JCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Special Issue, 5, March 2019 E-ISSN: 2347-2693

Two – Dimensional Grammars Based on Patterns

Christopher Kezia Parimalam[1*], J.D. Emerald[2]

[1,2]Department of Mathematics, Queen Mary's College, Chennai, India

[*]*Corresponding Author: keziamaths@gmail.com, Tel.: 9840538143*

DOI: https://doi.org/10.26438/ijcse/v7si5.216220 | Available online at: www.ijcseonline.org

Abstract- A language generating model called Pattern languages was introduced by Dassow , motivated by Angulin's Pattern languages that use strings as language descriptors. Investigation of patterns has been of relevance in many areas such as combinatorics on words, learning theory and so on. Pattern grammars provide an alternative method in defining languages in automaton theory. Several methods to generate two-dimensional languages known as array languages or picture languages have been defined and investigated in literature and they have been extending the techniques and results of formal string language theory. A picture is defined as a rectangular array of terminal symbols in a rectangular plane. In this paper we extend the Pattern languages defined for strings by Dassow, to a two-dimensional case, while the simplicity and compactness of their descriptors as defined in one dimensional case are preserved. Hence, Two-dimensional Pattern languages are defined and investigated for their closure properties based on array operations.

Keywords—Two-dimensional patterns, Component, Two-dimensional axioms, Catenation, Factorization of arrays .

I. INTRODUCTION

String grammars are studied widely in the field of Computer Science, Mathematics and linguistics since they describe various forms of language constructs. The string grammar plays a significant and crucial role in the analysis of any language especially in high level languages.The study of syntactic methods of describing pictures considered as connected, digitized finite arrays in two-dimensional plane has been of great interest [2]. Picture languages generated by array grammars or recognized by array automata have been advocated since 1970 for problems arising in the frame work of pattern recognition and image processing.

In this context, a pattern α is a string over an alphabet $\{x_1, x_2, x_3, \cdots\}$ of variables. For some finite alphabet Σ of terminal symbols, the pattern language described by α (with respect to Σ) is the set of all words over Σ that can be derived from α by uniformly substituting the variables in α by non-empty terminal words was introduced by Angu3333lin[1]. A new generative device called Pattern grammars was introduced by Dassow et.al., [4] to modify the pattern languages defined by Angulin [1] namely, not allowing the replacing of variables by arbitrary strings, but to adopt the following strategy more usual in formal language theory: start from a finite set of given strings(axioms), replace them by variables in a given set of pattern(s), all strings generated (identified) in this way constitute the associated languages. Intuitively, this way of obtaining languages is related to parallel rewriting (all occurrences of a given variable are replaced by the same string, hence in parallel). In this paper we generalize the concept of pattern grammars as language descriptors to two-dimensional case, while preserving the simplicity and compactness of the descriptors.

II. PRELIMINARIES

In this section, we briefly recall the standard definitions and notations regarding one- and two- dimensional words and languages as dealt in [3].

For a finite alphabet Σ, a string or word (over Σ) is a finite sequence of symbols from Σ, and ϵ stands for the empty string. The notation Σ^+ denotes the set of all nonempty strings over Σ, and $\Sigma^* = \Sigma^+ \cup \{\epsilon\}$. For the concatenation of two strings $\omega_1.\omega_2$ or $\omega_1\omega_2$. We say that a string $v \epsilon \Sigma^*$ is a factor of a string $\omega \epsilon \Sigma^*$ if there are $u_1, u_2 \epsilon \Sigma^*$ such that $\omega = u_1.v.u_2$. If u_1 or u_2 is empty string then v is a prefix (or a suffix respectively) of ω .the notation $|\omega|$ stands for the length of a string ω.

