Electronic Notes in Theoretical Computer Science 46 (2001)
URL: http://www.elsevier.nl/locate/entcs/volume46.html 14 pages

# A Note on Parallel Splicing on Images

P. Helen Chandra [1], K. G. Subramanian [2], and D. G. Thomas

*Department of Mathematics, Madras Christian College,*
*Tambaram, Chennai (formerly Madras) 600 059, INDIA*

D. L. Van

[b] *Hanoi Institute of Mathematics,*
*P.O.Box 631 Bo Ho, 10000 Hanoi, Vietnam*

**Abstract**

The concept of splicing on images which is done in parallel is introduced. This is an extension of the operation of splicing on strings extensively studied in the context of DNA computing. Various properties of splicing on images are examined.

## 1 Introduction

$L$-systems which were introduced in the seventies to model biological development initiated the use of parallel rewriting of strings and enriched both formal language theory and life sciences with major developments [4, 7]. Splicing systems are another model recently introduced by Head [2] on biological considerations. These systems are intended to model certain recombinant behavior of $DNA$ molecules and are of current interest and study [3].

On the other hand, in syntactic approaches to generation and recognition of images or pictures considered as digitized arrays, several two-dimensional grammars have been proposed and studied [6]. Extending the $L$-system type rewriting to arrays, a generative model was proposed in [8]. In [1], an elegant generalization of the concept of local and recognizable string languages to two-dimensional picture languages has been done. Recently, Krithivasan etal [5] extended the concept of splicing to arrays and defined array splicing systems.

In this paper, a new method of applying the splicing operation on images of rectangular arrays is introduced. Splicing rules that involve $2 \times 1$ or $1 \times 2$ dominoes are considered. Two arrays are column spliced or row spliced by using the domino splicing rules in parallel. The resulting model called $H$ array

[1] Email: chandrajac@yahoo.com
[2] Email: kgsmcc@kmronline.com

splicing system which is simple to handle is compared with other generative mechanisms of picture languages. Some closure results under geometric operations and language theoretic operations are considered. The study initiated in this paper might prove useful to analyze better the structure of images.

## 2 Basic Definitions

Let $\Sigma$ be a finite alphabet. $\Sigma^*$ is the set of all words over $\Sigma$ including the empty word $\lambda$. An image or a picture over $\Sigma$ is a rectangular array of elements of $\Sigma$. The set of all images is denoted by $\Sigma^{**}$. An image or a picture of size $m \times n$ is an array of the form

$$\begin{array}{ccccc} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ & \cdots & \cdots & \cdots & \\ & \cdots & \cdots & \cdots & \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{array}$$

or in short $[a_{ij}]_{m \times n}$. A picture language or a two-dimensional language over $\Sigma$ is a subset of $\Sigma^{**}$.

$$\text{Let } A = \begin{array}{ccc} a_{11} & \cdots & a_{1p} \\ & \cdots & \\ a_{m1} & \cdots & a_{mp} \end{array}, \qquad B = \begin{array}{ccc} b_{11} & \cdots & b_{1q} \\ & \cdots & \\ b_{n1} & \cdots & b_{nq} \end{array}$$

The column concatenation $A \Phi B$ of $A$ and $B$ is defined only when $m = n$ and is given by

$$A \Phi B = \begin{array}{cccccc} a_{11} & \cdots & a_{1p} & b_{11} & \cdots & b_{1q} \\ & \cdots & & & \cdots & \\ & \cdots & & & \cdots & \\ a_{m1} & \cdots & a_{mp} & b_{n1} & \cdots & b_{nq} \end{array}$$

Similarly, the row concatenation $A \Theta B$ of $A$ and $B$ is defined only when

$p = q$ and is given by

$$A\Theta B = \begin{array}{c} a_{11} \cdots a_{1p} \\ \cdots \\ \cdots \\ a_{m1} \cdots a_{mp} \\ b_{11} \cdots b_{1q} \\ \cdots \\ \cdots \\ b_{n1} \cdots b_{nq} \end{array}$$

If $L_1, L_2$ are two picture languages over an alphabet $\Sigma$, the column concatenation $L_1 \Phi L_2$ of $L_1$ and $L_2$ is defined by

$$L_1 \Phi L_2 = \{A\Phi B | A \in L_1 \text{ and } B \in L_2\}.$$

The row concatenation $L_1 \Theta L_2$ of $L_1$ and $L_2$ is defined by

$$L_1 \Theta L_2 = \{A\Theta B | A \in L_1 \text{ and } B \in L_2\}.$$

We recall the notions of local and recognizable picture languages [1]. Given a picture A of size $(m, n)$, we denote by $B_{h,k}(A)$, for $h \leq m, k \leq n$, the set of all blocks (or sub-pictures) of A of size $(h, k)$. We call a square picture of size $(2, 2)$ as a tile. Let $\Gamma$ be a finite alphabet. A two-dimensional language $L \subseteq \Gamma^{**}$ is local if there exists a finite set $\theta$ of tiles over the alphabet $\Gamma \cup \{\#\}$ such that $L = \{A \in \Gamma^{**} | B_{2,2}(\hat{A}) \subseteq \theta\}$ where $\hat{A}$ is a picture of size $(m+2, n+2)$ obtained by surrounding $A$ with a special boundary symbol $\# \notin \Gamma$.

