Proceedings of Grammar Systems Week 2004, edited by E. Csuhaj-Varjú and Gy. Vaszil, MTA SZTAKI, Budapest, pages 276 - 286, 2004.

Image Splicing Grammar Systems

A. Roslin Sagaya Mary

Research Group on Mathematical Linguistics, Rovira i Virgili University,
Pl. Imperial Tarraco 1, 43005 Tarragona, Spain
writetoroslin@yahoo.co.uk

K. G. Subramanian

Department of Mathematics, Madras Christian College
Tambaram, Chennai 600 059, India
kgsmani1948@yahoo.com

K.S.Dersanambika

Department of Computer Science and Engineering,
Indian Institute of Technology,Madras
Chennai 600 036,India
dersanapdf@yahoo.com

Abstract

Motivated by the study of Dassow and Mitrana [3], we consider Grammar Systems that describe Images or Pictures of rectangular arrays with the components consisting of two-dimensional Grammars and domino splicing rules [8] with the grammars working in parallel and splicing rules acting on arrays of two components, yielding rectangular arrays of symbols. The resulting systems are called Image Splicing Grammar Systems (ISGS). Certain Properties of ISGS with different component grammars are obtained.

1 Introduction

The theory of Grammar systems is a well-investigated field of formal language theory, providing a theoretical framework for modelling various kinds of multi-agent systems at the symbolic level [2]. A grammar system consists of several grammars or other language identifying mechanisms, that cooperate according to some well-defined protocol. The components of the system correspond to the agents, the current string(s) in generation to a symbolic environment, and the system's behaviour is represented by the language. Among a variety of grammar system models, Parallel Communicating Grammar Systems, in which the components are generative grammars working on their own sentential forms in parallel and communicating with each other by sending their sentential forms by request, have been of intensive study [1, 2].

On the other hand, there has been a lot of interest in the study of formal language theory applied to DNA computing. Head [6] defined splicing systems motivated by the behaviour of DNA sequences under the action of restriction enzymes and ligases. The splicing systems make use of a new operation, called splicing on strings of symbols. Theoretical investigation of splicing on strings has been extensively done by different researchers [7].

A new type of Parallel Communicating grammar systems has been introduced in [3] by replacing communication by splicing of strings.

In syntactic approaches to generation and recognition of image or picture patterns, considered as digitized arrays, several two-dimensional grammars have been proposed and studied [9]. As a simple and effective extension of the operation of splicing on strings a new method of splicing on images of rectangular arrays is introduced in [8] . The idea here is that two rectangular arrays are column spliced or row spliced by using the domino splicing rules in parallel.

Freund [4] has introduced and investigated cooperating distributed array grammar systems extending the concept of cooperation in string grammar systems and using array grammars. Here, motivated by the study of Dassow and Mitrana [3], we consider Grammar Systems that describe Images or Pictures of rectangular arrays. The components of the Grammar system consist of two-dimensional Grammars [5, 10, 9] and domino splicing rules [8] with the grammars working in parallel and splicing rules acting on arrays of two components yielding rectangular arrays of symbols. The resulting systems are called Image Splicing Grammar Systems. Different component grammars such as Regular Matrix grammars [5, 10], Context-free Matrix grammars [5, 10], are considered and properties such as generative power, comparison etc. are obtained.

2 Preliminaries

The basic notions and notations on arrays are now recalled [5, 10].

Let Σ be a finite alphabet. Σ^* is the set of all words over Σ including the empty word λ. An image or a picture A over Σ is a rectangular $m \times n$ array of elements of Σ of the form

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

or in short $A = [a_{ij}]_{m\times n}$. We write an array A without enclosing it in square brackets when there is no confusion. The set of all images is denoted by Σ^{**}. A picture language or a two dimensional language over Σ is a subset of Σ^{**}.

Definition 1 For an array A of dimension $m \times n$ and an array B of dimension $m' \times n'$, the column catenation $A\Phi B$ is defined only when $m = m'$ and the row catenation $A\Theta B$ is defined only when $n = n'$.

Informally speaking, in row catenation $A\Theta B$, B is attached below A. In column catenation $A\Phi B$, B is attached to the right of X. We refer to [5, 10]for a formal definition of column and row catenations of rectangular arrays.

