Pure 2D Picture Grammars (P2DPG) and P2DPG with Regular Control

K.G. Subramanian[1,2], Atulya K. Nagar[2], and M. Geethalakshmi[3]

[1] School of Mathematical Sciences,
Universiti Sains Malaysia, 11800 Penang, Malaysia
kgs@usm.my
[2] Deanery of Business and Computer Sciences,
Liverpool Hope University, Hope Park, Liverpool L16 9JD UK
nagara@hope.ac.uk
[3] Department of Mathematics,
Dr. MGR Janaki College, Chennai 28, India

Abstract. In this note a new model of grammatical picture generation is introduced. The model is based on the notion of pure context-free grammars of formal string language theory. The resulting model, called Pure 2D context-free grammar (CFG), generates rectangular picture arrays of symbols. The generative power of this model in comparison to certain other related models is examined. Also we associate a regular control language with a Pure 2D CFG and notice that the generative power increases. Certain closure properties are obtained.

1 Introduction

Theoretical studies on digital pictures and picture analysis include syntactic techniques as one of the main areas of study. In the problem of generation and description of picture patterns considered as connected, digitized, finite arrays of symbols, syntactic methods have played a significant role on account of their structure-handling ability. Several picture language generating devices have been introduced in the literature based on generalizing to two dimensions different kinds of grammars like the Chomskian string grammars, the Lindenmayer systems (L systems) and so on and adapting the techniques and results of formal string language theory. See for example [5,6,14,1].

One of the earliest picture models was proposed by Siromoney et al [9], motivated by certain floor designs called "kolam" patterns. In this two-dimensional model, which we call as Siromoney matrix grammar, generation of rectangular arrays takes place 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. Recently there has been a renewed interest in the study of Siromoney matrix grammars [12,13].

Another very general rectangular array generating model, called extended controlled tabled L array system (ECTLAS) was proposed by Siromoney and

V.E. Brimkov, R.P. Barneva, H.A. Hauptman (Eds.): IWCIA 2008, LNCS 4958, pp. 330–341, 2008.

Siromoney [10], incorporating into arrays the developmental type of generation used in the well-known biologically motivated L-systems. 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.

Pure context-free grammars [4] which make use of only one kind of symbols, called terminal symbols, unlike the Chomskian grammars, have been investigated in formal string language theory for their language generating power and other properties. In this note we introduce a new two-dimensional grammar, called Pure 2D Context-free grammar (CFG), for picture array generation based on pure context-free rules. Unlike the models in [9,10], we allow rewriting any column or any row of the rectangular array rewritten and do not prescribe any priority of rewriting columns and rows as in [9] 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 model with those in [9,10,11,2]. Certain closure properties are obtained. Also we associate a regular control language with a Pure 2D CFG and notice that the generative power increases. Interpretation of the letter symbols in picture arrays by primitive patterns is a well-known technique to obtain interesting classes of "kolam" [9] pictures or "chain code" [3,12] pictures and so on. We indicate here chain code interpretation of the picture arrays generated by Pure 2D CF grammars.

2 Preliminaries

Let Σ be a finite alphabet. A word or string $w = a_1 a_2 \dots a_n \quad (n \geq 1)$ 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^* as horizontal words. For any word $w = a_1 a_2 \dots a_n$, we denote by w^T the vertical word

$$\begin{matrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{matrix}$$

We also define $(w^T)^T = w$. We set λ^T as λ itself. A rectangular $m \times n$ array M over Σ 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. $\Sigma^{**} - \{\lambda\} = \Sigma^{++}$. We denote respectively by $\circ$ and $\diamond$ the *column concatenation* and *row concatenation* of arrays in V^{**}. In contrast to the case of strings, these operations are partially defined, namely, for any $X, Y \in V^{**}$, $X \circ Y$ 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 [5,6] for array grammars. For notions of formal language theory we refer to [8]. We briefly recall pure context-free grammars [4] and the rectangular picture generating models in [9,10,11,1,2].

Definition 1 ([4]). *A pure context-free grammar 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 $G = (\{a, b, c\}, \{c \rightarrow acb\}, \{acb\})$ generates the language $\{a^n cb^n / n \geq 1\}$.

We restrict ourselves to recalling Tabled $0L$ array systems (T0LAS) introduced in [10] for generating rectangular picture arrays.

Definition 2. *A tabled $0L$ array system ($T0LAS$) 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 rightmost or leftmost column or the uppermost or lowermost 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 Tabled $0L$ array language ($T0LAL$) if and only if there exists a tabled $0L$ array system G such that $\mathcal{M}(G) = \{M | M_0 \Rightarrow^* M, M \in T^{**}\}$. The family of Tabled $0L$ array languages is denoted by $L(T0LAL)$.