**Example 2.1** The picture language $M$ consisting of arrays $A$ (Fig. 1a) of all sizes describing token $L$ of $1's$ (interpreting $0's$ as blank) (Fig. 1b) is a local language.

$$\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{array} \qquad\qquad \begin{array}{cccccc} 1 & & & & & \\ 1 & & & & & \\ 1 & & & & & \\ 1 & & & & & \\ 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

Fig. 1a. Array A describing token $L$ — Fig. 1b. Token $L$ of 1's

The corresponding set

$$\theta = \left\{ \begin{array}{cccccccccccc} \# & \# & \# & \# & \# & 1 & 1 & \# & \# & \# & \# & \# \\ \# & 1, & 0 & \#, & \# & \#, & \# & \#, & 1 & 0, & 0 & 0 \\ \\ \# & 1 & 0 & \# & 1 & 1 & 0 & \# & 1 & 0 & 0 & 0 \\ \# & 1, & 0 & \#, & \# & \#, & 1 & \#, & 1 & 0, & 0 & 0 \\ \\ 1 & 0 & 0 & 0 & & & & & & & & \\ 1 & 1, & 1 & 1 & & & & & & & & \end{array} \right\}$$

A tiling system ($TS$) is a 4-tuple $T = (\Sigma.\Gamma, \theta, \pi)$, where $\Sigma$ and $\Gamma$ are two finite alphabets, $\theta$ is finite set of tiles over the alphabet $\Gamma \cup \{\#\}$ and $\pi : \Gamma \to \Sigma$ is a projection. The tiling system $T$ defines a language $L$ over the alphabet $\Sigma$ as follows: $L = \pi(L')$ where $L' = L(\theta)$ is the local language over $\Gamma$ corresponding to the set of tiles $\theta$. We write $L = L(T)$. We say that a language $L \subseteq \Sigma^{**}$ is recognizable by tiling systems (or tiling recognizable) if there exists a tiling system $T = (\Sigma, \Gamma, \theta, \pi)$ such that $L = L(T)$. We denote by $\mathcal{L}(TS)$ the family of all two-dimensional languages recognizable by tiling system. In other words $L \in \mathcal{L}(TS)$ if it is a projection of some local language.

Different systems for generating pictures using grammars have been studied in the literature [1]. We recall here models that consist of two sets of rewriting rules: horizontal and vertical rules, respectively. These models operate by first generating a (horizontal) string $\sigma$ using the horizontal rules; then generating a rectangular picture from the top row $\sigma$ by applying in parallel vertical rules. These grammars actually formalize the parallel generation of two-dimensional languages.

A two-dimensional right-linear grammar ($2RLG$) is defined by a 7-tuple $G = (V_h, V_v, \Sigma_I, \Sigma, S, R_h, R_v)$, where $V_h$ is a finite set of horizontal variables; $V_v$ is a finite set of vertical variables; $\Sigma_I \subseteq V_v$ is a finite set of intermediates; $\Sigma$ is a finite set of terminals; $S \in V_h$ is a starting symbol; $R_h$ is a finite set of horizontal rules of the form $S_1 \to AS_2$ or $S_1 \to A$, where $S_1, S_2 \in V_h$ and $A \in \Sigma_I$; $R_v$ is a finite set of vertical rules of the form $W \to aW'$ or $W \to a$, where $W, W' \in V_v$ and $a \in \Sigma$.