Definition 2 (Two-Dimensional Matrix Grammars) A $2D$ **matrix grammar** is a $2-$tuple (G_1, G_2) where
$G_1 = (H_1, I_1, P_1, S)$ is a Regular, CF or CS 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, context sensitive $2D$ matrix grammars if G_1 is regular, context-free, context sensitive 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 all matrices 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, CS 2DMatrix grammars by 2DRML, 2DCFML, 2DCSML respectively.

We next recall the notion of splicing on strings [6, 7].

Definition 3 Let V be an alphabet and $\#, \$$ two special symbols not in V. A splicing rule over V is a string of the form $r = u_1\#u_2\$u_3\#u_4$, where $u_i \in V^*, 1 \leq i \leq 4$. For such a rule r and strings $x, y, z \in V^*$, we write

$(x, y) \vdash_r z$ iff $x = x_1u_1u_2x_2,\ y = y_1u_3u_4y_2,\ z = x_1u_1u_4y_2$

for some $x_1, x_2, y_1, y_2 \in V^*$. We say that z is obtained by splicing x, y, as indicated by the rule r; u_1u_2 and u_3u_4 are called the sites of the splicing.

We now recall the notions of domino splicing rules and Splicing of arrays using these rules [8].

Definition 4 Let V be an alphabet. $\#, \$$ are two special symbols, not in V. A domino over V is of the form $\boxed{\begin{array}{c} a \\ \hline b \end{array}}$ or $\boxed{\begin{array}{c|c} a & b \end{array}}$ for some $a, b \in V$

A domino column splicing rule over V is of the form $r = \alpha_1 \# \alpha_2 \$ \alpha_3 \# \alpha_4$ where each $\alpha_i = \boxed{\begin{array}{c} a \\ \hline b \end{array}}$ for some $a, b \in V \cup \{\#\}$.

A domino row splicing rule over V is of the form $r = \beta_1 \# \beta_2 \$ \beta_3 \# \beta_4$ where each $\beta_i = \boxed{\begin{array}{c|c} a & b \end{array}}$ for some $a, b \in V \cup \{\#\}$.

We refer to $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ of a column splicing rule $r = \alpha_1 \# \alpha_2 \$ \alpha_3 \# \alpha_4$ as the first, second, third and fourth dominoes of r respectively. Similarly for a row splicing rule $r = \beta_1 \# \beta_2 \$ \beta_3 \# \beta_4$. $\beta_1, \beta_2, \beta_3, \beta_4$ are the first, second, third and fourth dominoes of r respectively.

Given two arrays X and Y of sizes $m \times p$ and $m \times q$ respectively,

$$X = \begin{array}{cccccc} a_{11} & \cdots & a_{1,j} & a_{1,j+1} & \cdots & a_{1p} \\ a_{21} & \cdots & a_{2,j} & a_{2,j+1} & \cdots & a_{2p} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{m,j} & a_{m,j+1} & \cdots & a_{mp} \end{array} ,$$

$$Y = \begin{array}{cccccc} b_{11} & \cdots & b_{1,k} & b_{1,k+1} & \cdots & b_{1q} \\ b_{21} & \cdots & b_{2,k} & b_{2,k+1} & \cdots & b_{2q} \\ \vdots & \ddots & \cdots & \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{m,k} & b_{m,k+1} & \cdots & b_{mq} \end{array}$$

a_{ir}, $b_{is} \in V$, for $1 \leq i \leq m$, $1 \leq r \leq p$, $1 \leq s \leq q$. We write $(X, Y) \quad |\underline{\Phi} \quad Z$ if there is a sequence of column splicing rules $r_1, r_2, ...r_m$ (not necessarily all different) such that

$$r_i = \boxed{\begin{array}{c} a_{i,j} \\ \hline a_{i+1,j} \end{array}} \# \boxed{\begin{array}{c} a_{i,j+1} \\ \hline a_{i+1,j+1} \end{array}} \$ \boxed{\begin{array}{c} b_{i,k} \\ \hline b_{i+1,k} \end{array}} \# \boxed{\begin{array}{c} b_{i,k+1} \\ \hline b_{i+1,k+1} \end{array}}$$

for all i, $1 \leq i \leq m-1$ and for some j, k $1 \leq j \leq p-1$, $1 \leq k \leq q-1$ and

$$Z = \begin{array}{cccccc} a_{11} & \cdots & a_{1,j} & b_{1,k+1} & \cdots & b_{1q} \\ a_{21} & \cdots & a_{2,j} & b_{2,k+1} & \cdots & b_{2q} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{m,j} & b_{m,k+1} & \cdots & b_{mq} \end{array}$$

