Cooperating distributed context-free hexagonal array grammar systems with permitting contexts

SujathakumariK[1*], Dersanambika K.S[2†]
[1] Department of Mathematics
S.N College, Punalur, Kerala 691305, India
[2] Department of Mathematics
Fatima Mata National College, Kollam 691 001, India

Abstract

In this paper we associate permitting symbols with rules of Grammars in the components of cooperating distributed context-free hexagonal array grammar systems as a control mechanism and investigating the generative power of the resulting systems in the terminal mode. This feature of associating permitting symbols with rules when extended to patterns in the form of connected arrays also requires checking of symbols, but this is simpler than usual pattern matching. The benefit of allowing permitting symbols is that it enables us to reduce the number of components required in a cooperating distributed hexagonal array grammar system for generating a set of picture arrays.

Subject Classification: 68RXX
Keywords: Hexagonal arrays, Cooperating hexagonal array grammar systems, Generative power

1 Introduction

In the Context of image analysis and image processing a variety of generative models for digitalized picture arrays in the two dimensional plane have been proposed [10]. Out of the different techniques adopted for various models, grammar based techniques utilize the rich theory of formal grammars and languages and develop array grammars generating two dimensional languages whose elements are picture arrays. There are two distinct types of array grammars, isometric array grammars and non-isometric array grammars. Since application of rewriting rule can increase or decrease the length of the rewritten part, the dimension of rewritten sub array can change in the case of non-isometric grammars but application of such a rule is shape preserving in the case of isometric grammars due to the fact that the left and right sides of an array rewriting rule is geometrically identical.

*email-nksujathakumari@gmail.com
†email-dersanapdf@yahoo.com

In order to handle more context with rewriting systems, a system with several components is composed and defined a cooperation protocol for these components to generate a common sentential form. Such devices are known as cooperating distributed (CD)grammar systems [3]. Components are represented by grammars or other rewriting devices, and the protocol for mutual cooperation modifying the common sentential form according to their own rules. A variety of string grammar system models [3] have been introduced and studied in the literature. Rudolf Freund extended the concept of grammar system to arrays [2] by introducing array grammar system and further J. Dassow, R. Freund and Gh. Păun elaborated the power of cooperation in array grammar system (cooperating array grammar system) for various non-context-free sets of arrays which can be generated in a simple way by cooperating array grammar systems and simple picture description [1]. They also proved that the cooperation increases the generative capacity even in the case of systems with regular array grammar components.

Different kinds of control mechanism that are added to component grammars for regulated rewriting rules have been considered in string grammar systems and such control devices are known to increase the generative power of the grammar in many cases [1]. Random context grammar is viewed as one of the prototype mechanism in which components grammars that permit or forbid the application of a rule based on the presence or absence of a set of symbols.

Hexagonal arrays and hexagonal patterns are found in the literature on picture processing and image analysis. The class of Hexagonal kolam array language (HKAL) was introduced by Siromoneys [9]. The class of Hexagonal array language was introduced by Subramanian. The class of local and recognizable picture languages were introduced by Dersanambika et.al. [6]. Recently we extended cooperative distributed grammar system to Hexagonal arrays and different capabilities of the system are studied [8].

In this paper we associate permitting symbols with rules of the grammar in the components of cooperating distributed context-free hexagonal array grammar systems as a control mechanism and investigating the generative power of the resulting systems in the terminal mode. This feature of associating permitting symbols with rules when extended to patterns in the form of connected arrays also requires checking of symbols, but this is simpler than usual pattern matching. The benefit of allowing permitting symbols is that it enables us to reduce the number of components required in a cooperating distributed hexagonal array grammar system for generating a set of picture arrays.

2 Preliminaries and definitions

Let V be a finite non-empty set of symbols. The set of all hexagonal arrays made up of elements of V is denoted by V^{**H}. The size of the hexagonal array is defined by the parameters LU(left upper), LL(left lower), RU(right upper), RL(right lower), U(upper), L(lower) as shown in Figure 1. For $X \in V^{**H}$ the length of the left upper side of X is denoted by $|X|_{LU}$. Similarly we define $|X|_{RU}$,$|X|_{R}$,$|X|_{RL}$,$|X|_{LL}$ and $|X|_{L}$.