A two-dimensional word (or array) over Σ is a tuple $W = \big((a_{1.1}, a_{1.2}, \cdots, a_{1.n}), (a_{2.1}, a_{2.2}, \cdots, a_{2.n}), \cdots, (a_{m.1}, a_{m.2}, \cdots, a_{m.n})\big)$, where m, n$\epsilon\mathbb{N}$ and, for every i, $1 \le i \le m$, $1 \le j \le n$, $a_{i.j} \epsilon \Sigma$. We define the number of columns (or width) and number of rows (or height) of W by $|W|_c = n$ and $|W|_r = m$, respectively. The empty array is denoted by λ, i.e., $|\lambda|_c = |\lambda|_r = 0$. For the sake of convenience, we denote W by $[a_{i.j}]_{m,n}$ or by a matrix of one of the following form:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \qquad \begin{matrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{matrix}$$

If we want to refer to the j^{th}symbol in row i of the array When we use W[i,j] =$a_{i.j}$.

By Σ^{++}, we denote the set of all nonempty arrays over Σ, and Σ^{**} =Σ^{++} ∪ {λ}.Every subset L⊆ $\Sigma^{**}$is an array language.

Let W = $\left[a_{i.j}\right]_{m,n}$ and $W' = \left[b_{i.j}\right]_{m',n'}$ be two non-empty arrays over Σ. The column concatenation of W and W', ⦶denoted by W W', is undefined if $m \neq m'$ and the array is

$$\begin{matrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & b_{1,1} & b_{1,2} & \cdots & b_{1,n'} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & b_{2,1} & b_{2,2} & \cdots & b_{2,n'} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} & b_{m',1} & b_{m',2} & \cdots & b_{m',n'} \end{matrix}$$

Otherwise. The row concatenation of W and W', denoted by W ⊖ W', is undefined if $n \neq n'$ and the array is

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \\ \\ b_{1,1} & b_{1,2} & \cdots & b_{1,n'} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,n'} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m',1} & b_{m',2} & \cdots & b_{m',n'} \end{bmatrix}$$

Otherwise. Intuitively speaking, the vertical lines and the horizontal lines in the symbols ⦶ and ⊖ respectively, indicate the edge where the array are concatenated. In order to denote that, e.g., U⊖ V is undefined, we also write U⊖ V = undef.

Example 1

$W_1 = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}$, $W_2 = \begin{bmatrix} a & b \\ b & b \end{bmatrix}$, $W_3 = [a \quad b \quad a]$, $W_4 = [a \quad a]$

Then, $W_1 \ominus W_2 = W_1 \ominus W_4 = W_2 \ominus W_3 = undef$, $Similarly$,W_1 ⦶ $W_3 = W_1$ ⦶ $W_4 = W_2$ ⦶ $W_3 =$ W_2 ⦶ $W_4 = undef$.

Now, ($W_1 \ominus W_3$) ⦶($W_2 \ominus W_4$) = $\begin{bmatrix} a & b & c & a & b \\ d & e & f & b & b \\ a & b & a & a & a \end{bmatrix}$ = (W_1 ⦶ W_2) ⊖ (W_3 ⦶ W_4)

The row and column catenation for array languages L_1 and L_2 is defined by $L_1 \ominus L_2 = \{ U \ominus V \mid U \in L_1 \; and \; V \in L_{2,} U \ominus V \neq undef\}$ and $L_1 L_2$ ={ U⦶V |U ∈ L_1 and $V \in L_{2,}$U ⦶ V ≠ undef}. For an array language L and K∈ ℕ, $L^{\ominus k}$, denotes the k-fold row concatenationof L, i.e.,$L^{\ominus k} =$ $L_1 \ominus L_2 \ominus \cdots \ominus L_k . L_i = L, 1 \leq i \leq k$. The k-fold column concatenation is defined anologously denoted by $L^{⦶k}$. The row and column concatenation closure of an array language L is defined by $L^{\ominus * ⦶} = \bigcup_{k=1}^{\infty} L^{\ominus ⦶ k}$and $L^* = \bigcup_{k=1}^{\infty} L^k$, respectively. Obvisously, the row and column concatenation closure of an array language correspond to thr Kleene clouse of a string language.