In other words, we can imagine that a 2×1 window is moved down the j^{th} column of X The sequence of dominoes collected are the first dominoes of the rules $r_1, r_2, ..., r_m$ (not all necessarily different). When a 2×1 window is moved down the $j + 1^{st}$ column of X the sequence of dominoes collected are the second dominoes of the rules $r_1, r_2, ..., r_m$. Likewise for the k^{th} and $k + 1^{st}$ columns of Y When such rules exist in the system, the column splicing of the arrays X and Y amounts to the array X being vertically "cut" between j^{th} and $j + 1^{st}$ columns and the array Y between k^{th} and $k + 1^{st}$ columns and the resulting left subarray of X "pasted" (column catenated) with the right subarray of Y to yield Z We now say that Z is obtained from X and Y by domino column splicing in parallel.

We can similarly define row splicing operation of two arrays U and V of sizes $p \times n$ and $q \times n$, using row splicing rules to yield an array W.

$$U = \begin{array}{cccc} c_{11} & c_{12} & \cdots & c_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{r,1} & c_{r,2} & \cdots & c_{r,n} \\ c_{r+1,1} & c_{r+1,2} & \cdots & c_{r+1,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{p1} & c_{p2} & \cdots & c_{pn} \end{array}, \qquad V = \begin{array}{cccc} d_{11} & d_{12} & \cdots & d_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{s,1} & d_{s,2} & \cdots & d_{s,n} \\ d_{s+1,1} & d_{s+1,2} & \cdots & d_{s+1,n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{q1} & d_{q2} & \cdots & d_{qn} \end{array}$$

c_{rj}, $d_{sj} \in V$, for $1 \leq j \leq n$, $1 \leq r \leq p$, $1 \leq s \leq q$.

We write $(U, V) \;\; |\!\stackrel{\Theta}{-} \;\; W$ if there is a sequence of row splicing rules $r_1, r_2, \cdots r_n$ (not necessarily all different) such that

$$r_j = \boxed{c_{r,j} \mid c_{r,j+1}} \; \# \; \boxed{c_{r+1,j} \mid c_{r+1,j+1}} \; \$ \; \boxed{d_{s,j} \mid d_{s,j+1}} \; \# \; \boxed{d_{s+1,j} \mid d_{s+1,j+1}}$$

for all j, $1 \leq j \leq n-1$ and for some r, s $1 \leq r \leq p-1$, $1 \leq s \leq q-1$ and

$$W = \begin{array}{cccc} c_{11} & c_{12} & \cdots & c_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{r,1} & c_{r,2} & \cdots & c_{r,n} \\ d_{s+1,1} & d_{s+1,2} & \cdots & d_{s+1,n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{q1} & d_{q2} & \cdots & d_{qn} \end{array}$$

As done for the column splicing of arrays, we can imagine 1×2 windows being moved over respective rows. The row splicing of the arrays U and V can be thought of as U being horizontally "cut" between the r^{th} and $r+1^{st}$ rows and V between s^{th} and $s+1^{st}$ rows and the upper subarray of U "pasted" (row catenated) to the lower subarray of V to yield W We now say that W is obtained from U and V by domino row splicing in parallel.

We illustrate with an example.

