}$, $P_2 = P_{21} \cup P_{22} \cup P_{23}$, $P_{21} = \{S' \to A_2 \oplus S' \oplus B_2 \oplus C_2, (C_2 \oplus B_2) \to (B_2 \oplus C_2), S \to \Lambda\}$,

$$P_{22} = \left\{ L_{A_2} = \{(\cdot)_{3n} / n \geq 1\}, L_{B_2} = \left\{ \begin{matrix} (X)_{2n} \\ (\cdot)_n \end{matrix} / n \geq 1 \right\}, L_{C_2} = \left\{ \begin{matrix} (\cdot)_n \\ (X)_n / n \geq 1 \\ (\cdot)_n \end{matrix} \right\} \right\},$$

$P_{23} = \phi$. Let us consider a trajectory $T_C = \{r^n u^n / n \geq 1\}$. Then the shuffle of L_1 and L_2 over T_C is given by

$$L = L_1 \sqcup\!\sqcup_{T_C} L_2$$
$$= \left\{ \begin{matrix} \cdot & X & \cdot & \cdot & X & \cdot \\ X & X & \cdot & \cdot & X & X, \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{matrix} \quad \begin{matrix} \cdot & \cdot & X & X & \cdot & \cdot & \cdot & \cdot & X & X & \cdot & \cdot \\ X & X & X & X & \cdot & \cdot & \cdot & \cdot & X & X & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{matrix} \right\}$$

which is generated by the following $(CS : CS)AG$ $G_3 = (V_3, I, P_3, S)$ where $V_3 = V_{31} \cup V_{32}$, $V_{31} = \{S, S', S''\}$, $V_{32} = V_{12} \cup V_{22}$, $I = \{\cdot, X\}$, $P_3 = P_{31} \cup P_{32} \cup P_{33}$, $P_{31} = P_{11} \cup P_{21} \cup \{S'' \to S \oplus S'\}$, $P_{32} = P_{12} \cup P_{22}$, $P_{23} = \phi$. □

Theorem 3.5. *There exist two languages L_1 in $(CS : CS)AL$ and L_2 in $(CS : R)AL$ and a context free trajectory T such that $L = L_1 \sqcup\!\sqcup_T L_2$ is a $(CS : CS)AL$.*

Proof. Let L_1 be the same $(CS : CS)AL$ (first language) as considered in Theorem 3.4 that can be generated by the same $(CS : CS)AG$.

Let us consider a $(CS : R)AL$ L_2 consisting of pictures of the form

$$L_2 = \left\{ \begin{matrix} a & a & b \\ c & c & d, \\ e & e & f \end{matrix} \quad \begin{matrix} a & b & b \\ c & d & d, \\ e & f & f \end{matrix} \quad \begin{matrix} a & a & b \\ a & a & b \\ c & c & d \\ c & c & d, \\ e & e & f \\ e & e & f \end{matrix} \quad \ldots \right\}$$

generated by the following $(CS : R)AG$, $G_2 = (V_2, I', P_2, S')$ where $V_2 = V_{21} \cup V_{22}$,

$V_{21} = \{S'\}$, $V_{22} = \{A_2, B_2, C_2\}$, $I' = \{a, b, c, d, e, f\}$, $P_2 = P_{21} \cup P_{22} \cup P_{23}$, $P_{21} = \{S' \to A_2 \ominus S' \ominus B_2 \ominus C_2, (C_2 \ominus B_2) \to (B_2 \ominus C_2), S' \to \Lambda\}$, $P_{22} = \{L_{A_2} = \{a^n b^m / m, n \geq 1\}, L_{B_2} = \{c^n d^m / m, n \geq 1\}, L_{C_2} = \{e^n f^m / m, n \geq 1\}\}$, $P_{23} = \phi$.