Infact, it turns out that characteristic factorisation provides most promising approach to formalise how a two-dimensional word satisfies atwo-dimensional pattern. For a pattern $\alpha = \begin{bmatrix} \delta_1 & \delta_2 \\ \delta_1 & \delta_2 \end{bmatrix}$ if $\alpha = ([\delta_1]⦶[\delta_2]) \ominus ([\delta_1]⦶[\delta_2])$, a characteristic factorisation for atwo- dimensional word U for α is a a factorisation of the form U=([w_1 ⦶ w_2]) ⊖ ([w_1 ⦶ w_2]) .We say the factorisation is of column-row type . On the otherhand if$\alpha = ([\delta_1] \ominus [\delta_1])⦶([\delta_2] \ominus \delta_2])$,then U=($[w_1] \ominus [w_1]$)⦶($[w_2] \ominus [w_2]$), the factorisation is said to be of row-column type. A column-row factorisation preserves horizontal neighbourship relation of variables, but not necessarily the vertical neighbourship relation. It is vice-versa for a row-column factorisation.If a two-dimensional word U can be diassembled both into column-row as well as row-column factorisation and U=([v_1 ⦶ v_2]) ⊖ ([v_1 ⦶ v_2]) and U=([v_1] ⊖ [v_1]) ⦶([v_2] ⊖ [v_2]) . We say the factorisation is proper.

For the definition of two-dimensional patterns, we use the same set of variables Δ used in the definition of one-dimensional pattern languages. An array pattern is a non-empty two-dimensional word over Δ and a terminal array is a non-empty two-dimensional word over Σ. The different kinds of pattern images can be intrepreted to represent a grid to be placed over a terminal array. The vertical lines of the grid denote the column concatenation and the horizontal grids their row concatenation. Every area of the grid represents the occurrence of a variable δ_i in the array pattern or to be more specific the axiom substituted for the variables $\delta_i{}'s$. In particular the two rectangular areas of the grid that correspond to the occurences of the same variable must have identical content. Thus we can say that the terminal array W is a certain type of image of an array pattern as a tiling of W. W is said to satisfy a given pattern array α with n different variables if and only if n tiles are alloted to represent the n variables of α, and combining the tiles as indicated by the stucture of α, yeids W.

III. TWO-DIMENSIONAL PATTERN LANGUAGES

In this section we have extended the pattern languages defined by Dassow et.al., [4] from the one-dimensional to two dimensional case.

A pattern language is obtained by starting from a set of words called axioms and substituting them in a given P uniformly to obtain P(A). Now the set of words in P(A) is uniformly substituted in the given pattern to obtain P(P(A)). The process is continued and the the language defined by the grammar is

L(G)=A ∪ $P(A) \cup P(P(A) \cup P(P(P(A))) \cdots$

In one dimensional pattern the axiom is substituted instead of the pattern variable.The basic operation of substitution of a single variable in a word by another word cannot be that easily extended to a two-dimensional case. Thus we define a two dimensional pattern in a following manner.

A two-dimensional pattern is a mxn array, and the axionm set is a set of mxn dimensional arrays which satisfy the concatenation rules for arrays either by row, column, row-column or column-row factorisation or proper.

Definition 3.1: A two-dimensional pattern grammar is a four tuple $G = (\Sigma, \Delta, \mathrm{A}, \mathrm{P})$ where

$\Sigma =$ Set of terminal alphabets

$\Delta =$ Set of pattern variables

$\mathrm{A} =$ Set of two-dimensional axioms which satisfy the patterns defined by their factorisation

$\mathrm{P} =$ The two-dimensional pattern, defined by pattern variables only.

i.e., $\mathrm{P} = \begin{bmatrix} \delta_{1,1} & \delta_{1,2} & \cdots & \delta_{1,n} \\ \delta_{2,1} & \delta_{2,2} & \cdots & \delta_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{n,1} & \delta_{n,2} & \cdots & \delta_{m,n} \end{bmatrix}$ all the m x n entries are defined by a pattern variable.