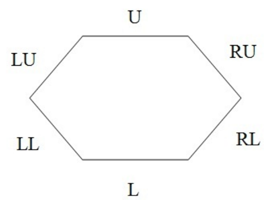

Figure 1

Definition 2.1. *An isometric hexagonal array grammar is a construct* $G = (N, T, S, P, \#)$, *where* N *and* T *are disjoint alphabets of non terminals and terminals respectively,* $S \in N$ *is the start symbol,* $\#$ *is a special symbol called blank symbol and* P *is a finite set of rewriting rules of the form* $\alpha \to \beta$ *where* α *and* β *are finite subpatterns of a hexagonal pattern over* $N \cup T \cup \{\#\}$ *satisfying the following conditions:*

1. *The shape of* α *and* β *are identical.*

2. α *contains at least one element of* N. *The elements of* T *appearing in* α *are not rewritten.*

3. *A non* $\#$ *symbol in* α *is not replaced by a blank symbol in* β.

4. *The application of the production* $\alpha \to \beta$ *preserves connectivity of the hexagonal array.*

For a hexagonal array grammar $G = (N, T, S, P, \#)$, *we can define* $x \Longrightarrow y$ *for* $x, y \in (N \cup T \cup \{\#\})$ *if there is a rule* $\alpha \to \beta \in P$ *such that* α *is a subpattern of* x *and* y *is obtained by replacing* α *in* x *by* β. *The reflexive closure of* $\Rightarrow$ *is denoted by* $\overset{*}{\Rightarrow}$. *The hexagonal array language generated by* G *is defined by* $L(G) = \{x \in (T \cup \{\#\})^{**H} : s \overset{*}{\Rightarrow} x\}$.

Definition 2.2. *A hexagonal array grammar is said to be context free if in the rule* $\alpha \to \beta$

1. *non* $\#$ *symbol in* α *are not replaced by* $\#$ *in* β

2. α *contain exactly one non-terminal and some occurrences of blank symbol.*

.The family of languages generated by a context free hexagonal array grammar is denoted by $CFHA$.

Definition 2.3. *A context free hexagonal array grammar is said to be regular if rules are of the form* $A \quad \# \to a \quad B$, $\# \quad A \to B \quad a$,

$$\begin{array}{cc} & A \\ \# & \end{array} \to \begin{array}{cc} & a \\ B & \end{array},\quad \begin{array}{cc} & \# \\ A & \end{array} \to \begin{array}{cc} & B \\ a & \end{array},\quad \begin{array}{cc} A & \\ & \# \end{array} \to \begin{array}{cc} a & \\ & B \end{array},$$

$$\begin{array}{cc} \# & \\ & A \end{array} \to \begin{array}{cc} B & \\ & a \end{array}, A \to a$$

The family of languages generated by a regular hexagonal array grammar is denoted by $REGHA$.

Definition 2.4. *A cooperating hexagonal array grammar system (of type* $X, X \in \{CFHA, REGHA\}$*, and degree* $n, n \geq 1$*), is a construct* $\Gamma = (N, T, S, P_1, P_2, \ldots, P_n)$*, where* N *and* T *are non-terminal and terminal alphabets respectively,* $S \in N$ *and* $P_1, P_2, \ldots, P_n$ *are finite sets of regular respectively context free rules over* $N \cup T$*.*

Definition 2.5. *Let* Γ *be a cooperating hexagonal array grammar system. Let* $x, y \in \Gamma^*$*. Then we write* $x \overset{k}{\underset{p_i}{\Rightarrow}} y$ *if and only if there are words* $x_1, x_2, \ldots, x_{k+1}$ *such that*

1. $x = x_1, y = x_{k+1}$,
2. $x_j \underset{p_i}{\Rightarrow} x_{j+1}$*, that is,* $x_j = x'_j A_j x''_j$, $x_{j+1} = x'_j W_j x''_j$, $A_j \to W_j \in P_i,\ 1 \leq j \leq k$.

Moreover, we write

$x \overset{\leq k}{\underset{P_i}{\Rightarrow}} y$ *if and only if* $x \overset{k'}{\underset{P_i}{\Rightarrow}} y$ *for some* $k' \leq k$,

$x \overset{\geq k}{\underset{P_i}{\Rightarrow}} y$ *if and only if* $x \overset{\leq k'}{\underset{P_i}{\Rightarrow}} y$ *for some* $k' \geq k$,

$x \overset{*}{\underset{P_i}{\Rightarrow}} y$ *if and only if* $x \overset{k}{\underset{P_i}{\Rightarrow}} y$ *for some* k

$x \overset{t}{\underset{P_i}{\Rightarrow}} y$ *if and only if* $x \overset{*}{\underset{P_i}{\Rightarrow}} y$ *and there is no* $z \neq y$ *with* $y \overset{*}{\underset{P_i}{\Rightarrow}} z$.

By $CD_n(X, f)$ *we denote family of hexagonal array language generated by cooperating hexagonal array grammar system consisting of at most* n *components of type* $X \in (REGHA, CFHA)$ *in the mode* f.