Let us consider the trajectory $T_C = \{r^n u^n / n \geq 1\}$. Then, the shuffle of L_1 and L_2 over T_C is given by

$$L = L_1 \sqcup\!\sqcup_{T_C} L_2$$
$$= \left\{ \begin{array}{cccccc} \cdot & X & \cdot & a & a & b \\ X & X & \cdot & c & c & d \\ \cdot & \cdot & \cdot & e & e & f \end{array}, \begin{array}{cccccc} \cdot & X & \cdot & a & b & b \\ X & X & \cdot & c & d & d \\ \cdot & \cdot & \cdot & e & f & f \end{array}, \begin{array}{cccccc} \cdot & X & \cdot & a & a & b \\ \cdot & X & \cdot & a & a & b \\ X & X & \cdot & c & c & d \\ X & X & \cdot & c & c & d \\ \cdot & \cdot & \cdot & e & e & f \\ \cdot & \cdot & \cdot & e & e & f \end{array}, \ldots \right\}$$

and is generated by the following $(CS : CS)AG$ $G = (V, I'', P, S'')$ where $V = (V_1 \cup \{S''\})$, $I'' = I \cup I'$, $P = P_1 \cup P_2 \cup P_3$, $P_1 = P_{11} \cup P_{21} \cup \{S'' \to S \obar S'\}$, $P_2 = P_{12} \cup P_{22}$, $P_3 = \phi$. □

4. Application to kolam pattern generation

"Kolam" refers to decorative artwork drawn on the floor with the kolam drawing generally starting with a certain number pattern of points and curly lines going around these points. A classification of kolam patterns based on their generation by different array grammars was considered by Siromoney et al. [10]. The approach for generation of kolam patterns adopts the technique referred to as Narasimhan's method of kolam generation (Siromoney et al., 1974). The kolam patterns are coded as rectangular arrays of symbols. The array languages generated by array grammars with shuffle operation on finite arrays over trajectories have strong connection with kolam patterns. As an illustration, we consider the following example [10].

Example 4.1. Let us consider a language L_1 in $(R : R)AL$ and T a context-free regular language such that $L_1 \sqcup\!\sqcup_T L_1$ is a $(CF : R)AL$.

Let L_1 be the set of kolam patterns generated by the following $(R : R)AG$, $G = (V, I, P, S)$ where $V = V_1 \cup V_2$, $I = \{\triangledown, \lozenge, \blacktriangledown, \blacksquare, \square, \Delta, B\}$ (B stands for blank), $V_1 = \{S\}$, $V_2 = \{E, F\}$, $P = P_1 \cup P_2$ with $P_1 = \{S \to (S \obar E) \ominus F\}$,

$$P_2 = \left\{ \begin{array}{cccccc} B & B & B & B & B & \triangledown \\ B & B & B & B & B & \lozenge \\ B & B & B & B & \square & \blacktriangledown \\ B & B & B & B & \Delta & \blacktriangledown \\ B & B & \square & \Delta & B & \blacksquare \\ \triangledown & \lozenge & \blacktriangledown & \blacktriangledown & \blacksquare & \lozenge \end{array} \right\},$$

$$L_E = \left\{ \begin{array}{cc} \triangledown & B \\ \Diamond & B \\ \multicolumn{2}{c}{\left(\begin{array}{cc} \blacktriangledown & \blacksquare \\ \blacktriangledown & \Delta \\ \blacksquare & \blacktriangledown \\ \Diamond & \blacktriangledown \end{array} \right)_n} \end{array} \quad n \geq 1 \right\} \cup \left\{ \begin{array}{cc} B & \triangledown \\ B & \Diamond \\ \multicolumn{2}{c}{\left(\begin{array}{cc} \blacksquare & \blacktriangledown \\ \Delta & \blacktriangledown \\ \blacktriangledown & \blacksquare \\ \blacktriangledown & \Diamond \end{array} \right)_n} \\ \square & \blacktriangledown \\ \Delta & \blacktriangledown \end{array} \quad n \geq 1 \right\},$$