The pattern follows one of the concatenation (factorisation) (i) row, (ii) column (iii) row-column (iv)column-row (v) proper (where it satisfies both row-column and column-row).

For a given pattern P and a language $L \subseteq \Sigma^{**}, P(L)$ is the set of two-dimensional arrays obtained by replacing each occurrence of variables in the pattern of P by the arrays in A, the different occurrences of the same variable being replaced by the same array.

The language generated by G, denoted by $L(G)$, is the smallest language $L \subseteq \Sigma^{**}$ for which we have:

(i) $A \subseteq L$

(ii) P(L) $\subseteq L$.

Thus, L (G) consists of all arrays which can be obtained starting from the axioms and using finitely many times the patterns, in the way described in the pattern languages for strings. L(G)=A∪ $P(A) \cup P(P(A) \cup P(P(P(A))) \cdots$

A language $L(G)$as above is called a Pattern Language and we denote it as

$\mathcal{L}_{p,f}$ where f denotes the array factorization type of the pattern defined; and its family is denoted by $\mathcal{L}_{\mathrm{P,f}}$.

Definition 3.2: Let $G = (\Sigma, \Delta, \mathrm{A}, \mathrm{P})$ be a two-dimensional Pattern grammar, $\mathrm{P} \in \Delta^{++}$, be an two-dimensional pattern. We define the variants of two-dimensional pattern languages as follows:

- $\mathcal{L}_{p,r}$={W ∈ $\sum^{++}$|W is obtained by substitution of the axioms into patterns that follow row factorization}
- $\mathcal{L}_{p,c}$={W ∈ $\sum^{++}$|W is obtained by substitution of the axioms into patterns that follow column factorization}
- $\mathcal{L}_{p,rc}$={W ∈ $\sum^{++}$|W is obtained by substitution of the axioms into patterns that follow row-column factorization}
- $\mathcal{L}_{p,cr}$={W ∈ $\sum^{++}$|W is obtained by substitution of the axioms into patterns that follow column-row factorization}
- $\mathcal{L}_{p,p}$={W ∈ $\sum^{++}$|W is obtained by substitution of the axioms into patterns that follow proper factorization}

For some fixed Pattern P, we see that $\mathcal{L}_{\mathrm{P,p}}$ is a subset of $\{\mathcal{L}_{\mathrm{P,rc}}, \mathcal{L}_{\mathrm{P,cr}}\}$.

Examples:

1. $\mathrm{G}_1 = (\{a\}, \{\delta\}, \{[a]\}, \mathrm{P}\}$, P= $\begin{bmatrix}\delta \\ \delta\end{bmatrix}_{\mathrm{r}}$

$L(G_1) = L_{P,r} = A \cup$ set of all arrays of dimension $[a\ \ a\ \ a \cdots a]^T$ of dimension 2m x 1.

2. $\mathrm{G}_2 = (\{a\}, \{\delta\}, \{[a]\}, \mathrm{P}), \mathrm{P} = [\delta\ \ \delta]_c$

$L(G_2) = L_{P,c} = A \cup$ set of all arrays of dimension $[a\ \ a\ \ a \cdots a]$ of dimension 1 x 2n.

$$L(G_2) = L(G_1)^T$$

3. $G_3 = (\{a,b\}, \{\delta_1, \delta_2\}, \{[a\ \ a], \begin{bmatrix} b & b \\ b & b\end{bmatrix}\}, \mathrm{P})$, P = $\begin{bmatrix}\delta_1 & \delta_2 \\ \delta_2 & \delta_1\end{bmatrix}_{\mathrm{rc}}$

$i.e, P = (\delta_1 \ominus \delta_2) \oplus (\delta_2 \ominus \delta_1)$

L(G$_3$)= $\mathrm{L}_{\mathrm{P,rc}}$ = A ∪ set of all arrays of dimension $(\mathrm{sm}_1 + \mathrm{lm}_2) \times 2\mathrm{kn}$.