Definition 2.6. *A random context grammar is a quadruple* $G = (N, T, P, S)$ *where* N *is the alphabet of non-terminals,* T *is the alphabet of terminals such that* $N \cap T = \phi, V = N \cup T, S \in N$ *is the start symbol,and* P *is a finite set of productions of the form* $(A \to x, Per, For)$ *where* $A \to x$ *is a context free production,*$A \in N$ *and* $x \in V^*$*,and* $Per, For \subseteq N$*. For* $U, V \in V^*$ *and a production* $(A \to x, Per, For) \in P$*, the relation* $Av \Rightarrow uxv$ *holds provided that* $per \subseteq alph(uv)$ *and* $alph(uv) \cap For = \phi$*. A permitting (forbidding)grammar is a random context grammar* $G = (N, T, P, S)$ *where for each production*
$(A \to x, Per, For) \in P$*, it holds that* $For = \phi$ $(Per = \phi$ *respectively)*

3 Cooperating distributed context-free hexagonal array grammar system with permitting symbols

The set of all symbols in the labeled cells of the array p is denoted by alph(p) A permitting CF hexagonal array rule is an array grammar G is of the form $(\alpha \to \beta, per)$, where $\alpha \to \beta$ is a context-free hexagonal array rewriting rule and $per \subseteq N$, where $N$is the set of non -terminals of the grammar. If $per = \phi$, then we avoid mentioning it in the rule.For any two arrays p, q and a permitting CF hexagonal array rule $(\alpha \to \beta, per)$, the array q is derived from p by replacing α in p by β provided that $per \subseteq alph(p \backslash \alpha)$. A permitting cooperating distributed context-free hexagonal array grammar systems (pCDCFHAGS) is $\Gamma = (N, T, P_1, P_2, \ldots, P_N, S)$, $n \geq 1$ where N is a finite set of non terminals,$S \in N$ is the start symbol,T is a finite set of terminals, $N \bigcap T = \phi$ and each P_i, for $1 \leq i \leq n$ is a finite set of permitting CF hexagonal array rewriting rules.

For any two hexagonal arrays p, q, we denote $p \underset{p_i}{\overset{t}{\Rightarrow}} q$ an array rewriting step performed by applying a permitting cooperating CF Hexagonal array rule in $p_i, 1 \leq i \leq n$, and by $p \underset{p_i}{\overset{*}{\Rightarrow}} q$ the transitive closure of $p \underset{p_i}{\overset{t}{\Rightarrow}} q$. Also we say that the array p derives an array q in the terminal mode or t mode and write $p \underset{p_i}{\overset{t}{\Rightarrow}} q$, if$p \underset{p_i}{\overset{*}{\Rightarrow}} q$, and there is no array s such that $q \underset{p_i}{\overset{t}{\Rightarrow}} s$ The array language generated by Γ in the t mode is defined as $L(\Gamma) = \{q \colon s = p_o \underset{p_{i_1}}{\overset{t}{\Rightarrow}} p_1 \underset{p_{i_2}}{\overset{t}{\Rightarrow}} p_2, \ldots, \underset{p_{i_m}}{\overset{t}{\Rightarrow}} p_m = q \in V^{**H}, m \geq 1, i_j \in \{1, 2, \ldots, n\}$ for $1 \leq j \leq m\}$.

Note that $i_1, i_2, \ldots, i_m$ is any sequence of symbols belonging to $\{1, 2, \ldots, n\}$ where repeated symbols are allowed. Also $pCD_n(HCFA, t)$ denote the family of array languages generated in the t mode by permitting cooperating CF hexagonal array grammar systems with at most n components

Example 3.1. *Consider the context-free hexagonal array grammars with rules $G = (\{S, A, B\}, \{a\}, P, S, \#)$ where*

$$P = \left\{ 1)\ \begin{array}{cc} & \# \\ S & \\ & \# \end{array} \Rightarrow \begin{array}{cc} & A \\ a & \\ & B \end{array},\ 2)\ \begin{array}{cc} & \# \\ A & \end{array} \to \begin{array}{cc} & A' \\ a & \end{array},\right.$$

$$\left. 3)\ \begin{array}{cc} B & \\ & \# \end{array} \to \begin{array}{cc} a & \\ & B' \end{array},\ 4)\ A' \to A,\ 5)\ B' \to B,\ 6)\ A \to a,\ 7)\ B \to a \right\}$$

$$\begin{array}{cccccc} & & & & & a \\ & & & & a & \\ & & & \cdot & & \\ & & \cdot & & & \\ & a & & & & \\ a & & & & & \\ & a & & & & \\ & & a & & & \end{array}$$

Figure 2

G generates hexagonal arrays over $\{a\}$ is in the shape of left arrow head but size of left upper arm and left lower arm are not necessarily equal.

