Pure 2D Picture Grammars and Languages

K.G. Subramanian [a,*], Rosihan M. Ali [a], M.Geethalakshmi [b], Atulya K. Nagar [c]

[a] *School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia*

[b] *Department of Mathematics, Dr. MGR Janaki College, Chennai 600028, India*

[c] *Intelligent and Distributed Systems Laboratory, Department of Computer Science, Liverpool Hope University, Hope Park Liverpool, L16 9JD, UK*

Abstract

A new syntactic model, called pure 2D context-free grammar ($P2DCFG$), is introduced based on the notion of pure context-free string grammar. The rectangular picture generative power of this 2D grammar model is investigated. Certain closure properties are obtained. An analogue of this 2D grammar model called pure 2D hexagonal context-free grammar ($P2DHCFG$) is also considered to generate hexagonal picture arrays on triangular grids.

Key words: Pure grammars; Two-dimensional grammars; Picture languages; Rectangular arrays; Hexagonal arrays

1 Introduction

Syntactic techniques of generation of digital picture arrays have become established as one of the major areas of theoretical studies in picture analysis, basically due to the structure handling ability of the syntactic models. A number of two-dimensional (2D) rectangular and non-rectangular picture generating mechanisms such as two-dimensional grammars and automata have been

* Corresponding author

Email addresses: kgsmani1948@yahoo.com (K.G. Subramanian), rosihan@cs.usm.my (Rosihan M. Ali), gitaraghu@yahoo.com (M.Geethalakshmi), nagara@hope.ac.uk (Atulya K. Nagar).

introduced in the literature [1–4]. Two-dimensional matrix grammars [5], array grammars [6,7], tiling systems [1,8], chain-code picture grammars [9], to mention a few, are some of the picture generating devices.

Motivated by certain floor designs called "kolam" patterns [7], a 2D rectangular picture array model which we call here as Siromoney matrix grammar (SMG), was proposed by Siromoney et al. [5]. This is one of the earliest picture models, simple and easy to handle and has been widely investigated for its theoretical properties and applications [10–15]. Generation of rectangular arrays takes place in this model in two phases with a sequential mode of rewriting in the first phase generating strings of intermediate symbols and a parallel mode of rewriting these strings in the second phase to yield rectangular picture patterns. But the disadvantage of the SMG is that rectangular arrays that maintain a proportion cannot be generated. The $SMG's$ have been extended in [16] by allowing a finite set of tables of rules in the second phase of generation. Although this model has more generative power than SMG, it still cannot maintain proportion between the height and the width of the arrays generated.

A generalization of SMG, which we call here as Siromoney array grammars (SAG), has been made in [6] to overcome such a disadvantage of not maintaining proportion but again this model SAG has two phases of derivation with the first phase involving both column and row array concatenation operators $\circ$ and $\diamond$. Although this feature is helpful to maintain proportion between rows and columns of picture arrays, the disadvantage is that the column and row operators $\circ$ and $\diamond$ are not associative unlike string concatenation. This requires use of suitable parentheses in the first phase of generation of SAG in order to avoid ambiguity.

Another very general rectangular array generating model, called extended controlled tabled L array system ($ECTLAS$) was proposed in [17], incorporating into arrays the developmental type of generation used in the well-known biologically motivated L-systems [18]. Here the symbols either on the left, right, up or down borders of a rectangular array are rewritten simultaneously by equal length strings to generate rectangular picture arrays. Although this model is general enough to generate interesting rectangular pictures and avoids independent derivation phases as in SMG [5] and SAG [6], the disadvantage is that this model [17] allows rewriting only at the borders of a rectangular array.

In the Chomsky hierarchy and related types of grammars [19], the alphabet is divided into two parts: nonterminal symbols and terminal symbols. Words consisting of entirely terminal symbols are considered to be in the language generated. But in the original rewriting systems of Thue such a distinction is not made. Following this original rewriting systems of Thue, pure grammars

considered in [20,21] use only a single set of symbols which may be used as both terminal or nonterminal symbols. Pure grammars have been investigated in formal string language theory for their language generating power and other properties [22–28].

Here we introduce a new two-dimensional grammar based on pure context-free rules, called pure 2D context-free grammar ($P2DCFG$), for rectangular picture array generation. In this 2D model we allow rewriting any column or any row of the rectangular array rewritten unlike the models in [5,16,17] and we do not prescribe any priority of rewriting columns and rows as in [5,16] in which the second phase of generation can take place only after the first phase is over. We compare the generative power of the new 2D model with other models considered in the literature [5,8,16,17]. Certain closure properties of this 2D model are also obtained.

It is known [19] that controlling the derivation in string grammars by a regular control language generally does not increase the generative power but here the generative power increases when we associate a regular control language with a $P2DCFG$. We also indicate possible application of $P2DCFG$ to generate pictures with complex primitives via the notion of interpretation. A preliminary version of this model and some of its properties were considered by the authors in [29]. Although several 2D grammars have been proposed in the literature, as far as we know no attempt has been made in the literature to examine the effect of pure grammar type of rewriting of arrays except in a specific model [17], called $T0LAS$, but this model allows rewriting only at the borders of an array.

Motivated by the fact that hexagonal arrays and hexagonal patterns occur in many places in the literature on picture processing and scene analysis, the problem of generation of hexagonal arrays on triangular grids has been considered and formal models have been proposed in [30] and these models have been further studied in [31,32]. Here we examine the problem of generation of hexagonal arrays by introducing a pure 2D hexagonal context-free grammar ($P2DHCFG$) analogous to the $P2DCFG$ generating rectangular arrays.

2 Basic Definitions and Results

Let Σ be a finite alphabet. A word or a string $w = a_1a_2 \dots a_n \quad (n \geq 1)$ over Σ is a sequence of symbols from Σ. The length of a word w is denoted by $|w|$. The set of all words over Σ, including the empty word λ with no symbols, is denoted by Σ^*. We call words of Σ^* horizontal words. For any word $w = a_1a_2 \dots a_n$,

we denote by w^T the vertical word

$$\begin{matrix} a_1 \\ \vdots \\ a_n \end{matrix}$$

We also define $(w^T)^T = w$. We set λ^T as λ itself.

A rectangular $m \times n$ array M over Σ (also called a picture array) is of the form

$$M = \begin{matrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{matrix}$$

where each $a_{ij} \in \Sigma, 1 \leq i \leq m, 1 \leq j \leq n$. The set of all rectangular arrays over Σ is denoted by Σ^{**}, which includes the empty array λ. Σ^{++} consists of all the rectangular arrays of Σ^{**} excluding the empty array λ. i.e. $\Sigma^{++} = \Sigma^{**} - \{\lambda\}$. We denote respectively by $\circ$ and $\diamond$, the *column concatenation* and *row concatenation* of arrays in Σ^{**}. In contrast to the case of strings, these operations are partially defined, namely, for any $X, Y \in \Sigma^{**}$, $X \circ Y$ is defined if and only if X and Y have the same number of rows. Similarly $X \diamond Y$ is defined if and only if X and Y have the same number of columns.

