A Picture Array Generating Model Based on Flat Splicing Operation

K.G. Subramanian[1], Atulya K. Nagar[1], and Linqiang Pan[2⋆]

[1] Department of Mathematics and Computer Science,
Faculty of Science, Liverpool Hope University,
Liverpool, L16 9JD, UK
kgsmani1948@gmail.com, nagara@hope.ac.uk

[2] Key Laboratory of Image Information Processing and Intelligent Control,
School of Automation,
Huazhong University of Science and Technology,
Wuhan 430074, Hubei, China
lqpan@mail.hust.edu.cn

Abstract. The bio-inspired operations of linear and circular splicing respectively on linear and circular strings of symbols have been extensively investigated by many researchers for their theoretical properties. Recently, another kind of splicing of two words, referred to as flat splicing on strings, has been considered. We here extend this operation to flat splicing on picture arrays, thus defining a new model of picture generation, which we call as array flat splicing system (AFS) and obtain some results on the generative power of AFS in comparison with certain well-known picture array defining models.

Keywords: Splicing on words, Flat splicing, Picture array, Picture language

1 Introduction

In modelling the recombinant behaviour of DNA molecules, Head defined an operation on strings of symbols, called splicing [4]. Subsequently, several theoretical studies on the power of this operation in terms of language theoretic results have been established [5, 6]. Recently, a specific kind of splicing on circular words has been suitably adapted to linear words, resulting in a splicing operation, referred to as flat splicing [1]. While the usual splicing on two words involves the idea of "cutting" and "pasting" according to a splicing rule [5], the flat splicing on a pair of words (u, v) involves "cutting" u and "inserting" v into it, as dictated by a flat splicing rule.

Motivated by problems in image analysis and picture processing, several two-dimensional picture array generating models have been proposed and investigated, e.g., [3, 10]. One such study is done in [2], by extending the operation of

⋆ Corresponding author.

splicing on words [5] to arrays and the generative power of the resulting splicing system, called H array splicing system, is examined in [2].

We here extend the operation of flat splicing on linear words considered in [1] to picture arrays and define a system called array flat splicing system (AFS). The extension we have defined is more close to the alphabetic case of flat splicing on words considered in [1]. We then make a theoretical investigation of (AFS) by comparing the family of picture languages generated by these systems with the families of picture languages of certain well-known two-dimensional picture generating models.

2 Preliminaries

We refer to [3, 7] for concepts and results related to formal languages, array grammars and two-dimensional languages.

Given a finite alphabet Σ, a linear word or simply, a word (also called a string) α is a finite sequence $a_1a_2\cdots a_n$ of letters a_i, $1 \leq i \leq n$, in Σ. The set of all words over Σ, including the empty word λ with no symbols, is denoted by Σ^*. The length of a word α is the number of letters in the word, denoted by $|\alpha|$. For any word $\alpha = a_1a_2\dots a_n$ ($n \geq 1$), we denote by ${}^t\alpha$ the word α written vertically. For example, if $\alpha = bab$ over $\{a, b\}$, then

$$ {}^t\alpha = \begin{matrix} b \\ a \\ b \end{matrix}. $$

An $p \times q$ picture array (also called an array or a picture) X over an alphabet Σ is a rectangular array with p rows and q columns and is of the form

$$ X = \begin{matrix} \mathtt{a}_{11} & \cdots & \mathtt{a}_{1q} \\ \vdots & \ddots & \vdots \\ \mathtt{a}_{p1} & \cdots & \mathtt{a}_{pq} \end{matrix} $$

where each symbol $a_{ij} \in \Sigma, 1 \leq i \leq p$, $1 \leq j \leq q$. For the sake of convenience, we may write $X = [a_{ij}]_{p,q}$. The topmost row of X is considered as the first row and the bottommost row of X, the last row while the leftmost column is considered as the first column of X and the rightmost column, the last column of X. We denote the number of rows and the number of columns of X, respectively, by $|X|_r$ and $|X|_c$. The set of all rectangular arrays over Σ is denoted by Σ^{**}, which contains the empty array λ with no symbols. $\Sigma^{++} = V^{**} - \{\lambda\}$. A picture language is a subset of V^{**}.