Example 3.2. *The pCDCFHAGS $G_1 = (\{S, A, B, A', B', C, D\}, \{a\}, P_1, S)$ where P_1 consists of the following rules.*

$$1.\ S \begin{array}{c} \# \\ \# \end{array} \Rightarrow a \begin{array}{c} A \\ B \end{array},$$

$$2.\left(A \;\; ^{\#} \rightarrow a \;\; ^{A'},\ \{B\} \right),$$

$$3.\left(B \;\; _{\#} \rightarrow a \;\; _{B'}, \{A'\} \right),$$

$$4.\,(A' \rightarrow A, \{B'\}),$$

$$5.\,(B' \rightarrow B, \{A\}),$$

$$6.\,(A \rightarrow C, \{B\}),$$

$$7.\,(B \rightarrow D, \{C\}),$$

$$8. C \rightarrow a,$$

$$9. D \rightarrow a$$

generates (in the t mode) the set H_{LA} of all arrays over $\{a\}$ in the shape of a left arrow *head with left upper arm and left lower arm are equal in size (Figure 3).*

$$\begin{array}{cccc} & & & a \\ & & a & \\ & a & & \\ a & & & \\ & a & & \\ & & a & \\ & & & a \end{array}$$

Figure 3

The derivations starts with rule 1 followed by rule 2 which can be applied as the permitting symbol B is present in the array. This grows left upper arm (LU) by one cell. Then rule 3 can be applied due to the presence of permitting symbol A' and this grows left lower arm (LL) by one cell. An application of rule 4 followed by 5, again noting that the permitting symbols of the respective rules are present changes A' to A and B' to B. the repeated application of the process growing both the left upper arm and left lower arm equal in size. Rules 6 and 7 are applied changing A to C and B to D so that the derivation can be terminated by the application of rules 8 and 9 thus yielding a hexagonal array in the shape of a left arrow head with size of LU and LL are equal.

Remark 3.3. *A $pCDCFHAGS$ where the set of permitting symbols in all the components is empty, is simply a cooperating distributed CF hexagonal array system ($CDCFHAGS$)[8]. The family of array languages generated in the t mode by a $CDCFHAGS$ with at most n components is denoted by $CD_n(CFHA, t)$. If the rules in all the components are only in the form of rules of a regular array grammar, then this family is denoted by $CD_n(REGHA, t)$.*

We now show that the set H_{HF} of all $n \times n \times n (n \geq 3)$ arrays over $\{a\}$ in the form of hollow hexagonal frame. Figure 3 can be generated (in the t mode) by a $pCDCFHAGS$ with only two components.

Lemma 3.4. $H_{HF} \in pCD_2(HCFA, t)$.

Proof. The set H_{HF} is generated (in the t mode) by the $pCDCFHAGS$ $G_2 = (\{S, A, B, A', B', C, D, C', D', E, F, E', F', X, Y\}, \{a\}, P_1, P_2, S)$. The rules in the componentP_1 are given by

$$1.\ S \begin{array}{c} \# \\ \# \end{array} \Rightarrow a \begin{array}{c} A \\ B \end{array},$$

$$2.\left(A \begin{array}{c} \# \\ \ \end{array} \rightarrow a \begin{array}{c} A' \\ \ \end{array},\ \{B\}\right),$$

$$3.\left(B \begin{array}{c} \ \\ \# \end{array} \rightarrow a \begin{array}{c} \ \\ B' \end{array}, \{A'\} \right),$$

$$4.\,(A' \rightarrow A, \{B'\}),$$

$$5.\,(B' \rightarrow B, \{A\}),$$

$$6.\left(A' \ \# \rightarrow \ a \ \ C, \{B'\} \right),$$

$$7.\left(B' \ \# \rightarrow \ a \ \ D, \{C\} \right),$$

$$8.\left(C \ \# \rightarrow \ a \ \ C', \{D\} \right),$$

$$9.\left(D \ \# \rightarrow \ a \ \ D', \{C'\} \right),$$

$$10.\,(C' \rightarrow C, \{D'\}),$$

$$11.\,(D' \rightarrow D, \{C\}),$$

$$12.\left(C' \begin{array}{c} \ \\ \# \end{array} \rightarrow a \begin{array}{c} \ \\ E \end{array}, \{D'\} \right),$$

$$13.\left(D' \begin{array}{c} \# \\ \ \end{array} \rightarrow a \begin{array}{c} F \\ \ \end{array}, \{E\} \right),$$

$$14.\left(E \begin{array}{c} \ \\ \# \end{array} \rightarrow a \begin{array}{c} \ \\ E' \end{array}, \{F\} \right),$$

$$15.\left(F \begin{array}{c} \# \\ \ \end{array} \rightarrow a \begin{array}{c} F' \\ \ \end{array}, \{E'\} \right),$$

$$16.\,(E' \rightarrow E, \{F'\}),$$

17. $(F' \rightarrow F, \{E\})$,

18. $\left(\begin{array}{cc} E & \\ & \# \end{array} \rightarrow \begin{array}{cc} a & \\ & X \end{array}, \{F\} \right)$,