We refer to [1–3] for array grammars and two-dimensional languages. For notions of formal language theory we refer to [19]. We briefly recall pure context-free grammars [21] and the rectangular picture generating models in [1,5,8,16,17].

A pure context-free grammar [21] is $G = (\Sigma, P, \Omega)$ where Σ is a finite alphabet, Ω is a set of axiom words and P is a finite set of context-free rules of the form $a \rightarrow \alpha, a \in \Sigma, \alpha \in \Sigma^*$. Derivations are done as in a context-free grammar except that unlike a context-free grammar, there is only one kind of symbol, namely the terminal symbol. The language generated consists of all words generated from each axiom word.

Example 1 The pure context-free grammar $G = (\Sigma = \{a, b, c\}, P = \{c \rightarrow acb\}, \Omega = \{acb\})$ generates the language $\{a^n c b^n | n \geq 1\}$.

In the 2D grammar model introduced in [5], which we call as Siromoney matrix grammar, a horizontal word $S_{i1} \ldots S_{in}$ over intermediate symbols is generated by a Chomskian grammar. Then from each intermediate symbol S_{ij} a vertical word of the same length over terminal symbols is derived to constitute the jth column of the rectangular array generated. We recall this model restricting to regular and context-free cases.

Definition 1 A Siromoney matrix grammar [5] is a 2−tuple (G_1, G_2) where $G_1 = (H_1, I_1, P_1, S)$ is a regular or a context-free grammar,

- H_1 is a finite set of horizontal nonterminals,
- $I_1 = \{S_1, S_2, \cdots, S_k\}$, a finite set of intermediates, $H_1 \cap I_1 = \emptyset$,
- P_1 is a finite set of production rules called horizontal production rules,
- S is the start symbol, $S \in H_1$,

$G_2 = (G_{21}, G_{22}, \cdots, G_{2k})$ where $G_{2i} = (V_{2i}, T, P_{2i}, S_i), 1 \leq i \leq k$ are regular grammars,

- V_{2i} is a finite set of vertical nonterminals, $V_{2i} \cap V_{2j} = \emptyset,\ i \neq j$,
- T is a finite set of terminals,
- P_{2i} is a finite set of right linear production rules of the form

$X \longrightarrow aY$ or $X \longrightarrow a$ where $X, Y \in V_{2i}, a \in T$

- $S_i \in V_{2i}$ is the start symbol of G_{2i}.

The type of G_1 gives the type of G ; so we speak about regular or context-free Siromoney matrix grammars if G_1 is regular or context-free respectively.

Derivations are defined as follows: First a string $S_{i1}S_{i2} \cdots S_{in} \in I_1^*$ is generated horizontally using the horizontal production rules of P_1 in G_1. That is, $S \Rightarrow S_{i1}S_{i2} \cdots S_{in} \in I_1^*$.
Vertical derivations proceed as follows: We write

$$\begin{array}{c} A_{i1} \cdots A_{in} \\ \Downarrow \\ a_{i1} \cdots a_{in} \\ B_{i1} \cdots B_{in} \end{array}$$

if $A_{ij} \rightarrow a_{ij}B_{ij}$ are rules in $P_{2j}, 1 \leq j \leq n$. The derivation terminates if $A_j \rightarrow a_{mj}$ are all terminal rules in G_2.

The set $L(G)$ of picture arrays generated by G consists of all $m \times n$ arrays $[a_{ij}]$ such that $1 \leq i \leq m,\ 1 \leq j \leq n$ and $S \Rightarrow^*_{G_1} S_{i1}S_{i2} \cdots S_{in} \Rightarrow^*_{G_2} [a_{ij}]$. We denote the picture language classes of regular, context-free Siromoney matrix grammars by $RML, CFML$ respectively.

The regular/context-free Siromoney matrix grammars were extended in [16] by specifying a finite set of tables of rules in the second phase of generation with each table having either right-linear nonterminal rules or right-linear terminal rules. The resulting families of picture array languages are denoted by $TRML$ and $TCFML$ and are known to properly include RML and $CFML$ respectively.

We now recall the rectangular array generating model considered in [17].

Definition 2 A table $0L$ array system ($T0LAS$) [17] is $G = (T, \mathcal{P}, M_0)$ where
- T is a finite nonempty set (the alphabet of G);
- $\mathcal{P}$ is a finite set of tables, $\{t_1, t_2, \ldots, t_k\}$, and each t_i, $i = 1, \ldots, k$, is a left, right, up, or down table consisting respectively, of a finite set of left, right, up, or down rules only. The rules within a table are context-free in nature but all right hand sides of rules within the same table are of the same length;
- $M_0 \in \Sigma^{++}$ is an axiom array of G.

A derivation in G takes place as follows: Starting with a rectangular array $M_1 \in \Sigma^{++}$, all the symbols of either the leftmost/rightmost column or the uppermost/bottommost row of M_1 are rewritten in parallel respectively by the rules of a left or a right table or an up or a down table to yield a rectangular array M_2. A set $\mathcal{M}(G)$ of rectangular arrays is called a table $0L$ array language if and only if there exists a table $0L$ array system G such that $\mathcal{M}(G) = \{M | M_0 \Rightarrow^* M, M \in T^{**}\}$. The family of table $0L$ array languages is denoted by $T0LAL$.

Another interesting model called tiling system (TS) describing rectangular picture arrays was introduced in [1,8]. This model is based on a well known characterization of recognizable string languages in terms of local languages and projections. In fact the notion of a local string language is extended to two dimensions. The idea [1] here is that a "window" of size 2×2 is moved around a rectangular picture or array of terminal symbols and a record is made of 2×2 tiles (or 2×2 rectangular arrays) observed through the window. The order and the number of occurrences of these tiles is not taken into account. If the set of recorded 2×2 tiles is included in a given set of 2×2 tiles, then the rectangular array is 'accepted' as a member of a 'local picture language' to be formed. A picture language of rectangular arrays is said to be tiling recognizable [1] if it is the image under a projection, which is a letter-to-letter mapping, of a local picture language. We now briefly recall these notions.

Given a rectangular picture array p of size $m \times n$ over an alphabet Σ, $\hat{p}$ is an $(m+2) \times (n+2)$ picture array obtained by surrounding p by the special symbol $\# \notin \Sigma$ in its border. A square picture array of size 2×2 is called a tile. The set of all tiles which are sub-pictures of p is denoted by $B_{2\times 2}(p)$.

Definition 3 Let Γ be a finite alphabet. A two-dimensional language or a picture array language $L \subseteq \Gamma^{**}$ is called local if there exists a finite set Θ of tiles over the alphabet $\Gamma \cup \{\#\}$ such that $L = \{p \in \Gamma^{**} | B_{2\times 2}(\hat{p}) \subseteq \Theta\}$. The family of local picture array languages will be denoted by LOC [1,8].

Definition 4 A tiling system (TS) is a 4-tuple $T = (\Sigma, \Gamma, \Theta, \pi)$ where Σ and Γ are two finite alphabets, Θ is a finite set of tiles over the alphabet $\Gamma \cup \{\#\}$ and $\pi : \Gamma \to \Sigma$ is a projection.