**Example 2.2**

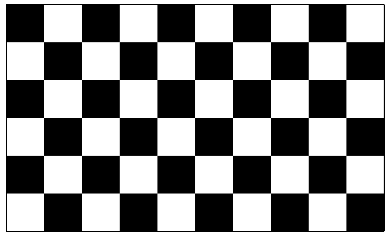
Let $G = (V_h, V_v, \Sigma_I, \Sigma, S, R_h, R_v)$ be a grammar, where :

$V_h = \{S, T\}; V_v = \{A, B, C, D\}; \Sigma_I = \{A, B\}; \Sigma = \{0, 1\};$

$R_h = \{S \to AT; T \to BS; T \to B\};$

$R_v = \{A \to 1C; C \to 0A; C \to 0; B \to 0D; D \to 1B; D \to 1.\}$

In the first phase, $G$ generates the string language $H(G) = \{AB\}^+$. In the second phase, starting from strings of $H(G)$ considered as top rows of pictures, by application of the vertical rules in $R_v$, we obtain the arrays of the picture language $L$ generated by $G$, which is the set of “chessboard” pictures of even side-length; i.e., pictures of the following form:



represented by

$$\begin{array}{cccccccccc}
1&0&1&0&1&0&1&0&1&0\\
0&1&0&1&0&1&0&1&0&1\\
1&0&1&0&1&0&1&0&1&0\\
0&1&0&1&0&1&0&1&0&1\\
1&0&1&0&1&0&1&0&1&0\\
0&1&0&1&0&1&0&1&0&1
\end{array}$$

with 1 standing for black and 0 for white.

We denote by $\mathcal{L}(2RLG)$, the family of picture languages generated by two-dimensional right linear grammars.

The array splicing system introduced in [5], is a generalization of the splicing system on strings originally considered by Head [2]. We refer to [5] for details of the array splicing systems. We informally describe the idea. Four types of splicing are considered in [5]. The idea here is that the arrays X and Y involved in splicing are “split” into sub arrays suitably, ‘crossings’, which are sub arrays of X and Y are required to be the same for splicing to take place. ‘Type-$i$ prefixes’ are exchanged due to splicing.

## 3 H array Splicing Systems

We now introduce the main notion of H array Splicing Systems.

**Definition 3.1** Let $V$ be an alphabet. $\#$, $\$$ are two special symbols, not in V. A domino over V is of the form $\begin{array}{|c|}\hline a\\\hline b\\\hline\end{array}$ or $\begin{array}{|c|c|}\hline a&b\\\hline\end{array}$, $a, b \in V$.

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

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

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

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

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

We write $(X, Y) \mid\!\stackrel{\Phi}{-}\, Z$ if there exist column splicing rules $r_1, r_2, r_3, \cdots, r_{m-1}$ not all different such that

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

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

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

In particular if any of the symbols $a_{ij}$ is $\lambda$ then for all $i, (1 \leq i \leq m), a_{ij} = \lambda$. Likewise for $a_{i,j+1}, b_{ik}, b_{i,k+1} (1 \leq i \leq m)$. 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 $W$.

$$U = \begin{array}{cccc} a_{11} & a_{12} & & a_{1n} \\ \cdots & \cdots & & \cdots \\ a_{i,1} & a_{i,2} & \cdots & a_{i,n} \\ a_{i+1,1} & a_{i+1,2} & \cdots & a_{i+1,n} \\ \cdots & \cdots & & \cdots \\ a_{p1} & a_{p2} & & a_{pn} \end{array} \qquad V = \begin{array}{cccc} b_{11} & b_{12} & & b_{1n} \\ \cdots & \cdots & & \cdots \\ b_{k,1} & b_{k,2} & \cdots & b_{k,n} \\ b_{k+1,1} & b_{k+1,2} & \cdots & b_{k+1,n} \\ \cdots & \cdots & & \cdots \\ b_{q1} & b_{q2} & & b_{qn} \end{array}$$

We write $(U, V) \mid^{\Theta} W$ if there exist row splicing rules $r_1, r_2, r_3, \cdots r_{n-1}$ not all different such that

$$ri = \boxed{a_{i,j}}\boxed{a_{i,j+1}} \quad \# \quad \boxed{a_{i+1,j}}\boxed{a_{i+1,j+1}} \quad \$ \quad \boxed{b_{k,j}}\boxed{b_{k,j+1}} \quad \# \quad \boxed{b_{k+1,j}}\boxed{b_{k+1,j+1}}$$

for all $j, (1 \leq j \leq n-1)$ and for some $i, k(1 \leq i \leq p)$ and $(1 \leq k \leq q)$ and

$$W = \begin{array}{cccc} a_{11} & a_{12} & & a_{1n} \\ \cdots & \cdots & & \cdots \\ a_{i,1} & a_{i,2} & \cdots & a_{i,n} \\ b_{k+1,1} & b_{k+1,2} & \cdots & b_{k+1,n} \\ \cdots & \cdots & & \cdots \\ b_{q1} & b_{q2} & & b_{qn} \end{array}$$

We now say that $W$ is obtained from $U$ and $V$ by domino row splicing in parallel.

**Definition 3.2** We define an $H$ array scheme and an $H$ array splicing system. An $H$ array scheme is a triplet $\Gamma = (V, R_c, R_r)$ where $V$ is an alphabet, $R_c = a$ finite set of domino column splicing rules, and $R_r = a$ finite set of domino row splicing rules.