19. $\left(\begin{array}{cc} & \# \\ F & \end{array} \rightarrow \begin{array}{cc} & Y \\ a & \end{array}, \{X\} \right)$,

The rules in the component P_2 are given by

$$P_2 = \{X \rightarrow a, Y \rightarrow a, F \rightarrow a\}$$

$$\begin{array}{ccccccccccccc}
 & & & & a & a & a & a & a & & & & \\
 & & & a & & & & & & a & & & \\
 & & a & & & & & & & & a & & \\
 & a & & & & & & & & & & a & \\
a & & & & & & & & & & & & a \\
 & a & & & & & & & & & & a & \\
 & & a & & & & & & & & a & & \\
 & & & a & & & & & & a & & & \\
 & & & & a & a & a & a & a & & & &
\end{array}$$

Figure 4

Using t- mode of derivation, starting with the symbol S an application of rule (1) in the first component followed by rule(2) which can be applied as the permitting symbol B is present in the array, grows in the LU arm by one place. Since A' is the permuting symbol for rule (3), rule (3) can then applied which results the growth of LL arm by one place. Now the situations are ready for applying rules (4) and (5) and at this stage A' becomes A and B' becomes B. The process can be repeated and this in turn results the growth of LU and LL arms equal in length. Instead of rule (4) rule (6) is applied followed by (7),(8),(9),(10),(11) allows upper and lower arms to grow in equal length, with permitting symbols in all these rules directing the sequence of applications in the right order. If rule (12) is used instead of rule (10) and this is followed by rule (13) Right upper (RU) and Right lower (RL) arms grows equal in size and correct application of rule (18) and (19) will result in the symbol X in the RU arm and F in the lower right arm where X is at the position of right end point of RU and RL arm so that further application of productions in P_1 is not possible at any non-terminals. At this stage applying productions in P_2 and this in turn terminate the derivation yielding a hollow hexagon with its parallel arms are equal in size. □

Lemma 3.5. *1. $H_{HF} \in CD_3(CFHA, t)$*

2. $H_{HF} \notin CD_n(REHA, t)$ for $n \geq 1$

Proof follows from the result in [8].

Theorem 3.6. *1.* $CFHA = CD_1(CFHA, t) \subset pCD_1(CFHA, t)$.

2. $CD_2(CFHA, t) \subset pCD_2(CFHA, t)$.

3. $pCD_2(CFHA, 2) \setminus CD_n(REGHA, t) \neq \phi$ *for any* $n \geq 1$.

Proof. The equality follows from the results from [8]. We know that, a $pCDCFHAGS$ with empty set of permitting symbols associated with the rules is same as a cooperating distributed CF hexagonal array grammar system and so $CD_1(CFHA, t) = pCD_1(CFHA, t)$ if $per = \phi$. Examples (1) and (2) illustrated the fact that the set H_{LA} of all hexagonal arrays over a in the shape of left arrow head with left upper arm and left lower arm with equal size is generated by the $pCD_1CFHAGS$ with only one component and working in the t-mode and hence the inclusion is proper which proves (1). Similar arguments for inclusion in statement(2) are hold. From the proof of the lemma(1) it is very clear that under the strict application of the derivation rules with the respective permitive symbols generate a hollow hexagon with parallel arms equal in size. Such a generation is not possible in $CD_2CFHAGS$ since $per = \phi$ and incorrect application of terminating rule will leads to non-completion of the hollow hexagon with parallel arms equal in size. Thus $CD_2(CFHA, t) \subset pCD_2(CFHA, t)$.

Consider the array languages generated by $pCDCFHAGS$ in example (1) and in the proof of lemma (1). It can be seen that generating the arrow head of patterns of the language, we should require two growing heads at the same time. But in the $CDREGHAGS$ with any number of components the array rules contains only one growing head. So the same language cannot be generated by $CD_n(REGHA, t)$ and this proves (3). $\square$

To show the power of the cooperating hexagonal array grammar system with the array rules controlled by permitting symbols, consider the following example of a language of set of all hexagons with parallel arms are equal in length over a one letter alphabet.