$= \begin{bmatrix} w_i w_j \\ w_j w_i \end{bmatrix}$

Where $\mathrm{m_i} \times$ n, denotes the order of the axiom set,k denotes the number of times a pattern is applied, s,t are such that $0 \leq \mathrm{s}, \mathrm{t} \leq \mathrm{k}$.

4. G_4 = ({a, b}, $\{\{\delta_1, \delta_2\}, \left\{\begin{bmatrix} a \\ b \\ a\end{bmatrix}, \begin{bmatrix} b & b \\ a & a \\ b & a\end{bmatrix}\right\}, P)$, $P = \begin{bmatrix}\delta_1 & \delta_2 \\ \delta_1 & \delta_2\end{bmatrix}_{cr}$

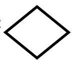
i.e., P= $(\delta_1 \oplus \delta_2) \ominus (\delta_2 \oplus \delta_1)$

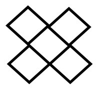
$L(G_4) = L_{P,cr} = A \cup$ set of all arrays $\begin{bmatrix} W \\ W \end{bmatrix}$ of dimension $2km \times 2(sn_1 + tn_2)$.

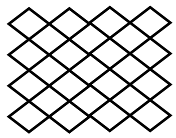
Where $m \times n_i$, denotes the order of the axiom set,k denotes the number of times a pattern is applied, s,t are such that $0 \leq \mathrm{s}, \mathrm{t} \leq \mathrm{k}$.

5. $G_5 = (\{a\},\{\delta\ \},\{[a]\},P), P = \begin{bmatrix}\delta & \delta \\ \delta & \delta\end{bmatrix}_p$

$(G_5) = L_{P,p} = A\ \cup of\ all\ arrays\ of$
$a's\ with\ dimension\ 2set\ n\ \times 2n$

if $a =$ 

then P(A) = 

P(P(A))= 

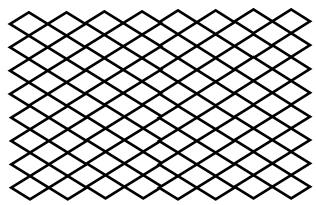
P(P(P(A))) = 

Figure 1: Picture generation of the Two-dimensional pattern grammar

IV. PROPERTIES OF TWO-DIMENSIONAL PATTERN LANGUGES

Theorem 4.1: For every Two-dimensional Pattern grammar generating a Pattern language $L_{\mathrm{P,f}}$, there exist a two-dimensional Pattern grammar generating $[L_{\mathrm{P,f}}]^T$.

Proof: Let $G = (\Sigma,\ \Delta,\ \mathrm{A},\ P)$ be a Two-dimensional Pattern grammar generating$\mathrm{L}_{\mathrm{P,f}}$. We construct a Two-dimensional Pattern grammar $G' = (\Sigma,\ \Delta,\ \mathrm{A}^T,\ P')$,
where

$$P' = \begin{cases} P_c^T\ if\ P = P_r\ ;\ P_r^T\ if\ P = P_c \\ P_{cr}^T\ if\ P = \ P_{rc};\ P_{rc}^T\ if\ P = P_{rc} \\ P_p^T\ if\ P = \ P_p \end{cases}$$

It can be easily verified from the definition of P', that

$$L_{P',\mathrm{f}} = [L_{\mathrm{P,f}}]^T.$$

Corollary 4.1 Let $\mathrm{G} = (\Sigma, \Delta, \mathrm{A}, \mathrm{P})$ be some a pattern grammar.Then, (i) $\left[L_{p,r}\right]^T = \ L_{P^T,c}$; $(ii)\left[L_{p,c}\right]^T = \ L_{P^T,r}$; (iii) $\left[L_{p,rc}\right]^T = \ L_{P^T,cr}$; (iv) $\left[L_{p,cr}\right]^T = \ L_{P^T,rc}$; .