The tiling system T recognizes a picture array language L over the alphabet Σ as follows: $L = \pi(L')$ where $L' = L(\Theta)$ is the local two-dimensional language over Γ corresponding to the set of tiles Θ. We write $L = L(T)$ and we say that L is the language recognized by T. A picture array language $L \subseteq \Sigma^{**}$ is tiling recognizable if there exists a tiling system T such that $L = L(T)$. The family of tiling recognizable picture array languages is denoted by REC [1,8].

3 Pure 2D Context-free Grammars

Based on the notion of pure context-free rules, a new two-dimensional grammar is introduced for picture generation. The salient feature of this model is that the shearing effect in replacing a subarray of a given rectangular array is taken care of by rewriting a row or a column of symbols in parallel by equal length strings and by using only terminal symbols as in a pure string grammar [21]. This new model is related to the model $T0LAS$ in [17] in the sense that a column or row of symbols of a rectangular array is rewritten in parallel. This feature as in [17] incorporates into arrays the parallel rewriting feature of the well-known and widely investigated Lindenmayer systems [18]. But the difference between this new model and the $T0LAS$ in [17] is that the rewriting is done only at the "edges" of a rectangular array in a $T0LAS$ whereas here we allow rewriting in parallel of any column or any row of symbols. We now define the new grammar model, a preliminary version of which was introduced by the authors in [29].

Definition 5 A pure 2D context-free grammar ($P2DCFG$) is a 4-tuple $G = (\Sigma, P_c, P_r, \mathcal{M}_0)$ where

- Σ is a finite set of symbols ;
- $P_c = \{t_{c_i} | 1 \leq i \leq m\}, P_r = \{t_{r_j} | 1 \leq j \leq n\}$;

Each $t_{c_i}, (1 \leq i \leq m)$, called a column table, is a set of context-free rules of the form $a \rightarrow \alpha, a \in \Sigma, \alpha \in \Sigma^*$ such that for any two rules $a \rightarrow \alpha, b \rightarrow \beta$ in t_{c_i}, we have $|\alpha| = |\beta|$ where $|\alpha|$ denotes the length of α;
Each $t_{r_j}, (1 \leq j \leq n)$, called a row table, is a set of context-free rules of the form $c \rightarrow \gamma^T, c \in \Sigma$ and $\gamma \in \Sigma^*$ such that for any two rules $c \rightarrow \gamma^T, d \rightarrow \delta^T$ in t_{r_j}, we have $|\gamma| = |\delta|$;

- $\mathcal{M}_0 \subseteq \Sigma^{**} - \{\lambda\}$ is a finite set of axiom arrays.

Derivations are defined as follows: For any two arrays M_1, M_2, we write $M_1 \Rightarrow M_2$ if M_2 is obtained from M_1 by either rewriting a column of M_1 by rules of some column table t_{c_i} in P_c or a row of M_1 by rules of some row table t_{r_j} in P_r. $\Rightarrow^*$ is the reflexive transitive closure of $\Rightarrow$.
The picture array language $L(G)$ generated by G is the set of rectangular picture arrays $\{M | M_0 \Rightarrow^* M \in \Sigma^{**}$, for some $M_0 \in \mathcal{M}_0\}$. The family of picture

$$
\begin{array}{ccccccccccc}
 & & & & & & x & . & . & x & \quad x & . & . & . & x \\
 & & & & x & . & . & x & \quad x & . & . & x & \quad x & . & . & . & x \\
 & & x & . & . & x & \quad x & . & . & x & \quad x & . & . & . & x \\
M_{01} \Rightarrow & & \mathbf{z} & \mathbf{y} & \mathbf{y} & \mathbf{z} & \Rightarrow & \mathbf{z} & \mathbf{y} & \mathbf{y} & \mathbf{z} & \Rightarrow & z & \mathbf{y} & y & z & \Rightarrow & z & y & y & y & z & = M_1 \\
 & & x & . & . & x & \quad x & . & . & x & \quad x & . & . & . & x \\
 & & & & & & x & . & . & x & \quad x & . & . & . & x \\
 & & & & & & & & & & x & . & . & x & \quad x & . & . & . & x
\end{array}
$$

Fig. 1. Derivation $M_{01} \Rightarrow^* M_1$

$$
\begin{array}{ccccccc}
x & x & x & y & x & x & x \\
. & . & . & z & . & . & . \\
. & . & . & z & . & . & . \\
. & . & . & z & . & . & . \\
. & . & . & z & . & . & .
\end{array}
$$

Fig. 2. A picture array M_2

array languages generated by pure 2D context-free grammars is denoted by $P2DCFL$.

Example 2 Consider the pure 2D context-free grammar $G_1 = (\Sigma_1, P_{c_1}, P_{r_1}, \{M_{01}\})$ where $\Sigma_1 = \{x, y, z, .\}$, $P_{c_1} = \{t_{c_1}\}, P_{r_1} = \{t_{r_1}\}$

$$
t_{c_1} = \{. \rightarrow .., y \rightarrow yy\},\ t_{r_1} = \left\{ \begin{array}{ccc} & . & \\ y \rightarrow y, & & z \rightarrow z \\ & . & \end{array} \right\},\ M_{01} = \begin{array}{ccc} x & . & x \\ z & \mathbf{y} & z \\ x & . & x \end{array}
$$

A sample derivation $M_{01} \Rightarrow^* M_1$, on using $t_{c_1}, t_{r_1}, t_{r_1}, t_{c_1}$ in this order, is given in Figure 1. A column or row of symbols in boldface in Figure 1, indicates the column or row rewritten in the subsequent step. Each of the arrays occurring in the derivation given belongs to the picture array language generated by G_1.

Example 3 Consider the pure 2D Context-free grammar $G_2 = (\Sigma_2, P_{c_2}, P_{r_2}, \{M_{02}\})$ where $\Sigma_2 = \{x, y, z, .\}$, $P_{c_2} = \{t_{c_2}\}, P_{r_2} = \{t_{r_2}\}$

$$
t_{c_2} = \{y \rightarrow xyx, z \rightarrow .z.\}\ t_{r_2} = \left\{ x \rightarrow \begin{array}{c} x \\ . \end{array}, y \rightarrow \begin{array}{c} y \\ z \end{array} \right\}\ M_{02} = \begin{array}{ccc} x & y & x \\ & . \ z \ . & \end{array}
$$

G_2 generates picture arrays M_2 of the form shown in Figure 2.

Here again we note that the number of rows in the generated picture array need not have any proportion to the number of columns but will have an equal number of columns to the left and right of the middle column $(yz \dots z)^T$.

We now examine certain closure properties of P2DCFL. We also consider geometric operations of transposition, reflection about base, reflection about leg. The operation of transposition of a rectangular array interchanges the rows and columns. The operation of reflection about the base reflects the rectangular array about the bottommost row and of reflection about the leg reflects the rectangular array about the leftmost column.

Theorem 1 The class of $P2DCFL$ is not closed under the operations of union, column catenation, row catenation but the class is closed under the geometric operations of transposition, reflection about the base and reflection about the leg.