For a given $H$ array scheme $\Gamma = (V, R_c, R_r)$ and a language $L \subseteq V^{**}$, we define

$$\Gamma(L) = \begin{cases} Z \in V^{**} | (X, Y) |^{\Phi} Z \text{ or } (X, Y) |^{\Theta} Z \text{ for some } X, Y \in L, \\ \qquad\qquad p_i \in R_c \text{ and } q_j \in R_r (1 \leq i \leq m-1), (1 \leq j \leq n-1) \end{cases}$$

In other words, $\Gamma(L)$ consists of arrays obtained by column or row splicing any two arrays of L using the array column or row splicing rules.
Iteratively we define

$$\Gamma^0(L) = L$$
$$\Gamma^{i+1}(L) = \Gamma^i(L) \bigcup \Gamma(\Gamma^i(L)), i \geq 0$$
$$\Gamma^*(L) = \bigcup_{i \geq 0} \Gamma^i(L).$$

An $H$ array splicing system is defined by $S = (\Gamma, I)$ where $\Gamma = (V, R_c, R_r)$ and $I$ is a finite subset of $V^{**}$. The language of $S$ is defined by $L(S) = \Gamma^*(I)$ and we call it a splicing array language and denote the class of such languages by $FHA$.

We illustrate with an example.

**Example 3.3** Let $V = \{a, b\}$

$$I = \begin{matrix} a & b \\ b & a \end{matrix}$$

$$R_c = \left\{ p_1 : \boxed{\begin{matrix} a \\ b \end{matrix}} \ \# \ \boxed{\begin{matrix} \lambda \\ \lambda \end{matrix}} \ \$ \ \boxed{\begin{matrix} \lambda \\ \lambda \end{matrix}} \ \# \ \boxed{\begin{matrix} b \\ a \end{matrix}} \right.$$
$$\left. p_2 : \boxed{\begin{matrix} b \\ a \end{matrix}} \ \# \ \boxed{\begin{matrix} \lambda \\ \lambda \end{matrix}} \ \$ \ \boxed{\begin{matrix} \lambda \\ \lambda \end{matrix}} \ \# \ \boxed{\begin{matrix} a \\ b \end{matrix}} \right\}$$

$$R_r = \left\{ q_1 : \boxed{a | b} \ \# \ \boxed{\lambda | \lambda} \ \$ \ \boxed{\lambda | \lambda} \ \# \ \boxed{b | a} \right.$$
$$\left. q_2 : \boxed{b | a} \ \# \ \boxed{\lambda | \lambda} \ \$ \ \boxed{\lambda | \lambda} \ \# \ \boxed{a | b} \right\}$$

On column splicing in parallel using $p_2$, the arrays $\begin{matrix} a & b \\ b & a \end{matrix}, \begin{matrix} a & b \\ b & a \end{matrix}$ yield

$$\left[ \begin{array}{cc|c} a & b & \lambda \\ b & a & \lambda \end{array} , \begin{array}{c|cc} \lambda & a & b \\ \lambda & b & a \end{array} \right] \mid\!\underline{\Phi} \quad \left[ \begin{matrix} a & b & a & b \\ b & a & b & a \end{matrix} \right]$$

We have shown the empty column $\begin{matrix} \lambda \\ \lambda \end{matrix}$ to indicate the place where splicing is done. Likewise, row splicing in parallel using $q_1, q_2$, gives

$$\left[ \begin{array}{cccc} a & b & a & b \\ b & a & b & a \\ \hline \lambda & \lambda & \lambda & \lambda \end{array} , \begin{array}{cccc} \lambda & \lambda & \lambda & \lambda \\ \hline a & b & a & b \\ b & a & b & a \end{array} \right] \mid\!\underline{\Theta} \quad \left[ \begin{matrix} a & b & a & b \\ b & a & b & a \\ a & b & a & b \\ b & a & b & a \end{matrix} \right]$$

$L$ is the language consisting of all "chessboards" with even side-length [1].

$$\begin{matrix} a & b & a & b & a & b \\ b & a & b & a & b & a \\ a & b & a & b & a & b \\ b & a & b & a & b & a \end{matrix}$$

**Theorem 3.4** *The classes LOC of local array languages and FHA of splicing array languages are incomparable but not disjoint.*

**Proof.** The picture language $M$ consisting of all $m \times n$ arrays $(m \geq 2, n \geq 2)$ describing token $L$ of 1's is in LOC. A member of M is shown in Fig. 1a. Now we give an H array splicing system $S = (V, R_c, R_r, I)$ to describe $M$.