Example 3.7. *Consider the* $pCDCFHAGS$
$G_4 = (\{S, S', A, B, A', B', C, D, C', D', E', F', G, H, I, I', J, J', K, K', L, L', M, M', N, N'O, R, T, T', X\}, \{a, b\}, p_1, p_2, S)$.
The rules in the component p_1 *are*

$$1)\ S \quad \begin{matrix} \# \\ \# \end{matrix} \quad \# \ \rightarrow \ a \quad \begin{matrix} A \\ B \end{matrix} \quad S' \ ,$$

$$2) \left(A \quad {}^{\#} \ \rightarrow \ a \quad {}^{A'} \ , \{B\} \right),$$

3) $\left(\begin{array}{cc} B & \\ & \# \end{array} \to \begin{array}{cc} a & \\ & B' \end{array}, \{A'\}\right),$

4) $(A' \to A, \{B'\}),$

5) $(B' \to B, \{A\}),$

6) $\left(\begin{array}{cc} A' & \# \end{array} \to \begin{array}{cc} a & C \end{array}, \{B'\}\right),$

7) $\left(\begin{array}{cc} B' & \# \end{array} \to \begin{array}{cc} a & D \end{array}, \{C\}\right),$

8) $\left(\begin{array}{cc} C & \# \end{array} \to \begin{array}{cc} a & C' \end{array}, \{D\}\right),$

9) $\left(\begin{array}{cc} D & \# \end{array} \to \begin{array}{cc} a & D' \end{array}, \{C'\}\right),$

10) $(C' \to C, \{D'\}),$

11) $(D' \to D, \{C\}),$

12) $\left(\begin{array}{ccc} & C & \# \\ \# & & \end{array} \to \begin{array}{ccc} & a & a \\ E' & & \end{array}, \{D\}\right),$

13) $\left(\begin{array}{ccc} \# & & \\ & D & \# \end{array} \to \begin{array}{ccc} F' & & \\ & a & a \end{array}, \{E'\}\right),$

14) $(E' \to E, \{F'\}),$

15) $(F' \to F, \{E\}),$

16) $\left(\begin{array}{cc} & E \\ \# & \end{array} \to \begin{array}{cc} & a \\ E' & \end{array}, \{F\}\right),$

17) $\left(\begin{array}{cc} \# & \\ & F \end{array} \to \begin{array}{cc} F' & \\ & a \end{array}, \{E'\}\right),$

18) $\left(\begin{array}{ccc} E' & & H \\ & F & \end{array} \to \begin{array}{ccc} a & & T \\ & a & \end{array}, \{S'\}\right),$

19) $\left(\begin{array}{ccc} & \# & \\ S' & & \# \\ & \# & \end{array} \to \begin{array}{ccc} & G & \\ a & & S' \\ & H & \end{array}, \{T\}\right),$

20) $\left(T \quad \begin{matrix} \# \\ \# \end{matrix} \rightarrow a \quad \begin{matrix} I \\ J \end{matrix} \quad T', \{S'\} \right),$

21) $T \quad \begin{matrix} \# \\ \# \end{matrix} \rightarrow a \quad \begin{matrix} I \\ J \end{matrix} \quad ,$

22) $\left(G \quad \begin{matrix} \# \\ \ \end{matrix} \rightarrow a \quad \begin{matrix} G \\ \ \end{matrix} \quad , \{H\} \right),$

23) $\left(H \quad \begin{matrix} \ \\ \# \end{matrix} \rightarrow a \quad \begin{matrix} \ \\ H \end{matrix} \quad , \{G\} \right),$

24) $\left(I \quad \begin{matrix} \# \\ \ \end{matrix} \rightarrow a \quad \begin{matrix} I' \\ \ \end{matrix} \quad , \{J\} \right),$

25) $\left(J \quad \begin{matrix} \ \\ \# \end{matrix} \rightarrow a \quad \begin{matrix} \ \\ J' \end{matrix} \quad , \{I'\} \right),$

26) $(I' \rightarrow I, \{J'\}),$

27) $(J' \rightarrow J, \{I\}),$

28) $I \quad \begin{matrix} a \\ \ \end{matrix} \quad \# \rightarrow a \quad \begin{matrix} a \\ \ \end{matrix} \quad K \ ,$

29) $J \quad \begin{matrix} \ \\ a \end{matrix} \quad \# \rightarrow a \quad \begin{matrix} \ \\ a \end{matrix} \quad L \ ,$

30) $(\ K \quad \# \rightarrow a \quad K', \{L\}),$

31) $(\ L \quad \# \rightarrow a \quad L', \{K'\}),$

32) $(K' \rightarrow K, \{L'\}),$

33) $(\ L' \quad \# \rightarrow L, \{K\}),$

34) $\left(K \quad \begin{matrix} \ \\ \# \end{matrix} \rightarrow a \quad \begin{matrix} \ \\ M' \end{matrix} \quad , \{L\} \right),$