The first and the second concludes from the examples 1 and 2 given in section3. To prove (iii) and (iv) we consider pattern grammar with Pattern P_{rc} = $\begin{bmatrix}\delta_1 & \delta_2 \\ \delta_2 & \delta_1\end{bmatrix}$ and the Axiom = $\left\{[a], \begin{bmatrix}a \\ a\end{bmatrix}\right\}$,

Then P(A) = $\begin{bmatrix}a & a \\ a & a \\ a & a\end{bmatrix}$,

P(P(A)) = $\begin{bmatrix}a & a & a & a \\ a & a & a & a \\ a & a & a & a \\ a & a & a & a \\ a & a & a & a \\ a & a & a & a\end{bmatrix}$.

$L_{p,rc}\ = \{\ All\ arrays\ of\ \ a's\ with\ dimension$ $(\mathrm{sm}_1 + \mathrm{lm}_2) \times$ kn; Where m_i x n, denotes the order of the axiom set,k denotes the number of times a pattern is applied, s,t are such that $0 \leq \mathrm{s,t} \leq \mathrm{k}$.}

$\mathrm{P}_{\mathrm{cr}}^{\mathrm{T}} = \begin{bmatrix}\delta_1 & \delta_2 \\ \delta_2 & \delta_1\end{bmatrix}$ and Axiom = $\{\{[a], [a\ a]\}$, P(A) = $\begin{bmatrix}a & a & a \\ a & a & a\end{bmatrix}$; P(P(A)) = $\begin{bmatrix}a & a & a & a & a & a \\ a & a & a & a & a & a \\ a & a & a & a & a & a \\ a & a & a & a & a & a\end{bmatrix}$

$L_{p,rc}\ = \{\ All\ arrays\ of\ \ a's\ with\ dimension\ kn\ \times$ $(sm_1 + lm_2)$; Where n x m_i, denotes the order of the axiom set,k denotes the number of times a pattern is applied, s,t are such that $0 \leq\ s,t\ \leq\ k$.}. = $\left[L_{p,rc}\right]^T$. Also, $\left[L_{p,cr}\right]^T = L_{P^T,rc}$.

Proposition 4.1: Let G=(∑,ΔA,P) be a two-dimensional Pattern grammarΣ be an alphabet, and 'A' given set oftwo-dimensional axioms then neither $L_{p,rc}$ nor $L_{p,cr}$ are closed under transposition.

Proof: Consider a Pattern grammar in which the Pattern P $=\begin{bmatrix}\delta_1 & \delta_2 \\ \delta_2 & \delta_1\end{bmatrix}$, we see $\mathrm{P}^{\mathrm{T}} = \begin{bmatrix}\delta_1 & \delta_2 \\ \delta_2 & \delta_1\end{bmatrix}$; P = P^{T}; then we see that from the above $\left[L_{P,rc}\right]^T = \ L_{P^T,cr} = \notin\ \mathcal{L}_{P,rc}$.

Proposition 4.2: For any alphabet Σ, and a given set of Axioms $\mathcal{L}_{\mathrm{P,p}}$ is closed under transposition.

Proof: Given a two-dimensional Pattern grammar G=(∑,ΔA,P) and a two-dimensional Pattern P, such that P $=\mathrm{P}^{\mathrm{T}}$, since P satisfies both row-column and column-row factorisation, the substitution of the axioms into the two-dimensional pattern yeids the same set of arrays for each P(A), P(P(A))…Thus it can be easily verified that $[\mathrm{L}_{\mathrm{P,p}}]^{\mathrm{T}}=\mathrm{L}_{\mathrm{P}^{\mathrm{T}},\mathrm{p}} = \notin \mathcal{L}_{\mathrm{P,p.}}$

Theorem 4.2 For a given set of alphabets and a set of two-dimensional axioms A.