Proof Let the alphabet be $\{a, b, c, x, y\}$. Non-closure under union can be seen as follows: Consider the picture languages L_1 consisting of rectangular arrays with a middle column of $c's$ and equal size arrays over $a's$ to the left and $b's$ to the right of this column of $c's$ and L_2 consisting of similar arrays but with $x's$ and $y's$ in the place of $a's$ and $b's$. In other words $L_1 = \{X_1 \circ (c^n)^T \circ Y_1 | X_1 \in \{a\}^{++}, Y_1 \in \{b\}^{++}, |X_1|_c = |Y_1|_c, n \geq 1\}$ and $L_2 = \{X_2 \circ (c^n)^T \circ Y_2 | X_2 \in \{x\}^{++}, Y_2 \in \{y\}^{++}, |X_2|_c = |Y_2|_c, n \geq 1\}$ where $|X|_c$ stands for the number of columns of X. Note that X_1 is an array over a and Y_1 is an array over b. By the definition of the operation $\circ$, the array $X_1 \circ (c^n)^T \circ Y_1$ can be formed only when X_1 and Y_1 have as many rows as the number of $c's$ in the middle column of $c's$. Likewise for $X_2 \circ (c^n)^T \circ Y_2$. L_1 is generated by a $P2DCFG$ with a column table consisting of a rule $c \rightarrow acb$ and a row table with rules $a \rightarrow \begin{matrix} a \\ a \end{matrix}, b \rightarrow \begin{matrix} b \\ b \end{matrix}, c \rightarrow \begin{matrix} c \\ c \end{matrix}$. Likewise L_2 is also generated by a similar $P2DCFG$. It can be seen that $L_1 \cup L_2$ cannot be generated by any $P2DCFG$, since such a grammar will require a column table with rules of the forms $c \rightarrow acb$ and $c \rightarrow xcy$. But then this will yield arrays not in the union. Non-closure under column catenation of arrays can be seen by considering $L_1 \circ L_2$ and noting that any $P2DCFG$ generating $L_1 \circ L_2$ will again require a column table with rules $c \rightarrow acb$ and $c \rightarrow xcy$ but then this will lead to generating arrays not in the column catenation $L_1 \circ L_2$. Non-closure under row catenation can be seen in a similar manner.
If L is a picture array language generated by a $P2DCFG$ G and L^T is the transposition of L, then the $P2DCFG$ G' to generate L^T is formed by taking the column tables of G as row tables and row tables as column tables but for a rule $a \rightarrow \alpha$ in a column table of G, the rule $a \rightarrow \alpha^T$ ($\alpha \in \Sigma^{**}$) is added in the corresponding row table of G' and likewise for a rule $b \rightarrow \beta^T$ ($\beta \in \Sigma^{**}$) in

a row table of G, the rule $b \rightarrow \beta$ is added in the corresponding column table of G'. Closure under the operations of reflection about base, reflection about leg can be seen in a similar manner. $\Box$

We now compare the new 2D grammar model of pure 2D context-free grammar introduced here with those in [1,5,8,16,17].

Theorem 2 The family of $P2DCFL$ is incomparable with the families of RML and $CFML$ but not disjoint with these families.

Proof The picture language consisting of rectangular arrays of all sizes $m \times n$ $(m, n \geq 1)$ over a single symbol a is in $P2DCFL \cap RML$. In fact it is generated by a regular Siromoney matrix grammar G with rules in the horizontal phase given by $S \rightarrow S_1S, S \rightarrow S_1$ where S_1 is an intermediate symbol and with rules in the vertical phase given by $S \rightarrow (aS_1)^T, S_1 \rightarrow a$. A corresponding pure 2D CF grammar consists of a column table with the rule $a \rightarrow aa$ and a row table with the rule $a \rightarrow \begin{matrix} a \\ a \end{matrix}$ and the axiom array a.
The incomparability with $CFML$ can be seen as follows: The $P2DCFL$ in example 2 cannot be generated by any context-free Siromoney matrix grammar and hence by any regular Siromoney matrix grammar since each of the generated pictures of example 2, has an equal number of rows above and below the middle row $zy \ldots yz$. On the other hand a picture language consisting of rectangular arrays of the form $M_1 \circ M_2$ where M_1 is a rectangular array over the symbol a and M_2 over the symbol b with M_1 and M_2 having an equal number of columns can be generated by a context-free Siromoney matrix grammar. In fact the language of horizontal words is $S_1^n S_2^n$ (S_1, S_2 are intermediate symbols) in the first phase and $S_1 \rightarrow (aS_1)^T, S_1 \rightarrow a, S_2 \rightarrow (bS_2)^T, S_2 \rightarrow b$ are the rules in the vertical phase. This picture language, cannot be generated by any pure 2D context-free grammar since an argument similar to the fact that the string language $\{a^n b^n | n \geq 1\}$ is not a pure CFL [21] can be done in the two-dimensional case also.
The incomparability of $P2DCFL$ with RML can also be seen by noting that the picture language with rectangular arrays each row of which is a word in $a^3b^3(ab)^*$ cannot be generated by any $P2DCFG$. This can be seen by an argument analogous to the fact [21] that the string language $a^3b^3(ab)^*$ is not a pure CFL. On the other hand is generated by the RMG with the language of the horizontal phase as $S_1^3 S_2^3 (S_1S_2)^*$ and with rules in the vertical phase given by $S_1 \rightarrow (aS_1)^T, S_1 \rightarrow a, S_2 \rightarrow (bS_2)^T, S_2 \rightarrow b$. $\Box$

Theorem 3 The family of $P2DCFL$ is incomparable with the families of $TRML$ and $TCFML$ but not disjoint with these families.

Proof The proper inclusions $RML \subset TRML, CFML \subset TCFML$ are known

[16]. So due to the incomparability (Theorem 2) of $P2DCFL$ with RML and $CFML$, it is enough to note that the picture array language of example 2 generating picture arrays as shown in Figure 1 can neither belong to $TRML$ nor to $TCFML$, in view of the fact that in the generated picture arrays (Figure 1) of example 2, each has an equal number of rows above and below the middle row $zy \dots yz$ and no $TRMG$ or $TCFMG$ can generate arrays with this feature. □

Theorem 4 The family $P2DCFL$ contains languages that cannot be described by any $T0LAS$.

Proof Since in a $T0LAS$, a rectangular array can "grow" only at its borders by definition, it is clear that the picture array language generated by a $P2DCFG$ in example 3 consisting of picture arrays in the shape of token T (Figure 2) having an equal number of $x's$ to the left and right of the middle column, cannot be generated by any $T0LAS$. □

Theorem 5 Every language in the family $T0LAL$ is a coding of a $P2DCFL$.