35) $\left(L \quad \begin{matrix} \# \\ \ \end{matrix} \rightarrow a \quad \begin{matrix} N' \\ \ \end{matrix} \quad , \{M'\} \right),$

36) $(M' \rightarrow M, \{N'\}),$

37) $(N' \rightarrow N, \{M\})$,

38) $\left(\begin{array}{cc} M & \\ & \# \end{array} \rightarrow \begin{array}{cc} a & \\ & M' \end{array}, \{N\} \right)$,

39) $\left(\begin{array}{cc} & \# \\ N & \end{array} \rightarrow \begin{array}{cc} & N' \\ a & \end{array}, \{M'\} \right)$,

40) $\left(\begin{array}{cc} & \# \\ N & \end{array} \rightarrow \begin{array}{cc} & N' \\ a & \end{array}, \{M'\} \right)$,

41) $\left(\begin{array}{cc} & M' \\ N & \end{array} \rightarrow \begin{array}{cc} & N' \\ a & \end{array}, \{M'\} \right)$,

42) $\left(\begin{array}{ccc} & \# & \\ T & & \# \\ & \# & \end{array} \rightarrow \begin{array}{ccc} & O & \\ a & & a \\ & R & \end{array}, \{X\} \right)$,

43) $\left(\begin{array}{cc} R & \\ & \# \end{array} \rightarrow \begin{array}{cc} a & \\ & R \end{array}, \{O\} \right)$,

44) $\begin{array}{ccc} & a & \\ T & & \# \\ & a & \end{array} \rightarrow \begin{array}{ccc} & a & \\ a & & T \\ & a & \end{array}$,

45) $\begin{array}{ccc} & E & \\ S' & & \# \\ & F & \end{array} \rightarrow \begin{array}{ccc} & a & \\ a & & T \\ & a & \end{array}$

The rules in the component p_2 are given by

$$P_2 = \{G \rightarrow a, H \rightarrow a, O \rightarrow a, T \rightarrow a, X \rightarrow a, R \rightarrow a\}$$

$$\begin{array}{ccccccccccccccccc}
 & & & & a & & a & & a & & a & & a & & & & \\
 & & & a & & a & & a & & a & & a & & a & & & \\
 & & a & & a & & a & & a & & a & & a & & a & & \\
 & a & & a & & a & & a & & a & & a & & a & & a & \\
a & & a & & a & & a & & a & & a & & a & & a & & a \\
 & a & & a & & a & & a & & a & & a & & a & & a & \\
 & & a & & a & & a & & a & & a & & a & & a & & \\
 & & & a & & a & & a & & a & & a & & a & & & \\
 & & & & a & & a & & a & & a & & a & & & &
\end{array}$$

Figure 5

Starting with S, repeated application of the first five rules of the component P_1 in this order generate $<$ a having equal arms, with LU arm having non-terminal A and LL arm having non-terminal B. Once two rules $(A'\# \rightarrow aC, \{B'\})$ and $(B'\# \rightarrow aD, \{C\})$ of component P_1 are used, the generations of these two arms end with terminal a and then *starts the generations of upper and lower arms of the hexagon using rules (6) to (11). Then the right application of rule (12) to (17) then (18), (19), ..., (45) subjected to the permitting symbols will result to a hexagonal picture and finally by the application of rules in P_2, we get the required hexagon over the one letter alphabet $\{a\}$ as in Figure 5.*

In the Siromoney matrix grammar (9) rectangular arrays are generated in two phases; one in horizontal and the other in vertical. Further it was extended by associating a finite set of rules in the second phase of generation with each table having either right linear non-terminal rules of the form $A \to B$ or right-linear terminal rules of the form $A \to a$ and such array languages are denoted by $TRML$ and $TCFML$ and we have a well known result $TRML \subset TCFML, RML \subset CFML \subset TCFML$ (refer 11). Correspondingly it can be established for hexagonal arrays. Here we compare $pCDCFHAGS$ with these classes.

Theorem 3.8. $pCD_3(CFHA, t) \backslash TCFML \neq \phi$.