Let $X = [a_{ij}]_{p,q}$ and $Y = [b_{ij}]_{r,s}$ be two non-empty arrays over an alphabet Σ. The operation of column concatenation of arrays X and Y, denoted by $X \circ Y$, is defined only when $p = r$ and is given by

$$ \mathtt{X} \circ \mathtt{Y} = \begin{matrix} \mathtt{a}_{11} & \cdots & \mathtt{a}_{1q} & \mathtt{b}_{11} & \cdots & \mathtt{b}_{1s} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathtt{a}_{p1} & \cdots & \mathtt{a}_{pq} & \mathtt{b}_{p1} & \cdots & \mathtt{b}_{ps} \end{matrix} $$

Likewise, the operation of row concatenation of X and Y, denoted by $X \diamond Y$, is defined only when $q = s$ and is given by

$$\mathtt{X} \diamond \mathtt{Y} = \begin{array}{c} \mathtt{a}_{11} \cdots \mathtt{a}_{1q} \\ \cdots \cdots \cdots \\ \mathtt{a}_{p1} \cdots \mathtt{a}_{pq} \\ \mathtt{b}_{11} \cdots \mathtt{b}_{1q} \\ \cdots \cdots \cdots \\ \mathtt{b}_{r1} \cdots \mathtt{b}_{rq} \end{array}$$

Furthermore, $X \circ \lambda = \lambda \circ X = X \diamond \lambda = \lambda \diamond W = W$, for every array W.

We now recall an operation, called flat splicing on linear words, considered by Berstel et al. [1]. Given an alphabet Σ, a flat splicing rule r is of the form $(\alpha|\gamma - \delta|\beta)$, where $\alpha, \beta, \gamma, \delta$ are words over the alphabet Σ. Given two words $u = x\alpha\beta y$, $v = \gamma z\delta$, an application of the flat splicing rule $r = (\alpha|\gamma - \delta|\beta)$ to the pair (u, v) yields the word $w = x\alpha\gamma z\delta\beta y$. In other words, the second word v is inserted between α and β in the first word u as a result of applying the rule r.

3 Array Flat Splicing Systems

We extend the notion of flat splicing on words [1] to arrays. In fact we introduce two kinds of flat splicing rules, namely, column flat splicing rule and row flat splicing rule. We then define their application on a pair of arrays and thus introduce a new model of picture generation, namely, array flat splicing system.

Definition 1. *Let V be an alphabet.*

(i) *A column flat splicing rule is of the form* $({}^t(a_1a_2)|{}^t(x_1x_2) - {}^t(y_1y_2)|{}^t(b_1b_2))$ *where* $a_1, a_2, b_1, b_2 \in \Sigma \cup \{\lambda\}$ *with* $|a_1| = |a_2|$ *and* $|b_1| = |b_2|$, $x_1, x_2, y_1, y_2 \in \Sigma \cup \{\lambda\}$ *with* $|x_1| = |x_2|$ *and* $|y_1| = |y_2|$.

(ii) *A row flat splicing rule is of the form* $(c_1c_2|u_1u_2 - v_1v_2|d_1d_2)$ *where* $c_1, c_2, d_1, d_2 \in \Sigma \cup \{\lambda\}$ *with* $|c_1| = |c_2|$ *and* $|d_1| = |d_2|$, $u_1, u_2, v_1, v_2 \in \Sigma \cup \{\lambda\}$ *with* $|u_1| = |u_2|$ *and* $|v_1| = |v_2|$.