Let $V = \{0, 1\}$

$$R_c = \left\{ p_1 : \boxed{\begin{array}{c} 0 \\ 1 \end{array}} \;\#\; \boxed{\begin{array}{c} \lambda \\ \lambda \end{array}} \;\$\; \boxed{\begin{array}{c} 1 \\ 1 \end{array}} \;\#\; \boxed{\begin{array}{c} 0 \\ 1 \end{array}} \right.$$

$$\left. p_2 : \boxed{\begin{array}{c} 0 \\ 0 \end{array}} \;\#\; \boxed{\begin{array}{c} \lambda \\ \lambda \end{array}} \;\$\; \boxed{\begin{array}{c} 1 \\ 1 \end{array}} \;\#\; \boxed{\begin{array}{c} 0 \\ 0 \end{array}} \right\}$$

$$R_r = \left\{ q_1 : \boxed{1|0} \;\#\; \boxed{1|1} \;\$\; \boxed{\lambda|\lambda} \;\#\; \boxed{1|0} \right.$$

$$\left. q_2 : \boxed{0|0} \;\#\; \boxed{1|1} \;\$\; \boxed{\lambda|\lambda} \;\#\; \boxed{0|0} \right\}$$

and

$$I = \left\{ \begin{array}{c} 1\ 0 \\ 1\ 1 \end{array} \right\}$$

The picture language L of all images (Fig. 2) over $V = \{a\}$ with 3 columns is known to be not in LOC. But it is obtained by an H array splicing system where

$I \ = \ a \quad a \quad a$

$$R_r = \boxed{a|a} \;\#\; \boxed{\lambda|\lambda} \;\$\; \boxed{\lambda|\lambda} \;\#\; \boxed{a|a}$$

and $R_c = \varphi$.

The picture language of square images in which diagonal positions carry symbol 1 but the remaining positions carry symbol 0 (Fig. 3) is in LOC [1]. But it is not in FHA, since it is clear from the definition of row and column splicing that pictures with only square size cannot be generated.

$$\begin{array}{ccc} a\ a\ a & \qquad & 1\ 0\ 0\ 0 \\ a\ a\ a & & 0\ 1\ 0\ 0 \\ a\ a\ a & & 0\ 0\ 1\ 0 \\ a\ a\ a & & 0\ 0\ 0\ 1 \end{array}$$

Fig. 2 Fig. 3

□

**Remark 3.5** The class FHA intersects $\mathcal{L}(TS)$, since $LOC \subseteq \mathcal{L}(TS)$.

**Theorem 3.6** *The class FHA intersects* $\mathcal{L}(2RLG)$

**Proof.** The result follows on noting that the picture language of "chess-boards" with even side-length is generated by a FHA, and is generated by a $2RLG$[1]. □

**Theorem 3.7** *The class FHA intersects with the class of null-context splicing array languages of [5].*

**Proof.** Let Grid $< X, Y, m, n >$ represent an image G of size $< m, n >$ where $m, n$ are odd positive integers $m, n \geq 3$, and $G$ is given by

$$G[i,j] = \begin{cases} X \text{ if i is odd or j is odd} \\ Y \text{ otherwise} \end{cases}$$

where $1 \leq i \leq m, 1 \leq j \leq n$. $G$ is said to be a Grid defined over $< X, Y >$ of size $< m, n >$ . GRIDS $< X, Y >$ represent the set of all Grids over $< X, Y >$ . A member of GRIDS$< X, . >$ is shown in Fig. 4.

$$\begin{array}{ccccccc} X & X & X & X & X & X & X \\ X & . & X & . & X & . & X \\ X & X & X & X & X & X & X \\ X & . & X & . & X & . & X \\ X & X & X & X & X & X & X \\ X & . & X & . & X & . & X \\ X & X & X & X & X & X & X \end{array}$$

Fig 4. Grid $< X, ., 7, 7 >$

It is known that GRIDS $< X, ., m, n >$ is a null-context splicing array language[5].We give an H array splicing system $S = (V, R_c, R_r, I)$, generating it.