Proof. Consider the $pCDCFHAGS$
$G_5 = (\{S, A, B, C, A', B', C', X, Y, Z\}, \{a, b, c\}, p_1, p_2, p_3, S)$
where the components are

$$P_1 = \left\{ 1)\ \begin{array}{ccc} & \# & \\ S & & \# \\ & \# & \end{array} \to \begin{array}{cc} & A \\ X & \\ & B \end{array}, 2) \left(\begin{array}{cc} & \# \\ A & \end{array} \to \begin{array}{cc} & A' \\ Y & \end{array}, \{B\} \right), \right.$$

$$3) \left(\begin{array}{cc} B & \\ & \# \end{array} \to \begin{array}{cc} Y & \\ & B' \end{array}, \{A'\} \right), 4)\, (A' \to A, \{B'\}), 5)\, (B' \to B, \{A\}),$$

$$\left. 6)\, (A' \to Y, B' \to Y, Z' \to Z, C' \to C) \right\}$$

$$P_2 = \left\{ Y \;\; \# \;\; \# \to a \;\; b \;\; Z,\ X \;\; \# \;\; \# \to a \;\; a \;\; c, Z \to b \;\; Z, \right.$$
$$\left. C \to a \;\; C' \right\}$$

$$P_3 = \left\{ Z \to X \;\; a, C \to a \right\}$$

$$\begin{array}{ccccccccccccc}
 & & & & a & & b & & b & & a & & \\
 & & & a & & b & & b & & a & & & \\
 & & a & & b & & b & & a & & & & \\
 & a & & b & & b & & a & & & & & \\
a & & a & & a & & a & & & & & & \\
 & a & & b & & b & & a & & & & & \\
 & & a & & b & & b & & a & & & & \\
 & & & a & & b & & b & & a & & & \\
 & & & & a & & b & & b & & a & &
\end{array}$$

Figure 6

Except the rules $Z' \to Z, C' \to C$ in P_1 generates the LU and LL arm with a middle marker X and the symbols Y above and below X in both arms and which are equal in number. The first two rules of P_2 changes the symbols in the left most cell in to a.Then the remaining rules of P_2 and the last two rules of P_1 ($Z^, \to Z$,$C^, \to C$) and the rules of the component P_3 generate the arms such that each cell in the middle arm in the horizontal direction is made up of $a's$ and all other cells above and below are made up of $b's$ except the leftmost arms.The generation of the cells finally terminates, yielding the rightmost cells are rewritten by $a's$. Thus a hexagonal array in the shape of a left arrowhead (describing as Figure(6) is generated).If we treat $'b'$ as blank, such arrays cannot be in $TCFML$ and which in turn proves that $pCD_3(HCFA, t) \setminus TCFML \neq \phi$. □

Conclusion. In this paper, the picture array generating power of cooperating CF hexagonal array grammar systems endowed with permitting symbols are studied. It is seen that the control mechanism which we here used namely the 'permitting symbols' is shape preserving in picture generation and also it reduce size complexity.

References

[1] H. Bordihm, M. Holzer, in: C. Matin-Vide, F. Otto, H. Fernau (Eds.), *Random Context in Regulated Rewriting Versus Cooperating Distributed Grammar Systems*, in:Lecture Notes in Computer Science, Vol. 5159, Springer-Verlag, 2008,pp 125–136.

[2] J. Dassow, R. Freund, Gh. Păun, *Cooperating Array Grammer Systems*, J. Pattern Recognit, Artif. Intell. 9,1995, pp 1029–1053.

[3] J. Dassow, Gh. Păun, G. Rozenberg, *Grammar systems, in: G. Rozenberg, A. Salomaa (Eds)*, Vol. 2, Springer-Verlag, 1997, pp 155–213.

[4] R. Freund, *Array Grammars*, Tech. Report 15/00, XI Tarrogena Seminar in Formal Syntax and Semantics, 2000.

[5] K. G. Subramanian et. al., *On the Power of Permitting Features in Cooperating Context-Free Array Grammar System*, Discrete Applied Mathematics, Vol. 161(15),2013, pp 2328-2335.

[6] K. S. Dersanambika, K. Krithivasan, Martin-vide, K. G. Subramanian, *Local and Recognizable Hexagonal Picture Languages*, International Journal of Pattern Recognition and Artificial Intelligence, 19(7), 2012, pp 553–571.

[7] K. S. Dersanambika, K. Krithivasan, H. K. Agarwal, J. Guptha *Hexagonal Contextual Array p-Systems, Formal Models, Languages and Application*, Series in Machine Perception Artificial Intelligence 66, 2006, pp 79–96.

[8] Jismy Joseph, K. S. Dersanambika, K. Sujathakumari, *Cooperating Hexagonal Array Grammar System*, Communicated.

[9] G. Siromoney R. Siromoney, *Hexagonal Arrays and Rectangular Blocks*, Computer Graphics and Image Processing, 1976, pp 353–381.

[10] A. Rozenfield, *Formal Models for Picture Recognition*, Academic Press, New York, 1979.

[11] R. Siromoney, K. G. Subramanian, K. Rangarajan, *Parallel \ Sequential rectangular arrays with tables*, Int. J. Comput. Math.6A, 1977,pp 143-158.