(iii) *Let* $r_1, r_2, \cdots, r_{m-1}$ *be a sequence of* $(m-1)$ *column flat splicing rules given by*

$$r_i = ({}^t(\alpha_i\alpha_{i+1})|{}^t(\gamma_i\gamma_{i+1}) - {}^t(\delta_i\delta_{i+1})|{}^t(\beta_i\beta_{i+1})),$$

for $1 \leq i \leq (m-1)$. *Let* X, Y *be two picture arrays, each with* m *rows, for some* $m \geq 1$, *and given by*

$$X = X_1 \circ {}^t(\alpha_1\alpha_2 \cdots \alpha_m) \circ {}^t(\beta_1\beta_2 \cdots \beta_m) \circ X_2,$$

$$Y = {}^t(\gamma_1\gamma_2 \cdots \gamma_m) \circ Y' \circ {}^t(\delta_1\delta_2 \cdots \delta_m),$$

where X_1, X_2, Y' *are arrays over* Σ *with* m *rows,* α_i, β_i, $(1 \leq i \leq m)$, $\in \Sigma \cup \{\lambda\}$ *with* $|\alpha_1| = |\alpha_2| = \cdots = |\alpha_m|$, $|\beta_1| = |\beta_2| = \cdots = |\beta_m|$, γ_i, δ_i, $(1 \leq$

$i \le m$), $\in$ $\Sigma \cup \{\lambda\}$ with $|\gamma_1| = |\gamma_2| = \cdots = |\gamma_m|$, $|\delta_1| = |\delta_2| = \cdots = |\delta_m|$. An application of the column flat splicing rules $r_1, r_2, \cdots, r_{m-1}$ to the pair of arrays (X, Y) yields the array Z

$$= X_1 \circ {}^t(\alpha_1\alpha_2 \cdots \alpha_m) \circ {}^t(\gamma_1\gamma_2 \cdots \gamma_m) \circ Y' \circ {}^t(\delta_1\delta_2 \cdots \delta_m) \circ {}^t(\beta_1\beta_2 \cdots \beta_m) \circ X_2.$$

The pair (X, Y) yielding Z is then denoted by $(X, Y) \vdash_c Z$.

(iv) Let $s_1, s_2, \cdots, s_{n-1}$ be a sequence of $(n-1)$ row flat splicing rules given by

$$s_j = (\eta_j\eta_{j+1}|(\mu_j\mu_{j+1}) - (\nu_j\nu_{j+1})|\theta_j\theta_{j+1}),$$

for $1 \le j \le (n-1)$. Let U, V be two picture arrays, each with n columns, for some $n \ge 1$, and given by

$$U = U_1 \diamond (\eta_1\eta_2 \cdots \eta_n) \diamond (\theta_1\theta_2 \cdots \theta_n) \diamond U_2,$$

$$V = (\mu_1\mu_2 \cdots \mu_n) \diamond V' \diamond (\delta_1\delta_2 \cdots \delta_n)$$

where U_1, U_2, V' are arrays over Σ with n columns, η_j, θ_j, $(1 \le j \le n)$, $\in$ $\Sigma \cup \{\lambda\}$ with $|\eta_1| = |\eta_2| = \cdots = |\eta_n|$, $|\theta_1| = |\theta_2| = \cdots = |\theta_n|$, μ_j, ν_j, $(1 \le j \le n)$, $\in$ $\Sigma \cup \{\lambda\}$ with $|\mu_1| = |\mu_2| = \cdots = |\mu_n|$, $|\nu_1| = |\nu_2| = \cdots = |\nu_n|$. An application of the row flat splicing rules $s_1, s_2, \cdots, s_{n-1}$ to the pair of arrays (U, V) yields the array W

$$= U_1 \diamond (\eta_1\eta_2 \cdots \eta_n) \diamond (\mu_1\mu_2 \cdots \mu_n) \diamond V' \diamond (\delta_1\delta_2 \cdots \delta_n) \diamond (\theta_1\theta_2 \cdots \theta_n) \diamond U_2.$$

The pair (U, V) yielding W is then denoted by $(U, V) \vdash_r W$.

(v) An array flat splicing rule is either a column flat splicing rule or a row flat splicing rule. The notation $\vdash$ denotes either $\vdash_c$ or $\vdash_r$.