Example 1 Let $V = \{a, b\}$,

$$R_c = \{p_1 : \boxed{\begin{array}{c} a \\ \hline b \end{array}} \; \# \; \$ \; \# \; \boxed{\begin{array}{c} b \\ \hline a \end{array}}$$

$$p_2 : \boxed{\begin{array}{c} b \\ \hline a \end{array}} \; \# \; \$ \; \# \; \boxed{\begin{array}{c} a \\ \hline b \end{array}} \; \}$$

$$R_r = \{q_1 : \boxed{a \mid b} \; \# \; \$ \; \# \; \boxed{b \mid a}$$

$$q_2 : \boxed{b \mid a} \; \# \; \$ \; \# \; \boxed{a \mid b} \; \}$$

Column splicing in parallel of an array with itself using the rules given is shown below:

$$\begin{array}{cc|}a & b\\ b & a\end{array}\ \begin{array}{|cc}a & b\\ b & a\end{array}\ \ \mid\!\underline{\ \Phi\ }\ \ \begin{array}{cccc}a & b & a & b\\ b & a & b & a\end{array}$$

Likewise, row splicing in parallel of an array with itself using the rules given is shown below:

$$\begin{array}{cccc}a & b & a & b\\ b & a & b & a\\ \hline\end{array}\ \begin{array}{cccc}\hline a & b & a & b\\ b & a & b & a\end{array}\ \ \mid\!\underline{\ \Theta\ }\ \ \begin{array}{cccc}a & b & a & b\\ b & a & b & a\\ a & b & a & b\\ b & a & b & a\end{array}$$

A vertical bar '| ' or a horizontal bar '—' indicates the place where splicing is done.

3 Image Splicing Grammar System

We now introduce the notion of Image Splicing grammar system in which the component grammars are 2DMatrix grammars.

Definition 5 An Image Splicing Grammar system is a construct
$\Gamma = (V_h, \Sigma_I, V_v, T, (S_1, R_1^h, R_1^v), ..., (S_n, R_n^h, R_n^v), M)$ where,

V_h is a finite set of variables called horizontal variables;
V_v is a finite set of variables called vertical variables;
$\Sigma_I \subseteq V_v$ is a finite set of intermediates;
T is a finite set of terminals;
$S_i, 1 \leq i \leq n$ is the start symbol of the corresponding horizontal component;
$R_i^h, 1 \leq i \leq n$ is a finite set of rules called horizontal productions
and the rules can be regular or context free or context sensitive;
$R_i^v, 1 \leq i \leq n$ is a finite set of right linear rules called vertical productions;
M is a finite set of domino column or row splicing rules of the form

$$m = \alpha_1 \ \# \ \alpha_2 \ \$ \ \alpha_3 \ \# \ \alpha_4 \text{ or } \beta_1 \ \# \ \beta_2 \ \$ \ \beta_3 \ \# \ \beta_4$$

where $\alpha_i = \begin{array}{|c|}\hline a\\ \hline b\\ \hline\end{array}$ and $\beta_i = \begin{array}{|c|c|}\hline c & d\\ \hline\end{array}$ for some $a, b, c, d \in V_v \cup \{T\} \cup \{\lambda\}$.

The derivations take place in two phases as follows :

Each component grammar generates a word called intermediate word, over intermediates starting from its own start symbol and using its horizontal production rules ; the derivations in this phase are done with the component grammars working in parallel.

In the second phase any of the following steps can take place :

(i) each component grammar can rewrite as in a two dimensional matrix grammar using the vertical rules, starting from its own intermediate word generated in the first phase. (The component grammars rewrite in parallel).Note that the component grammars together terminate or together continue rewriting in the vertical direction.

(ii) At any instant the array X generated in the i^{th} component for some $1 \leq i \leq n$ and the array Y generated in the j^{th} component for some $1 \leq j \leq n$ can be spliced using column / row domino splicing rules as in definition 4, thus yielding array Z in i^{th} component and W in the j^{th} component; In fact Z will have a prefix of X column concatenated with a suffix of Y and W will have a prefix of Y, column concatenated with a suffix of X, the prefixes and suffixes being given by the splicing rules. In any other components (other than i^{th}, j^{th} components), the arrays generated at this instant will remain unchanged during this splicing process.

There is no priority between steps (i) and (ii).

The language $L_i(\Gamma)$ generated by the ith component of Γ consists of all arrays, generated over T, by the derivations described above.