Let $V = \{X, .\}$

$$I = Grid < X, ., 3, 3 >= \left\{ \begin{array}{ccc} X & X & X \\ X & . & X \\ X & X & X \end{array} \right\}$$

$$Rc = \left\{ p_1 : \begin{array}{|c|} \hline X \\ \hline X \\ \hline \end{array} \;\#\; \begin{array}{|c|} \hline \lambda \\ \hline \lambda \\ \hline \end{array} \;\$\; \begin{array}{|c|} \hline X \\ \hline X \\ \hline \end{array} \;\#\; \begin{array}{|c|} \hline X \\ \hline . \\ \hline \end{array} \right.$$

$$\left. p_2 : \begin{array}{|c|} \hline X \\ \hline X \\ \hline \end{array} \;\#\; \begin{array}{|c|} \hline \lambda \\ \hline \lambda \\ \hline \end{array} \;\$\; \begin{array}{|c|} \hline X \\ \hline X \\ \hline \end{array} \;\#\; \begin{array}{|c|} \hline . \\ \hline X \\ \hline \end{array} \right\}$$

$$R_r = \left\{ q_1 : \begin{array}{|c|c|} \hline X & X \\ \hline \end{array} \;\#\; \begin{array}{|c|c|} \hline \lambda & \lambda \\ \hline \end{array} \;\$\; \begin{array}{|c|c|} \hline X & X \\ \hline \end{array} \;\#\; \begin{array}{|c|c|} \hline X & . \\ \hline \end{array} \right.$$

$$\left. q_2 : \begin{array}{|c|c|} \hline X & X \\ \hline \end{array} \;\#\; \begin{array}{|c|c|} \hline \lambda & \lambda \\ \hline \end{array} \;\$\; \begin{array}{|c|c|} \hline X & X \\ \hline \end{array} \;\#\; \begin{array}{|c|c|} \hline . & X \\ \hline \end{array} \right\}$$

□

**Remark 3.8** We now consider row-column combination of two string languages[1]. Let V be a finite alphabet and let $S_1$ and $S_2 \subseteq V^*$ be two string languages over V.The row-column combination of $S_1$ and $S_2$ is a picture language $L = S_1 \oplus S_2 \subseteq V^{**}$ such that a picture $p \in V^{**}$ belongs to L if and only if the strings corresponding to the rows and columns of $p$ belong to $S_1$ and $S_2$ respectively.

We give an example of a picture language L in FHA which is a row-column combination picture language. Here L consists of all pictures over $V = \{0, 1\}$ whose first and last column consist only of 1's. In fact $L = (\{1\}S_1\{1\}) \oplus V^*$ where $S_1 \subseteq V^*$.

Let $V = \{0, 1\}$ $I = \left\{ \begin{matrix} 1 \ x_1 \ 1 \\ 1 \ x_2 \ 1 \end{matrix} \right\}$ where $x_i = 0$ or $1 (i = 1, 2)$.

$$R_c = p_1 : \begin{array}{|c|}\hline x_1 \\ \hline x_2 \\ \hline\end{array} \ \# \ \begin{array}{|c|}\hline 1 \\ \hline 1 \\ \hline\end{array} \ \$ \ \begin{array}{|c|}\hline 1 \\ \hline 1 \\ \hline\end{array} \ \# \ \begin{array}{|c|}\hline x_1 \\ \hline x_2 \\ \hline\end{array} \text{ and}$$

$$R_r = \left\{ \begin{array}{l} q_1 : \begin{array}{|c|c|}\hline 1 & x_1 \\ \hline\end{array} \ \# \ \begin{array}{|c|c|}\hline \lambda & \lambda \\ \hline\end{array} \ \$ \ \begin{array}{|c|c|}\hline \lambda & \lambda \\ \hline\end{array} \ \# \ \begin{array}{|c|c|}\hline 1 & x_1 \\ \hline\end{array} \\ q_2 : \begin{array}{|c|c|}\hline x_1 & 1 \\ \hline\end{array} \ \# \ \begin{array}{|c|c|}\hline \lambda & \lambda \\ \hline\end{array} \ \$ \ \begin{array}{|c|c|}\hline \lambda & \lambda \\ \hline\end{array} \ \# \ \begin{array}{|c|c|}\hline x_1 & 1 \\ \hline\end{array} \\ q_3 : \begin{array}{|c|c|}\hline x_1 & x_2 \\ \hline\end{array} \ \# \ \begin{array}{|c|c|}\hline \lambda & \lambda \\ \hline\end{array} \ \$ \ \begin{array}{|c|c|}\hline \lambda & \lambda \\ \hline\end{array} \ \# \ \begin{array}{|c|c|}\hline x_1 & x_2 \\ \hline\end{array} \end{array} \right\}$$

Now we examine certain closure results:

**Theorem 3.9** *The class FHA is closed under reflections on the base and right leg and rotations by* $90°, 180°$ *and* $270°$.