(vi) For a picture language $L \subseteq \Sigma^{**}$ and a set R of array flat splicing rules, we define

$$f(L) = \{M \in \Sigma^{**} \mid (X, Y) \vdash M, for X, Y \in L, and\ some\ rule\ in\ R\}.$$

Definition 2. *An array flat splicing system (AFS) is $\mathcal{A} = (\Sigma, M, R_c, R_r)$ where Σ is an alphabet, M is a finite set of arrays over Σ, called initial set, R_c is a finite set of column flat splicing rules and R_r is a finite set of row flat splicing rules.*

The picture language $L(\mathcal{A})$ generated by $\mathcal{A}$ is iteratively defined as follows:

$$f^0(M) = M; For\ i \ge 0, f^{i+1}(M) = f^i(M) \cup f(f^i(M));$$

$$L(\mathcal{A}) = f^*(M) = \cup_{i \ge 0} f^i(M).$$

The family of picture languages generated by array flat splicing systems is denoted by $L(AFS)$.

We illustrate the definitions and the working of array flat splicing systems with an example.

Example 1. Consider the array flat splicing system $\mathcal{A}_C$ with alphabet $\{a, b\}$ and the initial set $M = \{\begin{array}{cc} a & b \\ b & a \end{array}, \begin{array}{cc} b & a \\ a & b \end{array}\}$. The column flat splicing rules are c_1, c_2 where $c_1 = (\begin{array}{c|c} a & b \\ b & a \end{array} - \begin{array}{c|c} a & b \\ b & a \end{array}), c_2 = (\begin{array}{c|c} b & a \\ a & b \end{array} - \begin{array}{c|c} b & a \\ a & b \end{array})$. The row flat splicing rules are r_1, r_2 where $r_1 = (ab|ba - ab|ba)$, $r_2 = (ba|ab - ba|ab)$.

We illustrate the working of the array flat splicing system $\mathcal{A}_C$. The column flat splicing rule c_1 is applicable to the pair $(\begin{array}{cc} a & b \\ b & a \end{array}, \begin{array}{cc} b & a \\ a & b \end{array})$ since both the arrays have equal number of rows and the second array in the pair begins with the column $\begin{array}{c} b \\ a \end{array}$ and ends with the column $\begin{array}{c} a \\ b \end{array}$, as required in the rule c_1. The first array is "cut" between the columns $\begin{array}{c} a \\ b \end{array}$ and $\begin{array}{c} b \\ a \end{array}$ while the second array is "inserted" between them, yielding the array $\begin{array}{cccc} a & b & a & b \\ b & a & b & a \end{array}$. Likewise, the application of rule c_2 to the pair $(\begin{array}{cc} b & a \\ a & b \end{array}, \begin{array}{cc} a & b \\ b & a \end{array})$ yields the array $(\begin{array}{cccc} b & a & b & a \\ a & b & a & b \end{array})$. If we now apply this rule c_1 to the pair $(\begin{array}{cccc} a & b & a & b \\ b & a & b & a \end{array}, \begin{array}{cc} b & a \\ a & b \end{array})$, it will expand the "chess board pattern" columnwise. On the other hand, the row flat splicing rules r_1, r_2 can be used to expand the "chess board pattern" rowwise. For example, the sequence of rules r_1, r_2, r_1 could be applied to the pair of arrays $(\begin{array}{cccc} a & b & a & b \\ b & a & b & a \end{array}, \begin{array}{cccc} b & a & b & a \\ a & b & a & b \end{array})$ since both the arrays have the same number of columns. In fact, the first array is "cut" between the first row $a\ b\ a\ b$ and the second row $b\ a\ b\ a$. The second array satisfies the requirements of the sequence of rules r_1, r_2, r_1. The second array therefore can be "inserted" into the first array to yield the array $\begin{array}{cccc} a & b & a & b \\ b & a & b & a \\ a & b & a & b \\ b & a & b & a \end{array}$. Proceeding like this, we compute the successive terms $f^0(M), f^1(M), \cdots$. In fact

$$f^0(M) = M = \{\begin{array}{cc} a & b \\ b & a \end{array}, \begin{array}{cc} b & a \\ a & b \end{array}\},$$

$$f^1(M) = M \cup f(M) = \{\begin{array}{cc} a & b \\ b & a \end{array}, \begin{array}{cc} b & a \\ a & b \end{array}, \begin{array}{cccc} a & b & a & b \\ b & a & b & a \end{array}, \begin{array}{cccc} b & a & b & a \\ a & b & a & b \end{array}, \begin{array}{cc} a & b \\ b & a \\ a & b \\ b & a \end{array}, \begin{array}{cc} b & a \\ a & b \\ b & a \\ a & b \end{array}\},$$

$$\cdots.$$

Thus the picture language $L(\mathcal{A}_C) = f^*(M)$ consists of rectangular "chess board" patterns of even side length over the symbols a, b with a standing for a white unit square and b for a black unit square. Three such "chess board" patterns are shown in Fig. (1).