This language will be called the *individual language* of the system and we may choose this to be the language of the first component and $L_t(\Gamma) = \bigcup_{i=1}^{n} L_i(\Gamma)$ as the *total language*. The family of individual languages generated by ISGS with n components of type X for $X \in \{REG, CF\}$ is denoted by $I_{isgs}L_n(X)$, and the corresponding family of total languages by $T_{isgs}L_n(X)$ respectively and $Y_{isgs}L_n(X)$ when $Y \in \{I, T\}$. We basically deal with individual languages although the results obtained apply to total languages as well.

Example 2 Let $\Gamma = (\{S, X\}, \{A, B, C\}, \{A, B, C, D\} \{., x\}, (S, R^h, R^v), (S, R^h, R^v), (S, R^h, R^v), M)$
where
$R^h = \{S \to AX, X \to BX, X \to C$
$R^v = \{A \to xA, A \to x, B \to xD, D \to .D, D \to x, C \to xC, C \to x\}$.

$$M = \{ \begin{array}{|c|}\hline x \\ \hline . \\ \hline \end{array} \;\#\; \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array} \;\$\; \#\; \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array}$$

$$\begin{array}{|c|}\hline . \\ \hline . \\ \hline \end{array} \;\#\; \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array} \;\$\; \#\; \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array}$$

$$\begin{array}{|c|}\hline . \\ \hline D \\ \hline \end{array} \;\#\; \begin{array}{|c|}\hline x \\ \hline C \\ \hline \end{array} \;\$\; \#\; \begin{array}{|c|}\hline x \\ \hline A \\ \hline \end{array}$$

$$\begin{array}{|c|c|}\hline x & . \\ \hline \end{array} \;\#\; \begin{array}{|c|c|}\hline A & D \\ \hline \end{array} \;\$\; \#\; \begin{array}{|c|c|}\hline x & x \\ \hline \end{array}$$

$$\begin{array}{|c|c|}\hline . & . \\ \hline \end{array} \;\#\; \begin{array}{|c|c|}\hline D & D \\ \hline \end{array} \;\$\; \#\; \begin{array}{|c|c|}\hline x & x \\ \hline \end{array}$$

$$\begin{array}{|c|c|}\hline . & x \\ \hline \end{array} \;\#\; \begin{array}{|c|c|}\hline D & D \\ \hline \end{array} \;\$\; \#\; \begin{array}{|c|c|}\hline x & x \\ \hline \end{array} \}$$

The horizontal rules in a component generate intermediate words of the form AB^nC with the same value of $n \geq 1$ at a time. The vertical rules of the components generate from an intermediate word rectangle pictures of (.)'s surrounded or bordered by x's except the bottom border which will be of the form AD^nC. At this stage with domino splicing rules, column or row splicing of the array in a component with the array in another component can take place before rewriting is terminated in the components with terminating vertical rules. In fact any picture generated in the individual language of this Image splicing grammar system will be either (i) rectangular pictures in which any row, except the first and the last, will be of the form $(x(.)^n)^k x$ for some $k \epsilon \{1, 2, 3\}$ or (ii)rectangular pictures in which any column, except the first and the last, will have a similar feature or (iii)simply a column of $x's$. Two such pictures obtained are shown in Figures $1(a)$ and $1(b)$.

```
x x x x x x x x x x x x x x x x
x . . . . x . . . . x . . . . x
x . . . . x . . . . x . . . . x
x . . . . x . . . . x . . . . x
x x x x x x x x x x x x x x x x
```

Figure $1(a)$. A Picture of Example 2

```
x x x x x
x . . . x
x . . . x
x . . . x
x x x x x
x . . . x
x . . . x
x . . . x
x . . . x
x x x x x
x . . . x
x . . . x
x . . . x
x . . . x
x x x x x
```

Figure $1(b)$. Another Picture of Example 2

Example 3

Let $\Gamma = (\{S, X\}, \{A, B, E\}, \{A, B, C, D, E\} \{., x\}, (S, R^h, R^v), (S, R^h, R^v), (S, R^h, R^v), M)$ where