Proof Let L be a picture array language generated by a $T0LAS$ [17] $G = (T, \mathcal{P}, M_0)$. We construct a pure $2DCFG$ G' as follows: For each symbol a in the alphabet T of G, we introduce a new distinct symbol A. Let $T' = \{A | a \in T\}$. Each rule of the form $a \rightarrow a_1 a_2 \cdots a_m b, (b, a_i (1 \leq i \leq m) \in T)$ in a right table t of G, is replaced by a rule $A \rightarrow a_1 a_2 \cdots a_m B, A, B \in T'$. Here A corresponds to a and B to b. Each rule of the form $a \rightarrow a_1 a_2 \cdots a_m b, (b, a_i (1 \leq i \leq m) \in T)$ in a down table t of G, is replaced by a rule $A \rightarrow (a_1 a_2 \cdots a_m B)^T, A, B \in T'$. Likewise the rules in left and up tables are replaced by rules constructed with a similar idea. Then $G' = (T \cup T', \mathcal{P}', \{M_0^{'}\})$ where $\mathcal{P}'$ consists of the tables of G with each table having the rules modified as mentioned above. The modified left and right tables of G become the column tables of G' and the modified up and down tables of G become the row tables of G'. The axiom array $M_0^{'}$ is M_0 with its border symbols replaced by the new symbols. Define a coding c (a letter to letter mapping) by $c(A) = a$ where A is the new symbol introduced corresponding to a. It can be seen that $c(L(G')) = L$. □

Theorem 6 The family of $P2DCFL$ is incomparable with the families LOC [1,8] and REC [1,8].

Proof The language of square picture arrays with $1's$ in the main diagonal and $0's$ in other positions is known [1,8] to be in LOC and the language of square picture arrays over $0's$ is known [1,8] to be in REC but both these languages cannot be generated by any $P2DCFG$ for it can be seen that the language of square arrays cannot be generated by a $P2DCFG$ with at most two symbols $0, 1$. On the other hand a picture array language L_1 consisting of arrays $M = M_1 \circ c \circ M_1$ where M_1 is a string over the symbol a (M is a

picture array with only one row) is generated by a $P2DCFG$ with a column rule $c \rightarrow aca$ but L_1 is known [1,8] to be not in REC and hence not in LOC.□

4 P2DCFG with regular control

In formal language theory [19], one of the tools in regulating rewriting of words is to control the sequence of application of rules of a grammar by requiring the control words to belong to a language. Generally, if the control words constitute a regular language, the generative power of a grammar might not increase. Here we associate a regular control language with a pure 2D CFG and notice that the generative power increases.

Definition 6 A pure 2D context-free grammar with a regular control is $G_c = (G, Lab(G), \mathcal{C})$ where G is a pure 2D context-free grammar, $Lab(G)$ is a set of labels of the tables of G and $\mathcal{C} \subseteq Lab(G)^*$ is a regular (string) language. The words in $Lab(G)^*$ are called control words of G. Derivations $M_1 \Rightarrow_w M_2$ in G_c are done as in G except that if $w \in Lab(G)^*$ and $w = l_1 l_2 \dots l_m$, then the tables of rules with labels $l_1, l_2, \dots, l_m$ are successively applied starting with M_1 to yield M_2. The picture array language generated by G_c consists of all picture arrays obtained from the axiom array of G with the derivations controlled as described above. We denote by $(R)P2DCFL$ the family of picture array languages generated by pure 2D context-free grammars with a regular control.

Theorem 7 The family of $P2DCFL$ is properly contained in $(R)P2DCFL$.

Proof The containment follows from the fact that a $P2DCFL$ is generated by a $P2DCFG$ G with the regular control language $Lab(\mathcal{C})^*$.
The proper containment can be seen as follows: Consider the picture array language consisting of the picture arrays as shown in Figure 1 but with sizes $(2n+1) \times (n+2), n \geq 1$. In each array there is a proportion between the height (the number of rows in a picture array) and width (the number of columns in a picture array). The number of rows above and below the middle row $zy \dots yz$ equals the number of columns between the leftmost and rightmost columns, namely, $(x \dots xzx \dots x)^T$. This picture array language is generated by the pure 2D context-free grammar G in example 2 with a regular control language $\{(l_1 l_2)^n | n \geq 1\}$ on the labels l_1, l_2 of the tables t_{c_1}, t_{r_1} respectively. In fact the tables of rules generating the picture array language in example 2 are $t_{c_1} = \{. \rightarrow .., y \rightarrow yy\}$, $t_{r_1} = \left\{ \begin{array}{cc} . & x \\ y \rightarrow y\,, & z \rightarrow z \\ . & x \end{array} \right\}$. Since the control language on the labels of the tables consists of words $(l_1 l_2)^n$, an application of the rules

of the table t_{c_1} is immediately followed by an application of the rules of the table t_{r_1} so that the array rewritten grows one column followed by one row above and one row below the middle row $zy \ldots yz$. The resulting array is then collected in the language generated. This process is repeated so that the arrays generated have a proportion between the width and height as mentioned.□

Generating "square arrays" over one symbol a is of interest in picture array generation. Such square arrays can be generated by a 'simple' $P2DCFG$ with a regular control. In fact the $P2DCFG$ $(\{a\}, \{t_c\}, \{t_r\}, a)$ where $t_c = \{a \rightarrow aa\}, t_r = \{a \rightarrow (aa)^T\}$ with the regular control language $\{(l_1 l_2)^n | n \geq 1\}$ where l_1, l_2 are respectively the labels of t_c, t_r can be seen to generate the picture array language consisting of square arrays over one symbol a.

We now indicate applications of the $P2DCFG$ and $P2DCFG$ with regular control in generating interesting classes of chain code [9] pictures or "kolam" [7] pictures. This is done by replacing the letter symbols in the picture arrays generated by these grammars by 'primitive patterns'. This is a well-known technique to generate such picture patterns. Each symbol of a rectangular array is considered to occupy a unit square in the rectangular grid so that a row of symbols or a column of symbols in the array respectively occupies a horizontal or a vertical sequence of adjacent unit squares. A mapping i, called an interpretation, from the alphabet $\Sigma = \{a_1, a_2, \ldots, a_n\}$ of a $P2DCFG$ G to a set of primitive picture patterns $\{p_1, p_2, \ldots p_m\}$ is defined such that for $1 \leq i \leq n$, $i(a_i) = p_j$, for some $1 \leq j \leq m$. A primitive picture pattern could be a blank, denoted by b. Given a picture array M over Σ, $i(M)$ is obtained by replacing every symbol $a \in M$ by the corresponding picture pattern $i(a)$.

For instance, in Example 2, if we define, the interpretation mapping i by $i(x) = i(z) =$ | , $i(y) = -$ and $i(.) = b$, using two chain code primitives, namely, | , − , then the interpretation $i(M_1)$ of the array M_1 in Figure 1 will give a picture of the alphabetic letter H.

Likewise if the primitive picture patterns are those used in "kolam" pictures, we can obtain "kolam" patterns [7] from pure $2DCFL$ via suitable interpretation. "Kolam" [33] refers to decorative artwork drawn on the floor with the kolam drawing generally starting with a certain number of pattern points and curly lines going around these points. We illustrate by giving a pure 2D context-free grammar with a regular control to generate the rectangular arrays, a member of which is shown in Figure 3 and an interpretation i that yields the "kolam" patterns, a member of which is shown in Figure 5. The primitive patterns [33] used in a "kolam" pattern are shown in Figure 4.