In the 2D grammar model introduced in [9], 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 3. *A Siromoney matrix grammar is a $2-$tuple (G_1, G_2) where*

$G_1 = (H_1, I_1, P_1, S)$ is a regular or 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, context-free Siromoney matrix grammars if* G_1 *is regular, 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*

$$A_{i1} \cdots A_{in}$$

$$\Downarrow$$

$$a_{i1} \cdots a_{in}$$

$$B_{i1} \cdots B_{in}$$

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, CF Siromoney Matrix grammars by* $RML, CFML$ *respectively.*

The regular/context-free Siromoney Matrix grammars were extended in [11] 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.

Based on a well known characterization of recognizable string languages in terms of local languages and projections, an interesting model of Tiling Recognizable languages describing rectangular picture arrays was introduced in [1,2]. We now recall briefly 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 4. *Let* Γ *be a finite alphabet. A two-dimensional language or picture array language* $L \subseteq \Gamma^{**}$ *is local if there exists a finite set* Θ *of tiles over the alphabet* $\Gamma \cup \{\#\}$ *such that* $L = \{p \in \Gamma^{**} / B_{2\times 2}(\hat{p})\} \subseteq \Gamma^{**}$*. The family of local picture array languages will be denoted by LOC.*

Definition 5. *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.*

3 Pure 2D Picture Grammars

We now introduce a new two-dimensional grammar 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 column of symbols in parallel by equal length strings and by using only terminal symbols as in a pure string grammar. This new model is related to the model T0LAS in [10] in the sense that a column or row of symbols of a rectangular array is rewritten in parallel. This feature incorporates into arrays the parallel rewriting feature of the well-known and widely investigated Lindenmayer systems [7]. But the difference between this new model and the T0LAS in [10] 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 row of symbols. We now define the new grammar model.

Definition 6. *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* $a \to \alpha, a \in \Sigma, \alpha \in \Sigma^*$ *such that for any two rules* $a \to \alpha, b \to \beta$ *in* t_{c_i}, *we have* $|\alpha| = |\beta|$ *where* $|\alpha|$ *denotes the length of* $|\alpha|$;
Each $t_{r_j}, (1 \leq j \leq n)$, *called a row table, is a set of context-free rules of the form* $c \to \gamma^T, c \in \Sigma$ *and* $\gamma \in \Sigma^*$ *such that for any two rules* $c \to \gamma^T, d \to \delta^T$ *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 array languages generated by Pure 2D Context-free grammars is denoted by P2DCFL.*

$$M_0 \Rightarrow \begin{array}{cccc} x & b & b & x \\ z & y & y & z \\ x & b & b & x \end{array} \Rightarrow \begin{array}{cccc} x & b & b & x \\ x & b & b & x \\ z & y & y & z \\ x & b & b & x \\ x & b & b & x \end{array} \Rightarrow \begin{array}{cccc} x & b & b & x \\ x & b & b & x \\ x & b & b & x \\ z & y & y & z \\ x & b & b & x \\ x & b & b & x \\ x & b & b & x \end{array} \Rightarrow \begin{array}{ccccc} x & b & b & b & x \\ x & b & b & b & x \\ x & b & b & b & x \\ z & y & y & y & z \\ x & b & b & b & x \\ x & b & b & b & x \\ x & b & b & b & x \end{array} = M_1$$

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

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

Fig. 2. A picture array M_2

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

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

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:

Each of the arrays occurring in the derivation given belongs to the picture language generated by G_1.

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

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

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 $(yz \dots z)^T$.

4 Comparisons and Closure Results

We now compare the new 2D grammar model introduced here with those in [9,10,1,2].

Theorem 1. *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 over a single symbol a of all sizes $m \times n (m, n \geq 1)$ is generated by a regular Siromoney matrix grammar G. In fact the language of horizontal words generated in the first phase of G_1 is $\{S_1^n / n \geq 1\}$ where S_1 is an intermediate symbol and the language of vertical words generated by S_1 in the second phase is $\{(a^n)^T / n \geq 1\}$. 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 axiom array a. The incomparability with $CFML$ is due to the fact that it is known [9] that the picture languages in examples 2 and 3 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 $zy \ldots yz$ and in example 3, each of the generated pictures has an equal number of columns to the left and right of the middle column $(yz \ldots z)^T$. On the other hand a picture language consisting of rectangular arrays of the form $M_1 \circ M_2$ where M_1 and M_2 are rectangular arrays over the symbols a, b respectively with equal number of columns can be generated by a context-free Siromoney matrix grammar with the language of horizontal words $S_1^n S_2^n$ (S_1, S_2 are intermediate symbols) in the first phase and S_1, S_2 generating vertical words over a, b respectively. This picture language, cannot be generated by any Pure 2D context-free grammar since the string language $\{a^n b^n / n \geq 1\}$ is not a pure CFL [4] and an argument similar to this can be done in the two-dimensional case also. The incomparability with RML can be seen by considering a picture language with rectangular arrays each row of which is a word in $a^3 b^3 (ab)^*$, known [4] to be not a Pure CFL.