$R^h = \{S \rightarrow EXE, X \rightarrow AXB, X \rightarrow AB\}$
$R^v = \{A \rightarrow aC, C \rightarrow .C, C \rightarrow a,$
$\quad B \rightarrow bD, D \rightarrow .D, D \rightarrow b,$

$E \rightarrow xE, E \rightarrow x\}$

and
$$M = \{ \begin{array}{|c|}\hline b \\ \hline . \\ \hline \end{array} \; \# \; \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array} \; \$ \; \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array} \; \# \; \begin{array}{|c|}\hline a \\ \hline . \\ \hline \end{array}$$
$$\begin{array}{|c|}\hline . \\ \hline . \\ \hline \end{array} \; \# \; \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array} \; \$ \; \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array} \; \# \; \begin{array}{|c|}\hline . \\ \hline . \\ \hline \end{array}$$
$$\begin{array}{|c|}\hline . \\ \hline D \\ \hline \end{array} \; \# \; \begin{array}{|c|}\hline x \\ \hline E \\ \hline \end{array} \; \$ \; \begin{array}{|c|}\hline x \\ \hline E \\ \hline \end{array} \; \# \; \begin{array}{|c|}\hline . \\ \hline C \\ \hline \end{array} \}$$

The horizontal rules generate in a component intermediate words of the form EA^nB^nE with the same value of $n \geq 1$ at a time. The vertical rules of the components generate from an intermediate word rectangle pictures of (.)'s bordered on the top by words of the form xa^mb^mx, on the bottom by words of the form EC^mD^mE , the leftmost column being a column of x's ending with E and the rightmost column being a column of x's ending with E. At this stage with domino splicing rules, column splicing of the array in a component with the array in another component can take place before rewriting is terminated in the components with terminating vertical rules. One such picture obtained is shown in Figure 2.

$$\begin{array}{cccccccccccccc}
\mathbf{x} & a & a & b & b & a & a & b & b & a & a & b & b & \mathbf{x} \\
\mathbf{x} & . & . & . & . & . & . & . & . & . & . & . & . & \mathbf{x} \\
\mathbf{x} & . & . & . & . & . & . & . & . & . & . & . & . & \mathbf{x} \\
\mathbf{x} & . & . & . & . & . & . & . & . & . & . & . & . & \mathbf{x} \\
\mathbf{x} & a & a & b & b & a & a & b & b & a & a & b & b & \mathbf{x}
\end{array}$$

Figure 2. A Picture of Example 3.

Theorem 1 For $Y \in \{I, T\}$,

1. $2DRML = Y_{isgs}L_1(REG)$
2. $2DRML \subset Y_{isgs}L_2(REG)$
3. $2DCFML = Y_{isgs}L_1(CF)$
4. $2DCFML \subset Y_{isgs}L_2(CF)$

Statements (1) and (3) are obvious. The proper inclusion in statement (2) is a consequence of Example 2. In fact the pictures in Figures $1(a)$ and $1(b)$ cannot be generated by any $2DRMG$ as both the rules in both the horizontal and vertical phases are only regular rules. Likewise the proper inclusion in statement (4) is a consequence of Example 3 since the rules in the horizontal phase of a $2DCFMG$ are only CF rules and so the pictures as in Figure 2 require CS rules in the first phase.

Example 4 Let $\Gamma = (\{S_1, \cdots S_n, X\}, \{A_1, \cdots, A_n, B_1, \cdots, B_n, C_1, \ldots, C_n, \},$
$\{A_1, \cdots, A_n, B_1, \cdots, B_n, C_1, \ldots, C_n, D_1, \ldots, D_n\},$

$\{., x, a, b\}, (S, R^{h_1}, R^{v_1}), (S, R^{h_2}, R^{v_2}), \cdots, (S, R^{h_n}, R^{v_n}), M)$
where

$R^{h_1} = \{S_1 \to A_1X, X \to B_1X, X \to C_1$
$R^{v_1} = \{A_1 \to xA_1, A_1 \to x, B_1 \to aD_1, D_1 \to .D_1, D_1 \to a, C_1 \to xC_1, C_1 \to x, D_i \to a \ if \ i \geq 2 \ and \ i \ odd, D_i \to b \ if \ i \geq 2 \ and \ i \ even, C_i \to x\}.$