Consider the pure 2D context-free grammar with regular control
$G_{c1} = (G, Lab(G), \mathcal{C})$ where $G = (\Sigma, P_c, P_r, M_0)$, with
$\Sigma = \{u, u_t, u_b, u_l, v, v_t, v_b, v_r, x, y, z, w, s, s_1, s_2\}$,

$$M = \begin{array}{cccccccccccccc} z & z & z & z & z & u_t & x & x & x & v_t & z & z & z & z & z \\ u_l & s & s & s & s & s_1 & x & x & x & s_2 & s & s & s & s & v_r \\ y & y & y & y & y & u & x & x & x & v & y & y & y & y & y \\ u & s & s & s & s & s & x & x & x & s & s & s & s & s & v \\ y & y & y & y & y & u & x & x & x & v & y & y & y & y & y \\ u & s & s & s & s & s & x & x & x & s & s & s & s & s & v \\ w & w & w & w & w & u_b & x & x & x & v_b & w & w & w & w & w \end{array}$$

Fig. 3. An Array M generated by G_{c1}

$P_c = \{t_{c1}, t_{c2}\},\ P_r = \{t_{r1}\},\ \ Lab(G) = \{t_{c1}, t_{c2}, t_{r1}\},\ \mathcal{C} = \{(t_{c1}t_{c2}t_{r1})^n | n \geq 1\}.$

The tables of rules are given by

$t_{c1} = \{u_t \to zu_tx, u \to yux, u_b \to wu_bx, s_1 \to ss_1x, s \to ssx\},$

$t_{c2} = \{v_t \to v_tz, v \to vy, v_b \to v_bw, s_2 \to s_2s, s \to ss\},$

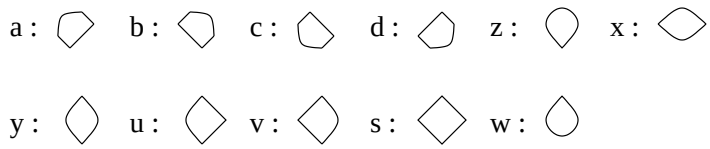
$$t_{r1} = \left\{ \begin{array}{ccccccccccc} u_l & & s & & s_1 & & x & & s_2 & & v_r \\ u_l \to y\ , & s \to y\ , & s_1 \to u\ , & x \to x\ , & s_2 \to v\ , & v_r \to y \\ u & & s & & s & & x & & s & & v \end{array} \right\}$$

$$M_0 = \begin{array}{cccccc} z & z & u_t & v_t & z & z \\ u_l & s & s_1 & s_2 & s & v_r \\ w & w & u_b & v_b & w & w \end{array}$$

The interpretation i is given by

$i(u_l) = i(u_t) = i(u_b) = i(u) = u$, $i(z) = z$, $i(x) = x$, $i(s_1) = i(s_2) = i(s) = s$, $i(v_r) = i(v_t) = i(v_b) = i(v) = v$, $i(w) = w$, $i(y) = y$

The axiom array M_0 yields the array M (Figure 3) with the derivation controlled by the control word $w = t_{c1}t_{c2}(t_{c1}t_{c2}t_{r1})^2$. i.e. $M_0 \Rightarrow_w M$. The interpretation i applied to the array M gives the "kolam" pattern in Figure 5.



a, b, c, d : saddle x, y :pupil u, v : fan s : diamond z, w : drop

Fig. 4. Primitive patterns of a "kolam" pattern

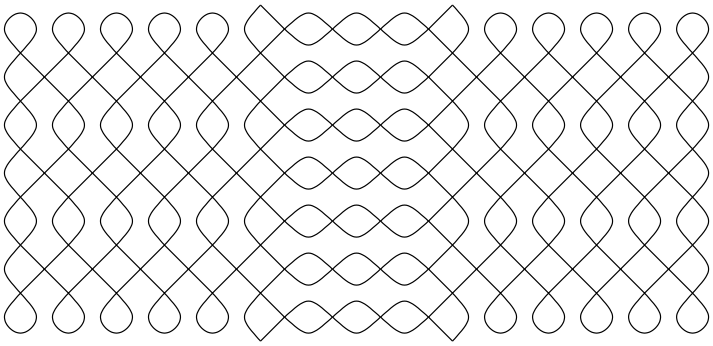


Fig. 5. A "kolam" pattern

5 Pure 2D Hexagonal Context-free Grammars

Hexagonal array generating models were introduced in [30] by considering hexagonal arrays on a triangular grid. The notion of 'catenation'of strings and column catenation and row catenation of rectangular arrays were extended in [30] to 'arrowhead catenation' of hexagonal arrays where the 'arrowhead' is specific kind of a hexagonal array. In fact it is observed in [30] that on treating a hexagonal array as a two-dimensional representation of a three dimensional block, the two-dimensional representation captured the effect of placing a block having one face equal to a hidden face of the original block (see [30] for more details on this notion and other motivations). In [31,32] the study on generation of hexagonal arrays is continued. Here we define pure 2D hexagonal context-free grammars ($P2DHCFG$) analogous to the rectangular case considered earlier and examine its properties. We first recall from [30] relevant notions pertaining to hexagonal arrays.

We consider a triangular grid made up of lines equally inclined and parallel to three fixed directions (upper right $\nearrow$, upper left $\nwarrow$, down $\downarrow$) and their duals (lower left $\swarrow$, lower right $\searrow$ up $\uparrow$). For formal definitions relating to hexagonal arrays, we refer to [30]. An example hexagonal array H is shown in Figure 6(Left). An "arrowhead" hexagonal array H_a (which is a convex hexagonal array) is shown in Figure 6(right). (In the "arrowhead" H_a the six "sides", read anti-clockwise, are $bxxx$, $xxxb$, bb, $bbbb$, bb whereas in the hexagonal array H the six "sides" are $bbbb$, $bbbb$, $bbbb$, $bbbb$, $bbbb$, $bbbb$.)

We will denote the set of all hexagonal arrays over Σ by Σ^{h**}.