Theorem 2. *The family of P2DCFL is incomparable with the families of $TRML$ and $TCFML$ but not disjoint with these families.*

Proof. In view of the proper inclusions $RML \subset TRML, CFML \subset TCFML$ and incomparability (Theorem 1) 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 picture arrays in Figure 1 each has an equal number of rows above and below the middle row $zy \ldots yz$.

Theorem 3. *Every language in the family $L(T0LAL)$ is a coding of a Pure 2D CFL.*

Proof. Let L be a picture array language generated by a $T0LAS$ [10] $G = (T, \mathcal{P}, M_0)$. We construct a Pure 2D CFG 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 $a \rightarrow a_1 a_2 \cdots a_m b, A, B \in T', a_i (1 \leq i \leq m), b \in T$ in a right table t, is replaced by a rule $A \rightarrow a_1 a_2 \cdots a_m B, A, B \in T', a_i (1 \leq i \leq m), b \in T$. Each rule of the form $a \rightarrow a_1 a_2 \cdots a_m b, A, B \in T', a_i (1 \leq i \leq m), b \in T$ in a down table t, is replaced by a rule $A \rightarrow (a_1 a_2 \cdots a_m B)^T, A, B \in T', a_i (1 \leq i \leq m), b \in T$. 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 replaced 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 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 4. *The family of Pure 2D Context-free languages is incomparable with LOC and REC.*

Proof. The language of square picture arrays with $1s$ in the main diagonal and $0s$ in other positions is known [1] to be in LOC and the language of square picture arrays over $0s$ is known [1] to be in REC but both these languages cannot be generated by any P2DCFG for the simple reason that the language of square arrays cannot be generated by a P2DCFG as the rewriting of a column and of a row are independent. 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 a (M is a picture array with only one row) is generated by a P2DCFG with a column rule $c \to aca$ but L_1 is known [1] to be not in REC and hence not in LOC.

It is a well-known tool in formal language theory [8] 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 7. *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* $w = l_1 l_2 \ldots l_m$ *then the tables of rules with labels* $l_1, l_2, \ldots l_m$ *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 the family of picture array languages generated by Pure 2D Context-free grammars with a regular control by* $(R)P2DCFL$*.*

Lemma 1. *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, generates picture arrays as shown in Figure 1 but with sizes* $(2n+1) \times (n+2), n \geq 1$*, and thus having a proportion between the height (the number of rows in a picture array) and width (the number of columns in a picture array). In fact the number of rows above and below the middle row* $zy \ldots yz$ *equals the number of columns between the leftmost and rightmost columns, namely,* $(x \ldots xzx \ldots x)^T$*.*

Proof. The tables of rules generating the picture array language in example 2 are $t_{c_1} = \{b \rightarrow bb, y \rightarrow yy\}$, $t_{r_1} = \left\{ y \rightarrow \begin{matrix} b \\ y \\ b \end{matrix}, z \rightarrow \begin{matrix} x \\ z \\ x \end{matrix} \right\}$. Since the control language on the labels of the tables consists of words $\{(l_1 l_2)^n / n \geq 1\}$, 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 in the statement of the theorem.

Theorem 5. *The family of P2DCFL is properly contained in* $(R)P2DCFL$.

Proof. The containment follows since every P2DCFL is generated by a P2DCFG G and the regular control language is $Lab(\mathcal{C})^*$ itself. The proper containment is a consequence of the Lemma 1.

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.

Theorem 6. *The picture array language consisting of square arrays over one symbol a is generated by a P2DCFG with a regular control.*

Proof. The P2DCFG $(\{a\}, \{t_{c_1}\}, \{t_{r_1}\}, a)$ where $t_{c_1} = \{a \rightarrow aa\}$, $t_{r_1} = \{a \rightarrow \begin{matrix} a \\ a \end{matrix}\}$ 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_1}, t_{r_1} can be seen to generate the picture array language consisting of square arrays over one symbol a.

We now examine some of the closure properties of P2DCFL. We also consider 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 7. *The family of P2DCFL is not closed under union, column catenation, row catenation but is closed under projection, transposition, reflection about the base and reflection about the leg.*

Proof. Let the alphabet be $\{a, b, c, x, y\}$. Non-closure under union follows by the fact that $L_1 = \{X_1 \circ (c^n)^T \circ Y_1 / X_1 \in \{a\}^{++}, Y_1 \in \{b\}^{++}, |X_1|_c = |Y_1|_c\}$ where $|X|_c$ stands for the number of columns of X, 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 $L_2 = \{X_2 \circ (c^n)^T \circ Y_2 / X_2 \in \{x\}^{++}, Y_2 \in \{y\}^{++}, |X_2|_c = |Y_2|_c\}$ 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 $c \to acb$ and $c \to xcy$. 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 $c \to acb$ and $c \to xcy$ 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 $a \to \alpha$ in a column table of G, the rule $a \to \alpha^T$ $(\alpha \in \Sigma^{**})$ is added in the corresponding row table of G' and likewise for a rule $b \to \beta^T$ $(\beta \in \Sigma^{**})$ in a row table of G, the rule $b \to \beta$ 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.