For$i > 1$ and i even
$R^{h_i} = \{S_i \to A_iX, X \to B_iX, X \to C_i$
$R^{v_i} = \{A_i \to xA_i, A_i \to x, B_i \to bD_i, D_i \to .D_i, C_i \to xC_i, \}.$

For$i > 1$ and i odd
$R^{h_i} = \{S_i \to A_iX, X \to B_iX, X \to C_i$
$R^{v_i} = \{A_i \to xA_i, A_i \to x, B_i \to aD_i, D_i \to .D_i, C_i \to xC_i, \}.$

$$M = \{ \begin{array}{|c|}\hline a \\ \hline . \\ \hline \end{array} \ \# \ \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array} \ \$ \ \# \ \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array}$$
$$\begin{array}{|c|}\hline b \\ \hline . \\ \hline \end{array} \ \# \ \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array} \ \$ \ \# \ \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array}$$
$$\begin{array}{|c|}\hline . \\ \hline . \\ \hline \end{array} \ \# \ \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array} \ \$ \ \# \ \begin{array}{|c|}\hline x \\ \hline x \\ \hline \end{array}$$
$$\begin{array}{|c|}\hline . \\ \hline D_i \\ \hline \end{array} \ \# \ \begin{array}{|c|}\hline x \\ \hline C_i \\ \hline \end{array} \ \$ \ \# \ \begin{array}{|c|}\hline x \\ \hline A_i \\ \hline \end{array} \}$$

We note that the top and bottom rows of the rectangular arrays generated in the individual language will be of the form $xa^mxb^mxa^mxb^m...x$ as there are n component grammars.

Theorem 2 For $Y \in \{I, T\}$,

1. $2DRML = Y_{isgs}L_1(REG) \subset Y_{isgs}L_2(REG) \subset \cdots \subset Y_{isgs}L_n(REG) \subset \cdots$

2. $2DCFML = Y_{isgs}L_1(CF) \subset Y_{isgs}L_2(CF) \subset \cdots \subset Y_{isgs}L_n(CF) \subset \cdots$

The first statement follows in view of the Example 4. The second can be shown on similar lines.

4 Conclusion

The image splicing grammar system introduced in this paper appears to be a powerful means of generating picture arrays. It remains to compare other picture generating mechanisms with these systems.

References

[1] E. Csuhaj-Varjú, Grammar systems: 12 years, 12 problems (short version), In R. Freund and A. Kelemenová (Eds.), Proceedings of the International Workshop on Grammar Systems 2000, 2000, 77-92, Silesian University, Opava.

[2] E. Csuhaj-Varjú, J. Dassow, J. Kelemen and Gh. Păun, Grammar systems: A grammatical approach to distribution and cooperation, Gordon and Breach Science Publishers, 1994.

[3] J. Dassow and V. Mitrana, Splicing grammar systems, Computers and Artificial Intelligence, 15, 1996, 109-122.

[4] R. Freund, Array Grammar Systems, Journal of Automata, Languages and Combinatorics 5,1, 2000, 13-29.

[5] D. Giammarresi and A. Restivo, Two-dimensional languages, In "Handbook of Formal Languages" Vol.3, Eds. G. Rozenberg and A. Salomaa, Springer Verlag, 1997, 215-267.

[6] T. Head, Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviours, Bull. Math. Biology, 49. 1987, 737-759.

[7] T. Head, Gh. Păun and D. Pixton, Language theory and molecular genetics: Generative mechanisms suggested by DNA recombination, In "Handbook of Formal Languages" Vol.2, Eds. G. Rozenberg and A. Salomaa, Springer Verlag, 1997, 295 - 360.

[8] P. Helen Chandra, K.G. Subramanian, and D.G. Thomas, Parallel Splicing on Images, To appear in Int. J. of pattern recognition and artificial intelligence.

[9] A. Rosenfeld and R. Siromoney, Picture languages - a survey, Languages of design, 1, 1993, 229–245.

[10] G. Siromoney, R. Siromoney and K. Krithivasan, Abstract families of matrices and picture languages, Computer Graphics and Image Processing, 1, 1972, 234-307.