Now we can define a pure 2D hexagonal context-free grammar analogous to

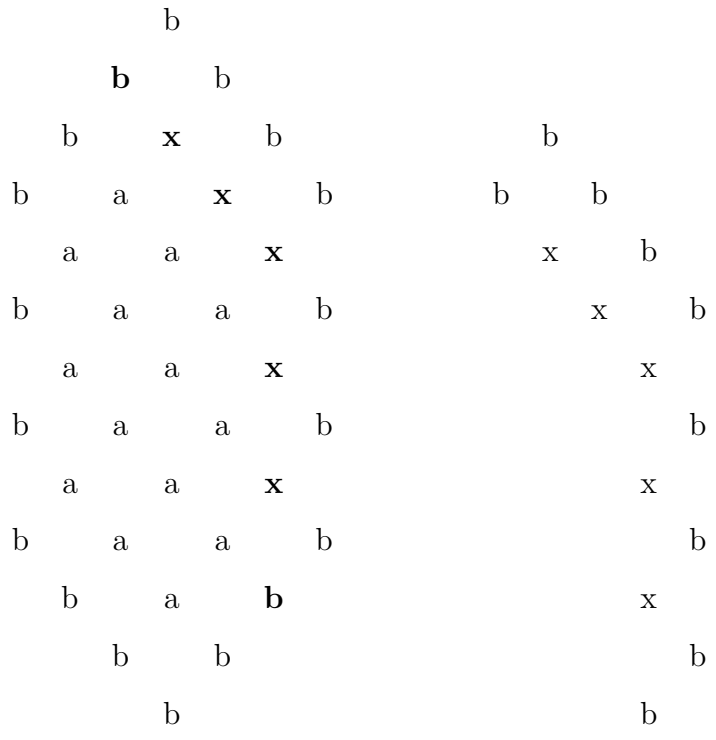


Fig. 6. Left: A hexagonal array H_0 Right: An "arrowhead" hexagonal array H_a

$P2DCFG$ by requiring only one kind of symbol and rewriting of a hexagonal array taking place with rules that rewrite all the symbols in an "arrowhead of thickness one" amounting to context-free rewriting. An "arrowhead" of thickness one is a degenerate arrowhead hexagonal array. The arrowhead rewriting can be in any of the six directions, namely $\nearrow$, $\nwarrow$, $\downarrow$ and their duals. This corresponds to column or row rewriting in the rectangular case of $P2DCFG$.

Definition 7 A pure $2D$ hexagonal context-free grammar ($P2DHCFG$) is $G = (\Sigma, P_{ur}, P_{ul}, P_d, P_{ll}, P_{lr}, P_u, \mathcal{H}_0)$ where

- Σ is a finite set of symbols ;
- $P_{ur} = \{t_{ur_i} | 1 \leq i \leq m\}$;

Each $t_{ur_i}, (1 \leq i \leq m)$, called a UR table, is a set of context-free rules of the form $a \rightarrow \alpha, a \in \Sigma, \alpha \in \Sigma^*$ such that for any two rules $a \rightarrow \alpha, b \rightarrow \beta$ in t_{ur_i}, we have $|\alpha| = |\beta|$ where $|\alpha|$ denotes the length of α;
Each of the other five components $P_{ul}, P_d, P_{ll}, P_{lr}, P_u$ is similarly defined.

- $\mathcal{H}_0 \subseteq \Sigma^{h**} - \{\lambda\}$ is a finite set of axiom arrays that are hexagonal arrays.

Derivations are defined as follows: For any two hexagonal arrays H_1, H_2, we write $H_1 \Rightarrow H_2$ if H_2 is obtained from H_1 by either rewriting all the symbols in an "arrowhead of thickness one" of H_1 by rules of a relevant table. $\Rightarrow^*$ is

the reflexive transitive closure of $\Rightarrow$.
The hexagonal picture array language $L(G)$ generated by G is the set of hexagonal picture arrays $\{H|H_0 \Rightarrow^* H \in \Sigma^{h**}$, for some $H_0 \in \mathcal{H}_0\}$. The family of hexagonal picture array languages generated by pure 2D hexagonal context-free grammars is denoted by $P2DHCFL$.

Example 4 Consider the following $P2DHCFG$ with alphabet $\Sigma = \{a, x, b\}$. We mention only the axiom hexagonal array and the tables of rules. The axiom hexagonal array H_0 is shown in Figure 6 (Left). We have only one table of rules that can be used to rewrite an "arrowhead of thickness one", namely $bxx < x > xxb$ where we have employed a compact notation [30] to indicate the fact that the symbol (here x) enclosed in <> is the "corner" of the "arrowhead". The table of rules is given by $t_{ur} = \{x \nearrow axb, b \nearrow bbb\}$. A hexagonal array H generated by this $P2DHCFG$ is shown in Figure 7. In fact in the hexagonal array H_0 (Fig. 6 (Left)), the symbols in the 'arrow head' that are rewritten by the rules of the table t_{ur}, are shown in bold and after rewriting the array H_0 yields H (Figure 7).

Among the different hexagonal array generating systems, table $0L$ hexagonal

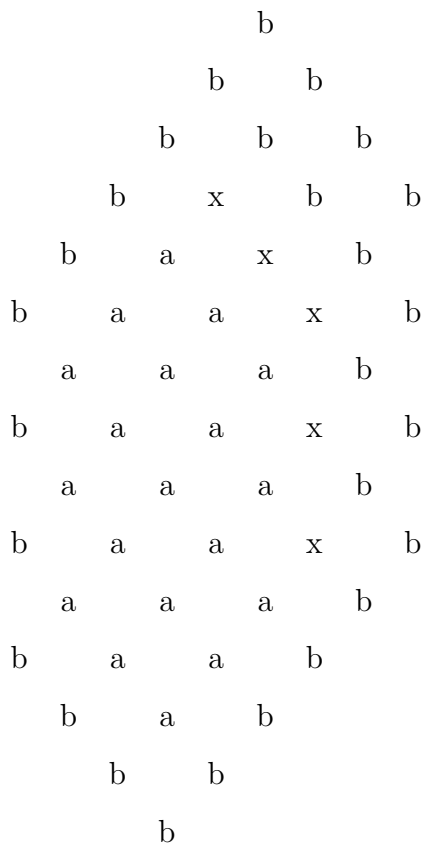


Fig. 7. A generated hexagonal array

array system ($T0LHAS$) [30] is an analogue of the corresponding rectangular

system $T0LAS$ [17]. In a $T0LHAS$ "growth" (rewriting) can take place only at the "borders" of the hexagonal array. The family of hexagonal arrays generated is denoted by $T0LHAL$ [30].

Here comparison of the generative power of the family $P2DHCFL$ with the family $T0LHAL$ can be made. We state the result in the following theorem.

Theorem 8 The family $P2DHCFL$ contains languages that cannot be described by any $T0LHAS$.
This statement again is a consequence of the fact that in a $T0LHAS$, "growth" in a hexagonal array can take place only in any of the six directions but only at the borders. But in a $P2DHCFG$ such a growth can take place even in the interior as in example 4 (Figure 7). □

6 Conclusion

A syntactic two-dimensional grammar model, called pure 2D context-free grammar, initially proposed in [29], based on the notion of pure string grammar [21] and generating rectangular picture arrays, has been considered here and its properties studied. The model is also modified suitably to give rise to an analogous 2D grammar model called pure 2D hexagonal context-free grammar generating hexagonal arrays. In the rectangular case $P2DCFG$ has been extended in [34] defining extended pure 2D hexagonal context-free grammar ($EP2DHCFG$) by allowing use of variables in the grammar and its rules and employing the well-known squeezing mechanism of obtaining arrays generated over a terminal alphabet. This notion has been extended to the hexagonal case in [35]. Also in the hexagonal case, the effect of regular control has been examined in [35] both for $P2DHCFG$ and $EP2DHCFG$. There also remain problems in both the rectangular and hexagonal cases of comparison with other 2D grammar models that have not been considered here. For example in the rectangular case comparison with the model in [36] and in the hexagonal case comparison with the local and recognizable hexagonal models in [37,38] can be made. The application of these grammars via the interpretation considered for handling more complex primitive patterns could be further explored.