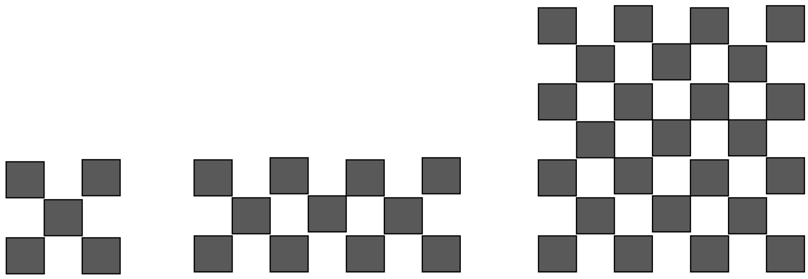

Fig. 1. Chess Board patterns

In what follows, we obtain some properties on the generative power of array flat splicing system by comparing this with certain other well-known picture array generative models. We first informally recall here the two-dimensional right-linear grammar (2RLG) [3] (originally introduced in [8]). There are two sets of rules in a 2RLG grammar: horizontal and vertical rules that correspond to Chomsky regular grammars. This model operates in two phases with the first phase generating a (horizontal) string over intermediate symbols using the horizontal rules and then the vertical rules are applied in parallel generating the columns of a rectangular array made of terminal symbols. We denote the family of array languages generated by two-dimensional right-linear grammars by $L(2RLG)$. We now show that there is a picture language generated by an array flat splicing system while no two-dimensional right-linear grammar can generate it.

Theorem 1. $L(AFS) \setminus L(2RLG) \neq \emptyset$.

Proof. We consider the picture language L_1 consisting of picture arrays with even sides, of the form $M_1 \diamond M_2$, where M_1 is a $m \times p$ rectangular array over the symbol a while M_2 is a $n \times q$ rectangular array over the symbol b, where $m, n \geq 1$ and $p, q \geq 2$. A member of L_1 is shown in Fig. 2. The language L_1 is generated by the AFS S_1 with alphabet $\{a\}$, initial array $\begin{smallmatrix} a & a \\ b & b \end{smallmatrix}$, column flat splicing rules $(\begin{smallmatrix} a \\ b \end{smallmatrix} | \begin{smallmatrix} a \\ b \end{smallmatrix} - \begin{smallmatrix} a \\ b \end{smallmatrix} | \begin{smallmatrix} a \\ b \end{smallmatrix})$, $(\begin{smallmatrix} a \\ a \end{smallmatrix} | \begin{smallmatrix} a \\ a \end{smallmatrix} - \begin{smallmatrix} a \\ a \end{smallmatrix} | \begin{smallmatrix} a \\ a \end{smallmatrix})$, $(\begin{smallmatrix} b \\ b \end{smallmatrix} | \begin{smallmatrix} b \\ b \end{smallmatrix} - \begin{smallmatrix} b \\ b \end{smallmatrix} | \begin{smallmatrix} b \\ b \end{smallmatrix})$, and row flat splicing rules $(\, aa \,|\, aa - bb \,|\, bb \,)$. It can be seen that the column flat splicing rules and the row flat splicing rule can respectively be used to expand an array columnwise and rowwise, starting with the initial array $\begin{smallmatrix} a & a \\ b & b \end{smallmatrix}$. But the language L_1 cannot be generated by any $2RLG$ since in the vertical derivation, there is no control which will synchronize the application of regular rules in order to generate a row of b's in passing from generation of a's to generation of b's in the columns. This proves the result. □

We now informally recall the notion of a local picture language [3]. Extending the notion of local string language, local picture language L is defined in terms

```
a a a a a a
a a a a a a
b b b b b b
b b b b b b
```

Fig. 2. A member of the language L_1

of "tiles" that are square arrays of side length two such that L contains all picture arrays that contain only the tiles that are allowed in defining the picture language L. The family of local picture languages is denoted by LOC. We now show that there is a picture language which is not in LOC but which can be generated by an array flat splicing system.