**Proof.** We first prove that FHA is closed under reflections.

Let $S = (\Gamma, I)$ where $\Gamma = (V, R_c, R_r)$ and $I$ is a finite subset of $V^{**}$ be a splicing system, with rules in $R_c$ of the form

$$p = \begin{array}{|c|}\hline a_1 \\ \hline b_1 \\ \hline\end{array} \ \# \ \begin{array}{|c|}\hline c_1 \\ \hline d_1 \\ \hline\end{array} \ \$ \ \begin{array}{|c|}\hline a_2 \\ \hline b_2 \\ \hline\end{array} \ \# \ \begin{array}{|c|}\hline c_2 \\ \hline d_2 \\ \hline\end{array}$$

and in $R_r$ of the form

$$q = \begin{array}{|c|c|}\hline a_1 & b_1 \\ \hline\end{array} \ \# \ \begin{array}{|c|c|}\hline c_1 & d_1 \\ \hline\end{array} \ \$ \ \begin{array}{|c|c|}\hline a_2 & b_2 \\ \hline\end{array} \ \# \ \begin{array}{|c|c|}\hline c_2 & d_2 \\ \hline\end{array}$$

describing a picture language L.

The picture language consisting of images which are reflections of arrays of L on the base can be obtained by an H array splicing system consisting of rules of the form

$$\begin{array}{|c|}\hline b_1 \\ \hline a_1 \\ \hline\end{array} \ \# \ \begin{array}{|c|}\hline d_1 \\ \hline c_1 \\ \hline\end{array} \ \$ \ \begin{array}{|c|}\hline b_2 \\ \hline a_2 \\ \hline\end{array} \ \# \ \begin{array}{|c|}\hline d_2 \\ \hline c_2 \\ \hline\end{array}$$

corresponding to p and rules of the form

$$\begin{array}{|c|c|}\hline c_2 & d_2 \\ \hline\end{array} \;\#\; \begin{array}{|c|c|}\hline a_2 & b_2 \\ \hline\end{array} \;\$\; \begin{array}{|c|c|}\hline c_1 & d_1 \\ \hline\end{array} \;\#\; \begin{array}{|c|c|}\hline a_1 & b_1 \\ \hline\end{array}$$

corresponding to q.

Similarly the reflections of arrays of $L$ on the right leg can be obtained by an $H$ array splicing system with modified rules

$$\begin{array}{|c|}\hline c_2 \\ \hline d_2 \\ \hline\end{array} \;\#\; \begin{array}{|c|}\hline a_2 \\ \hline b_2 \\ \hline\end{array} \;\$\; \begin{array}{|c|}\hline c_1 \\ \hline d_1 \\ \hline\end{array} \;\#\; \begin{array}{|c|}\hline a_1 \\ \hline b_1 \\ \hline\end{array}$$

and

$$\begin{array}{|c|c|}\hline b_1 & a_1 \\ \hline\end{array} \;\#\; \begin{array}{|c|c|}\hline d_1 & c_1 \\ \hline\end{array} \;\$\; \begin{array}{|c|c|}\hline b_2 & a_2 \\ \hline\end{array} \;\#\; \begin{array}{|c|c|}\hline d_2 & c_2 \\ \hline\end{array}$$

respectively corresponding to $p$ and $q$.

We next prove that $FHA$ is closed under rotations by $90°, 180°$ and $270°$. We mention only the modified rules of $R_c$ and $R_r$

$$\begin{array}{|c|}\hline c_2 \\ \hline d_2 \\ \hline\end{array} \;\#\; \begin{array}{|c|}\hline a_2 \\ \hline b_2 \\ \hline\end{array} \;\$\; \begin{array}{|c|}\hline c_1 \\ \hline d_1 \\ \hline\end{array} \;\#\; \begin{array}{|c|}\hline a_1 \\ \hline b_1 \\ \hline\end{array}$$

and

$$\begin{array}{|c|c|}\hline b_1 & a_1 \\ \hline\end{array} \;\#\; \begin{array}{|c|c|}\hline d_1 & c_1 \\ \hline\end{array} \;\$\; \begin{array}{|c|c|}\hline b_2 & a_2 \\ \hline\end{array} \;\#\; \begin{array}{|c|c|}\hline d_2 & c_2 \\ \hline\end{array}$$

for rotation by $90°$;

$$\begin{array}{|c|}\hline d_2 \\ \hline c_2 \\ \hline\end{array} \;\#\; \begin{array}{|c|}\hline b_2 \\ \hline a_2 \\ \hline\end{array} \;\$\; \begin{array}{|c|}\hline d_1 \\ \hline c_1 \\ \hline\end{array} \;\#\; \begin{array}{|c|}\hline b_1 \\ \hline a_1 \\ \hline\end{array}$$

and

$$\begin{array}{|c|c|}\hline d_2 & c_2 \\ \hline\end{array} \;\#\; \begin{array}{|c|c|}\hline b_2 & a_2 \\ \hline\end{array} \;\$\; \begin{array}{|c|c|}\hline d_1 & c_1 \\ \hline\end{array} \;\#\; \begin{array}{|c|c|}\hline b_1 & a_1 \\ \hline\end{array}$$