$$L_F = \left\{ \begin{array}{cc} \triangledown & \Diamond \\ B & B \end{array} \begin{pmatrix} \blacktriangledown & \blacktriangledown & \blacksquare & \Diamond \\ \square & \Delta & \blacksquare & \blacksquare \end{pmatrix}_n \begin{array}{cc} B & \square \\ \square & \Delta \end{array} \Big/ n \geq 1 \right\}$$

$$\cup \left\{ \begin{array}{cccc} B & B & \Delta & \Diamond \\ \triangledown & \Diamond & \blacktriangledown & \blacktriangledown \end{array} \begin{pmatrix} \blacktriangledown & \blacktriangledown & \Delta & \Diamond \\ \Delta & \Diamond & \blacktriangledown & \blacktriangledown \end{pmatrix}^n \begin{array}{cc} B & \Delta \\ \Delta & \Diamond \end{array} \Big/ n \geq 1 \right\}.$$

The set of instructions is:

(1) Join a $\blacktriangledown$ dot to the nearest $\blacktriangledown$ dot.
(2) Join a $\triangledown$ dot to the nearest $\triangledown$ dot.
(3) Join a $\Diamond$ dot to the nearest $\square$ dot.
(4) Join a $\square$ dot to the nearest Δ dot.
(5) Join a Δ dot to the nearest $\blacksquare$ dot.
(6) Join a $\blacksquare$ dot to the nearest $\Diamond$ dot.

A member of L_1 is given in Figure 1. Let us consider a context free trajectory

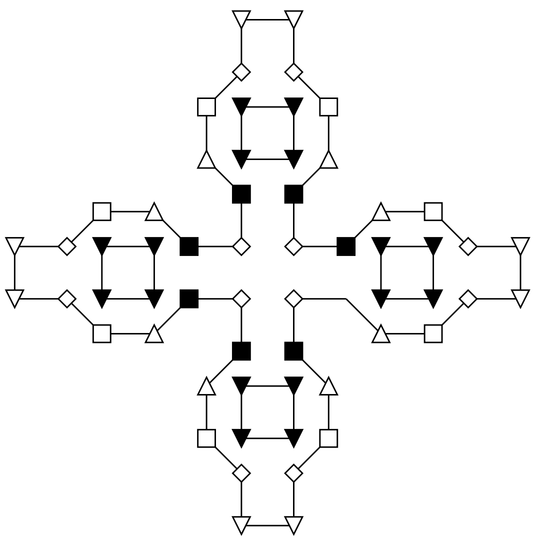

Figure 1. A kolam pattern.

$T_C = \{r^n u^n / n \geq 1\}$. Now the shuffling of L_1 over the trajectory T_C results in a $(CF : R)AL$ and can be generated by the following $(CF : R)AG$, $G_2 = (V, I, P, S)$, $V = V_1' \cup V_2)$, $I = \{\triangledown, \Diamond, \blacktriangledown, \blacksquare, \square, \Delta, B\}$ (B stands for blank and in the figures drawn the corresponding entry is left blank), $V_1' = \{S, S_1\}$, $V_2 = \{E, F\}$,

$P = P_1' \cup P_2$ where $P_1' = \{S \to (S_1 \varobar S_1), S_1 \to (S \varobar E) \ominus F\}$. One generated member of this pattern is given in Figure 2.