5 Interpretations of Picture Arrays

The idea of interpreting letter symbols in a picture array by primitive patterns is a well-known technique to obtain interesting classes of "kolam" [9] pictures or chain code [3] pictures and so on. We can employ here this technique to generate such picture patterns as an application of the Pure 2D CF grammars. Each symbol of a rectangular array is considered to occupy a unit square in the rectangular grid so that each row or column of symbols in the array respectively occupies a horizontal or 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 Pure2DCFG 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. 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, using two chain code primitives, namely, | , – the

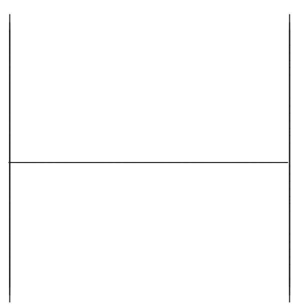

Fig. 3. The alphabetic letter H

interpretation mapping i by $i(x) = i(z) = |$, $i(y) = -$ and $i(b) = blank$ then the interpretation $i(M_1)$ of M_1 in Figure 1 will give a picture of the alphabetic letter H (Figure 3).

Likewise if the primitive picture patterns are those used in "kolam" pictures, we can obtain "kolam" patterns from Pure 2D CFL via suitable interpretation.

6 Conclusion

The picture array generating model based on pure context-free grammars introduced here does not prescribe a priority of rewriting column or row unlike [9,11] and does not allow rewriting only the borders of an array as in [10]. But it requires a "control" to maintain a "proportion" between the number of columns and the number of rows. In the case of string grammars, the class of pure CFLs [4], is included in the class of CFLs. Here we have seen that the family of Pure 2D CFLs becomes incomparable with the family of CFMLs introduced in [9]. But we can extend the model of Pure 2D CFG by allowing nonterminal symbols as well and this might increase the power of this model. It remains to be seen in future whether this kind of an extension will be more powerful than the 2D model in [10]. Also it remains to examine whether other properties [4] of pure string languages carry over to the Pure 2D Context-free grammars.

Acknowledgements

The authors are grateful to the referees for their very useful comments which improved the presentation of the paper.

References

1. Giammarresi, D., Restivo, A.: Two-dimensional languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 3, pp. 215–267. Springer, Heidelberg (1997)
2. Giammarresi, D., Restivo, A.: Recognizable Picture Languages. International Journal of Pattern Recognition and artificial Intelligence (Special issue on Parallel Image Processing), Nivat, M., Saoudi, A., Wang, P.S.P. (eds.), 31–46 (1992)
3. Maurer, H.A., Rozenberg, G., Welzl, E.: Chain-code picture languages. Lecture notes in Computer science, vol. 153, pp. 232–244. Springer, Heidelberg (1983)
4. Maurer, H.A., Salomaa, A., Wood, D.: Pure Grammars. Information and Control 44, 47–72 (1980)
5. Rosenfeld, A.: Picture Languages - Formal Models for Picture Recognition. Academic Press, New York (1979)
6. Rosenfeld, A., Siromoney, R.: Picture languages - a survey. Languages of design 1, 229–245 (1993)
7. Rozenberg, G., Salomaa, A.: The Mathematical Theory of L systems. Academic Press, New York (1980)
8. Salomaa, A.: Formal languages. Academic Press, London (1973)

9. Siromoney, G., Siromoney, R., Krithivasan, K.: Abstract families of matrices and picture languages. Computer Graphics and Image Processing 1, 234–307 (1972)
10. Siromoney, R., Siromoney, G.: Extended Controlled Tabled L- arrays. Information and Control 35(2), 119–138 (1977)
11. Siromoney, R., Subramanian, K.G., Rangarajan, K.: Parallel/Sequential rectangular arrays with tables. International Journal of Computer Mathematics, 143–158 (1977)
12. Stiebe, R.: Picture generation using matrix systems. Journal of Information Processing and Cybernetics 28, 311–327 (1992)
13. Stiebe, R.: Slender Siromoney Matrix Languages. In: Proceedings of the 1^{st} International Conference on Language and Automata: Theory and Applications, Tarragona, Spain (2007)
14. Wang, P.S.P.: Array grammars, Patterns and recognizers. World Scientific, Singapore (1989)