$L_{p,r}$ is closed under row concatenation⊖;

$L_{p,c}$ is closed under column concatenation ⦶ ;

$L_{p,rc}$ is closed under concatenation⊖;

$L_{p,cr}$ is closed under concatenation ⦶;

$L_{p,p}$ is closed under both row and column concatenation;

Proof: We prove this only for one case. The others follow similarly. Consider the case of $L_{p,rc}$. Let G_1 and G_2be two-dimensional Pattern grammarsin which the Patterns are defined as the row- columnfactorisation.LetP_1 and P_2 be two pattern variables with dimension m_1 x n_1, m_2 x n_2 respectively. The language generated by their grammars $L(G_1)$ and $L(G_2)$ are words obtained by substituting each variable in the pattern by two-dimensional words of dimension m_i x n. Hence all resulting two-dimensional words will have equal number of columns (kn). Thus $n_1 = n_2 = n$. Hence only concatenation of rows will be defined for the Patterns and the resulting patterns will be of dimension m_i x n and satisfies row -column factorisation type that belong to $L_{p,rc}$.

Corollary 4.2 For a given set of alphabet,and a set of two-dimensional axioms A.
$L_{p,r}$ is not closed under column concatenation$\mathbin{⦶}$;
$L_{p,c}$ is not closed under row concatenation $\ominus$;
$L_{p,rc}$ is not closed under column concatenation$\mathbin{⦶}$;
$L_{p,cr}$ is not closed under row concatenation $\ominus$;
$L_{p,p}$ is neither closed under row nor column concatenation;
This directly follows from the above theorem.

V. CONCLUSION

In this Paper we have defined an variant of a pattern language, the Two-dimensional Pattern grammars generating two-dimensional pattern languages and some of the array language properties applicable to them. In future we would be extending our study to compare them with the array grammars develop algorithms to learn the Two-dimensional Pattern languages.

REFERENCES

[1] D. Angulin, "*Finding Pattern common to set of strings*," Journal of Computer and System Sciences 21, 46-62, (1980)
[2] D. Giammarresi, A. Restivo, "*Recognizable picture languages*," International Journal of Pattern Recognition and Artificial Intelligence. 6:31-46, 1992.
[3] H. Fernau, Markus L. Schmid, K.G. Subramanian, "*Two-Dimensional Pattern Languages",* In S.Bensch, F. Drewes, R. Freund, and F.Otto, editors, "Fifth Workshop on Non-Classical Models for Automata and Applications, NCMA, Volume 294 of books@ocg.at, pp.117-132. Osterreichische Computer Gesellschaft, 2013.
[4] J. Dassow, G.Paun, A.Salomaa, "*Grammars based on Patterns*," International Journal of Foundations of Computer Science, Vol 4: 1-14, 1993.

Authors Profile

Mrs.Christopher Kezia Parimalam received the M.Sc. Degree in Mathematics in the year 1991, M.Phil. Degree in 1993.She has been a teaching faculty as lecturer from 1996 to 2009 and as an Assistant Professor from 2009 to 2014 in the Department of Mathematics in various colleges. She is currently pursuing her Ph.D in Queen Mary's College,Chennai-under University of Madras. Her research area includes Formal Languages and automaton theory.

Dr.Mrs. Emerald Princess Sheela J.D. received the M.Sc. Degree in Mathematics in the year 1991, M.Phil. Degree in 1993 and the Ph.D. Degree in 2001 from the University of Madras. She has been on the teaching faculty as Assistant Professor, Department of Mathematics, since 2001 in various colleges and is currently working in Queen Mary's College, Chennai -4. She has presented her research papers in two International Conferences in USA, one in Malaysia and one in UK. She is the author or co-author of about 20 research papers in her field of interest which includes formal languages and automata theory, cryptography and learning theory. She is also guiding three research students for the award of Ph.D. Degree in the University of Madras.