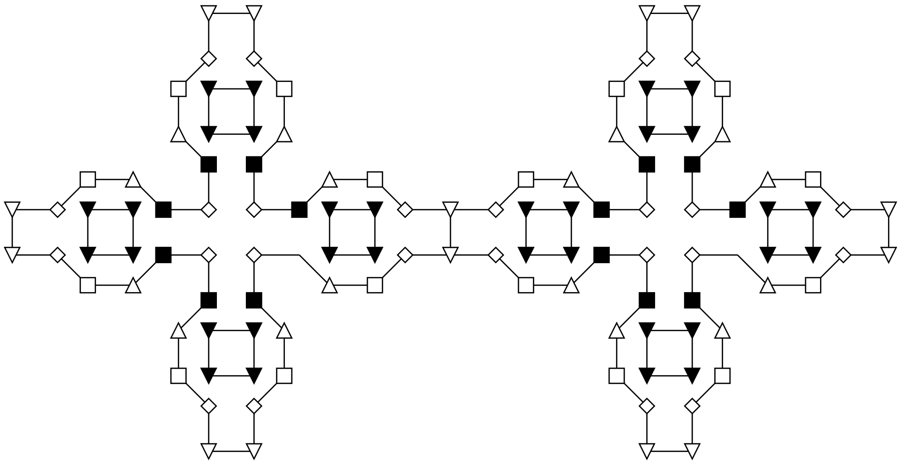

Figure 2. Extended kolam pattern.

5. Conclusion

The class of grammars with array rewriting rules is a powerful tool to describe interesting pictures. In this paper we combine the array languages generated by array grammars with shuffle operation on finite arrays over trajectories and study the picture generation of various classes of Siromoney array languages such as $(R : R)AG$, $(R : CF)AG$, $(CF : R)AG$, $(CF : CF)AG$, $(CS : R)AG$, $(CS : CS)AG$ and $(CF : CS)AG$. These languages have strong connection with kolam patterns. Our future interest is to associate the shuffle on trajectories with array automata and tiling patterns.

References

[1] H. Geetha, D. G. Thomas, T. Kalyani and A. S. Prasanna Venkatesan, *Shuffle on trajectories over finite array languages*, in: J. K. Aggarwal et al. (eds.), Combinatorial Image Analysis, Lecture Notes in Computer Science **6636**, 2011, 261–274.

[2] D. Giammarresi and A. Restivo, *Two-dimensional languages*, in: A. Salomaa and G. Rozenberg (eds.), Handbook of Formal Languages **3**, Springer-Verlag, 1997, 215–267.

[3] P. Helen Chandra, C. Martin-Vide, K. G. Subramanian, D. L. Van and P. S. P. Wang, *Parallel contextual array grammars and trajectories*, in: C. H. Chen and P. S. P. Wang (eds.), Handbook of Pattern Recognition and Computer Vision, 3rd ed., World Scientific, 2004, 55–70.

[4] K. Krithivasan and R. Siromoney, *Array automata and operations on array languages*, Int. J. Comput. Math. **4** (1974), 3–30.

[5] A. Mateescu, G. Rozenberg and A. Salomaa, *Fundamental study – shuffle on trajectories: Syntactic constraints*, Theor. Comput. Sci. **197** (1998), 1–56.

[6] A. Rosenfeld and R. Siromoney, *Picture languages – a survey*, Languages of Design **1** (1993) 229–244.

[7] A. Salomaa, *On grammars with restricted use of productions*, Ann. Acad. Sci. Fenn. Ser. A.I. **454** (1969), 1–32.
[8] G. Siromoney, R. Siromoney and K. Krithivasan, *Abstract families of matrices and picture languages*, Computer Graphics and Image Processing **1** (1972), 234–307.
[9] G. Siromoney, R. Siromoney and K. Krithivasan, *Picture languages with array rewriting rules*, Information and Control **22** (1973), 447–470.
[10] G. Siromoney, R. Siromoney and K. Krithivasan, *Array grammars and kolam*, Computer Graphics and Image Processing **3** (1974), 63–82.
[11] K. G. Subramanian, L. Revathi and R. Siromoney, *Siromoney array grammars and applications*, Int. J. Pattern Recogn. **3** (1989), 333–351.

D. K. Sheena Christy, Department of Mathematics, SRM University, Kattankulathur, Chennai – 603 203, India
e-mail: `sheena.lesley@gmail.com`

Vedhanayagam Masilamani, Department of Computer Science and Engineering, IIITD&M Kanchipuram, Chennai – 600 036, India
e-mail: `masila@iiitdm.ac.in`

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`