Acknowledgement The authors are grateful to the referees for their very useful comments which helped to improve the presentation of the paper. The authors K.G. Subramanian and Rosihan M. Ali gratefully acknowledge support for this research by a FRGS grant and a RU grant of the Universiti Sains Malaysia. A preliminary version of the paper was presented at the 12th International Workshop on Combinatorial Image Analysis, Buffalo, NY, USA, April 2008 [29].

References

[1] D. Giammarresi, A. Restivo, Two-dimensional languages, in: G. Rozenberg and A. Salomaa (Eds.), Handbook of Formal Languages , Springer Verlag, Vol. 3, 1997, pp. 215-267.

[2] A. Rosenfeld, Picture Languages - Formal Models for Picture Recognition, Academic Press, New York, 1979.

[3] A. Rosenfeld, R. Siromoney, Picture languages - a survey, Languages of design, 1 (1993) 229 - 245.

[4] P.S.P. Wang (ed.), Array grammars, Patterns and recognizers, Series in Computer science, Vol. 18, World Scientific, 1989.

[5] G. Siromoney, R. Siromoney, K. Krithivasan, Abstract families of matrices and picture languages, Comp. Graphics Image Process., 1 (1972) 234-307.

[6] G. Siromoney, R. Siromoney, K. Krithivasan, Picture languages with array rewriting rules, Inform. Control, 22 (1973) 447-470.

[7] G. Siromoney, R. Siromoney, K. Krithivasan, Array grammars and kolam, Comp. Graphics Image Process., 3 (1974) 63-82.

[8] D. Giammarresi, A. Restivo, Recognizable picture languages, in: M. Nivat, A. Saoudi and P.S.P. Wang (Eds.), Special issue on Parallel Image Processing, Int. J. of Pattern Recogn. Artificial Intelligence, 1992, pp. 31-46.

[9] H.A. Maurer, G. Rozenberg, E. Welzl, Chain-code picture languages, Lecture Notes in Computer science, Springer-Berlin, 153 (1983) 232-244.

[10] K. Krithivasan, Variations of the matrix models, Int. J. Comput. Math., 6 (1977) 171-190.

[11] K. Krithivasan, R. Siromoney, Characterizations of regular and context-free matrices, Int. J. Comput. Math., 4 (1974) 229-245.

[12] R. Stiebe, Picture generation using matrix systems, J. Inform. Process. Cybernetics, 28 (1992) 311-327.

[13] R. Stiebe, Slender Siromoney matrix languages, Proceedings of the 1^{st} International Conference on Language and Automata: Theory and Applications, Tarragona, Spain (2007).

[14] K.G. Subramanian, L. Revathi, R. Siromoney, Siromoney array grammars and applications, in: *Array grammars, Patterns and recognizers* P.S.P. Wang (Ed.), Series in Computer science, World Scientific, Vol.18, 1990, 55-74.

[15] K.G. Subramanian, R. Siromoney, G. Siromoney, A note on an extension of matrix grammars generating two-dimensional languages, Inform. Sci., 35 (1985) 223-233.

[16] R. Siromoney, K.G. Subramanian, K. Rangarajan, Parallel/Sequential rectangular arrays with tables, Int. J. Comput. Math., 6A (1977) 143-158.

[17] R. Siromoney, G. Siromoney, Extended Controlled Tabled L- arrays, Inform. Control, 35 (1977) 119-138.

[18] G. Rozenberg, A. Salomaa, The Mathematical Theory of L systems, Academic Press, New York, 1980.

[19] A. Salomaa, Formal languages, Academic Press, London, 1973.

[20] A. Gabrielian, Pure grammars and pure languages, Int. J. Comput. Math., 9 (1981) 3-16.

[21] H.A. Maurer, A. Salomaa, D. Wood, Pure Grammars, Inform. Control, 44 (1980) 47-72.

[22] H. Bordhin, H. Fernau, M. Holzer, Accepting pure grammars and systems, Publicationes Mathematicae Debrecen, 60 (Supplement) (2002) 483-510.

[23] G. Georgescu, On the index of pure context-free grammars and languages, in: G. Rozenberg, A. Salomaa (Eds.), *Developments in Language Theory, At the Crossroads of Mathematics, Computer Science and Biology*, Turku, Finland, 1993, World Scientific, Singapore, 1994, pp 60-69.

[24] E. Mäkinen, Normal forms for pure context-free grammars, Bulletin of the EATCS, 31 (1987) 35-37 .

[25] E. Mäkinen, On Szilard languages of pure context-free grammars, J. Inform. Process. Cybernetics, 22 (1986) 527-532.

[26] E. Mäkinen, F.L. Tiplea, Pattern ambiguities for pure context-free grammars, Fundam. Inform., 30 (1997) 183-191.

[27] P. Martinek, Limits of pure grammars with monotone productions, Fundam. Inform., 33 (1998) 265-280.

[28] M. Novotný, Construction of pure grammars, Fundam. Inform., 52 (2002) 345-360.

[29] K. G. Subramanian, A.K. Nagar, M. Geethalakshmi, Pure 2D picture grammars (P2DPG) and P2DPG with regular control, in: V.E. Brimkov, R.P. Barneva, H.A. Hauptman (Eds.), Combinatorial Image Analysis, Lecture Notes in Computer Science, Vol. 4958, Springer, Berlin, 2008, pp. 330-341.

[30] G. Siromoney, R. Siromoney, Hexagonal arrays and rectangular blocks, Comp. Graphics Image Process., 5 (1976) 353-381.

[31] M.Mahajan, K. Krithivasan, Hexagonal Cellular Automata, In *"A Perspective in theoretical Computer Science, Commemorative Volume for Gift Siromoney"*, Ed. R.Narasimhan, World Scientific, 1989, pp. 134-164.

[32] K.G. Subramanian, Hexagonal array grammars, Comp. Graphics Image Process., 10 (1979) 388-394.

[33] K.G. Subramanian, R. Saravanan, T. Robinson, P systems for array generation and application to kolam patterns, Forma, 22 (2007) 47-54.

[34] K. G. Subramanian, M. Geethalakshmi, A.K. Nagar, S.K. Lee, Two-dimensional picture grammar models, Proceedings of the 2nd European Modelling Symposium (EMS2008), Liverpool Hope university, England, 2008, pp. 263-267.

[35] K.G. Subramanian, M. Geethalakshmi, A. K. Nagar and S.K. Lee, Hexagonal picture languages, Accepted for presentation at the 5th Asian Mathematical Conference (AMC 2009), Malaysia, June 2009.

[36] K.G. Subramanian, D.L. Van, P. Helen Chandra and N.D. Quyen, Array grammars with contextual operations, Fundamenta Informaticae, 83 (2008) 411-428.

[37] K. S. Dersanambika, K. Krithivasan, C. Martin-Vide, K. G. Subramanian, Local and recognizable hexagonal picture languages. Int. J. Pattern Recogn. Artificial Intelligence 19 (2005) 853-871.

[38] K. S. Dersanambika, K. Krithivasan, C. Martin-Vide, K. G. Subramanian, Hexagonal pattern languages, Lecture Notes in Computer Science 3322, Springer-verlag (2004) 52-64.