Theorem 2. $L(AFS) \setminus LOC \neq \emptyset$.

Proof. We consider the picture language L_2 consisting of picture arrays over the symbol a, with each of these arrays having three columns and an even number of rows. A member of L_2 is shown in Fig. 3. This language is generated by

```
a a a
a a a
a a a
a a a
```

Fig. 3. A member of the language L_2

the AFS S_2 with alphabet $\{a\}$, initial array $\begin{matrix} a & a & a \\ a & a & a \end{matrix}$ and a row splicing rule $r = (aa|aa - aa|aa)$. On the other hand, L_2 is not a local picture language. Suppose it is, then L_2 will be defined by a set of tiles which will include a tile $\begin{matrix} a & a \\ a & a \end{matrix}$ besides the corner tiles and border tiles. But this will mean that L_2 can contain picture arrays over a having more than three columns. □

The notion of splicing on strings originally introduced by Head [4] has been extensively investigated theoretically [5]. Extending this notion, the operation of splicing applied to picture arrays (also called images) has been introduced in [2] and the generative power and other properties of an array splicing system, called H array splicing system, have been investigated in [2]. The family of picture languages generated by H array splicing systems is denoted by $L(HASL)$. The concept of array flat splicing considered here is different from the splicing notion studied in [2]. Yet we find the two families $L(HASL)$ and $L(AFS)$ have nonempty intersection.

Theorem 3. $L(AFS) \cap L(HASL) \neq \emptyset$.

Proof. The picture language of even-sided chess board patterns desctribed in Example 1 is in $L(AFS)$. In [2], a H array splicing system is given (Example 3.3 in [2]) generating this picture language. □

4 Conclusion and Discussion

The concept of flat splicing introduced in [1], especially, the alphabetic case, is extended to arrays here and a new model of picture array generation, called array flat splicing system is introduced. Comparison with other kinds of picture generating models (for example, models in [9]), closure properties of the family $L(AFS)$ remain to be investigated. Although the study here has been theoretical, possible application to generation of patterns such as "floor designs" using AFS can also be examined.

Acknowledgments. This work was supported by National Natural Science Foundation of China (61033003 and 61320106005), Ph.D. Programs Foundation of Ministry of Education of China (20120142130008).

References

1. J. Berstel, L. Boasson, I. Fagnot, Splicing systems and the Chomsky hierarchy, Theoret. Comput. Sci., 436 (2012) 2 -22.
2. P.H. Chandra, K.G. Subramanian, D.G. Thomas, Parallel splicing on images, Int. J. Pattern Recognition Artificial Int., 18(6) (2004) 1071-1091.
3. D. Giammarresi, A. Restivo, Two-dimensional languages. In: Rozenberg G., Salomaa, A. (Eds.) Handbook of Formal Languages, Vol. 3, Springer Verlag, pp. 215-267, 1997.
4. T. Head, Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviours, Bull. Math. Biol. 49 (1987) 735-759.
5. T. Head, 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, Vol. 2, Ch. 7, Springer Verlag, pp. 295-358, 1997.
6. Gh. Păun, G. Rozenberg and A. Salomaa, Computing by splicing, Theoret. Comput. Sci., 168 (1996) 321-336.
7. G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages Vol. 1–3, Springer, Berlin, 1997.
8. G. Siromoney, R. Siromoney and K. Krithivasan, Abstract families of matrices and picture languages, Comput. Graph. Imag. Process. 1 (1972) 284-307.
9. K.G. Subramanian, R.M. Ali, M. Geethalakshmi, A.K. Nagar, Pure 2D picture grammars and languages. Discrete Appl. Math. 157 (2009) 3401-3411.
10. K.G. Subramanian, K. Rangarajan and M. Mukund (eds.): Formal Models, Languages and Applications. Series in Machine Perception and Artificial Intelligence, Vol. 66, World Scientific Publishing, 2006.