for rotation by $180°$;

$$\begin{array}{|c|}\hline b_1 \\ \hline a_1 \\ \hline\end{array} \;\#\; \begin{array}{|c|}\hline d_1 \\ \hline c_1 \\ \hline\end{array} \;\$\; \begin{array}{|c|}\hline b_2 \\ \hline a_2 \\ \hline\end{array} \;\#\; \begin{array}{|c|}\hline d_2 \\ \hline c_2 \\ \hline\end{array}$$

and

$$\begin{array}{|c|c|}\hline c_2 & d_2 \\ \hline\end{array} \;\#\; \begin{array}{|c|c|}\hline a_2 & b_2 \\ \hline\end{array} \;\$\; \begin{array}{|c|c|}\hline c_1 & d_1 \\ \hline\end{array} \;\#\; \begin{array}{|c|c|}\hline a_1 & b_1 \\ \hline\end{array}$$

for rotation by $270°$. □

**Theorem 3.10** *The class $FHA$ is not closed under union and concatenation.*

**Proof.** Let $L_1$ be a language consisting of arrays with 3 rows and any number of columns with left border made of a's, right border of b's and inner part of x's. A member of $L_1$ is shown in fig. 5.

$$\begin{array}{ccccccc} a & x & x & x & x & x & b \\ a & x & x & x & x & x & b \\ a & x & x & x & x & x & b \end{array}$$

Fig. 5.

Similarly, let $L_2$ be another language of arrays as in $L_1$ but left border made of c's, right border of d's. It is clear that a member of $L_1 \cup L_2$ will have left and right borders only of a's and b's respectively or only of c's and d's respectively. Any column splicing rule required to generate $L_1 \cup L_2$ will have to increase the inner columns of x's. But on column splicing, two initial arrays of the form

$$\begin{array}{ccc} a & x & b \\ a & x & b \\ a & x & b \end{array} \quad , \quad \begin{array}{ccc} c & x & d \\ c & x & d \\ c & x & d \end{array}$$

of an $H$ array splicing system would yield arrays with left and right border of $a's$ and $d's$ or $c's$ and $b's$. These are not elements of $L_1 \cup L_2$.

Likewise, column splicing of two initial arrays of the form

$$\begin{array}{cccccc} a & x & b & c & x & d \\ a & x & b & c & x & d \\ a & x & b & c & x & d \end{array}$$

of an $H$ array splicing system that might generate $L_1 \Phi L_2$ would yield arrays that are not in $L_1 \Phi L_2$. An analogous argument applies to row concatenation.$\square$

## Conclusion

In this paper, an attempt has been made to extend in a simple but effective manner, the splicing operation to images of rectangular arrays. Although this new system intersects with the array splicing system of [5], it remains open to find out where exactly this class stands.

## Acknowledgement

The authors K.G. Subramanian and D.G. Thomas would like to thank Hanoi Institure of Mathematics, Vietnam for extending support for their visit to the institute during Oct. - Nov., 1999. The authors are grateful to Prof. D.L.Van for his collaboration.

## References


[1] Giammarresi, D. and A. Restivo, *Two-dimensional Languages*, In Handbook of Formal Languages, eds. G.Rozenberg and A.Salomaa, Springer Verlag **3** (1997), 215-265.

[2] Head, T., *Formal Language Theory and DNA: An Analysis of the Generative Capacity of Specific Recombinant Behaviours*, Bull. Math. Biology, **49** (1987), 735-759.

[3] Head, T., Gh. Păun and D. Pixton, *Language Theory and Molecular Genetics: Generative Mechanisms suggested by DNA Recombination*, In Handbook of Formal Languages, eds. G. Rozenberg and A. Salomaa, Springer Verlag **2** (1997), 295-358.

[4] Herman, G. T. and G. Rozenberg, "Developmental Systems and Languages", North-Holland, Amsterdam (1975).

[5] Krithivasan, K., V. T. Chakaravarthy and R. Rama, *Array Splicing Systems*, In Computing with Bio-molecules: Theory and Experiments, ed. Gh. Păun, Springer Verlag (1998).

[6] Rosenfeld, A. and R. Siromoney, *Picture Languages - A Survey*, Languages Des. **1** (1993).

[7] Rozenberg, G. and A. Salomaa, *The Mathematical Theory of $L-$Systems*, Academic Press, New York (1980).

[8] Siromoney, R. and G. Siromoney, *Extended Controlled Table $L-$Arrays*, Information and Control **35** (1977), 119 -138.