GEOMETRIC STRUCTURE OF TWO DIMENSIONAL HEXAGONAL PATTERNS

[1] K. Bhuvaneswari & *[2]T. Nancy Dora

[1]Associate Professor in Mathematics, 2Research Scholar in Mathematics

Mother Teresa Women's Universaity, Kodaikanal

[1]drkbmaths@gmail.com *[2]**nancydora.t@gmail.com**

***Abstract**:*

A new type of hexagonal regular expressions using operations like linking side borders, land corners and gluing corners for two dimensional languages using arbitrary shapes and tiling operations parameterized by restrictions of the connection interfaces is introduced. Expressions recognizing chains and pretzel shape pictures using hexagonal pixels over the given alphabet are developed. This is an interesting theoretic model comes from a different perspective: the technique of using tiling to describe the syntax and semantics of the computing system. The present approach is an attempt to extend these formalisms to different types of picture languages.

***Keywords**: Side borders, Land corners, Golf corners, Extreme cell, Gluing combinations. Hexagonal Regular expression.*

1. Introduction

The concept of tiling systems as a device for recognizing picture language is investigated in [5]. Tiling systems are a possible generalization of the concept of finite automata for word languages. Regular expressions provide a rich formalism for specifying and analyzing sequential models of computation. Simple expressions using two partial concatenation named row and column concatenation (denoted by $\ominus$ *and* $\mathbin{⦶}$) keep their arguments vertically above each other (horizontally next to each other respectively) provided they have the same width (height, resp...). Additionally, both of these concatenations may be iterated. More powerful type of regular expressions for picture languages, the so- called regular expressions with operators will brought [2]. It deals with different classes of expressions with operators, depending on how operators may be constructed. A novel representation for two dimensional languages was introduced in [3].

It is based on contours and their composition [4]. Based on contour composition a different type of regular expressions of the two-dimensional languages $n2RE$was introduced in [3].

A coordinate system [8] is described which provides a natural means for representing hexagonally organized pixels. The operations of the coordinate system and its computations are shown.

We have introduced here a new type of hexagonal regular expressions using operations like linking side borders, land corners and gluing corners for two dimensional languages using arbitrary shapes and tiling operations parameterized by restrictions of the connection interfaces. We also have introduced expression recognizing chain pictures and developed pretzel - like shapes using hexagonal pixels over the given alphabet.

2. PRELIMINARIES

First we recall the basic definitions of regular expressions defined in [1].

Definition 2.1

Let Σ be a fixed finite alphabet. A picture over $\sum$ of size (m,n) (where $(m,n) \geq 1)$ is a $m \times n$ matrix over $\sum$. For a picture P of size (m,n), we define $\overline{\underline{P}} = m$ and $|P| = n$. We denote the set of all pictures over Σ by $\sum^{+}_{+}$.

Definition 2.2

Let P, Q be pictures of size $(k,l),(m,n)$respectively. If $k = m$, then the column concatenation $P \ \varobar\ Q$ of the two pictures is the $k \times (n+l)$ – picture obtained by appending Q to the right of P.

Definition 2.3

The set $\cap -REG(\Sigma)$ of simple ***regular expressions*** over Σ with typical element r is defined by the following BNF-style rules:

$$r ::= a \,\middle|\, (r_1 \cup r_2) \,\middle|\, (r_1 \ominus r_2) \,\middle|\, (r_1 \varobar r_2) \,\middle|\, (r_1 \cap r_2) \,\middle|\, r^{\varobar +} \,\middle|\, r^{\ominus +}$$

Here a stands for an arbitrary letter from $\sum$. r_1, r_2 are regular expressions. The language generated by such an expression is defined in a straightforward way: For all $a \in \Sigma$, let $\mathcal{L}(a) = \{a\}$ (the singleton of the $1 \times 1 -$ picture a), and for two expressions r and s we define $\mathcal{L}(r \ominus s) = \mathcal{L}(r) \ominus \mathcal{L}(s)$ and so on. The subset of $\cap -REG(\Sigma)$ of monotonic expressions (i.e., expressions without intersection symbol) will be denoted by $REG(\Sigma)$. The classes of languages definable by such expressions will be denoted by the corresponding calligraphic notation $\cap -\mathcal{REG}(\Sigma)$ and $\mathcal{REG}(\Sigma)$.

A finite interactive system (FIS) [6,7]

The different type of 2-dimensional regular expressions, to be defined below, puts constraints on the connection points on the borders of the composed words. These constraints act on the following three types of elements: *side borders*, *land corners* (turning points on the border contour having 3 neighbouring cells outside the word and one neighbouring cell inside) and *golf corners* (turning points on the border contour with 3 neighbouring cells inside and one neighbouring cell outside). The resulting restricted composition operators extend the usual vertical and horizontal composition operators on rectangular word.

On each of the above eligible glueing combinations(x, y), the **constructing formulae** and **restricted formula** are defined.

The set of expressions obtained using all the operators defined so far are denoted by$x2RE$; they represent *two-dimensional regular expressions extended with composition operators on extreme cells*. Dropping the operators involving the extreme cells we get $n2RE$, the basic *new type of regular expressions for two-dimensional words*.

Definition 2.4

Hexagonal grid is an alternative representation of pixel tessellation scheme for the conventional square grid for sampling and representing discredited images. Each pixel is represented by a horizontal deflection followed by a deflection upward and to the right. These directions are represented by a pair of unit vectors $u$and$v$. We refer to this coordinate system as the “h$_2$”

system. Given a pixel with coordinates u, v (assumed integer), the coordinates of the neighbors are illustrated in Figure 2.1.

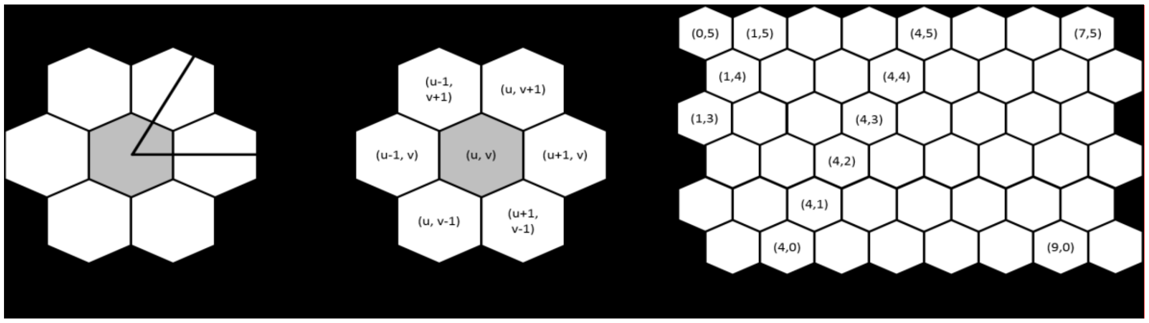


Figure 2.1. (a) A coordinate system based on unit vectors u and v, (b) the neighborhood of a hexagonal pixel and (c) labeling of a hexagonal pixel

3. Regular Expressions for Hexagonal patterns

A new type of two dimensional hexagonal regular expressions is introduced ……………..

Definition 3.1

Let Σ be a finite alphabet. A ***general 2-dimensional hexagonal picture*** is a set of hexagonal unit cells in the 2-dimensional space filled with symbols from the given alphabet. An example is shown in Figure 3.1. Let $h_a$be a hexagonal unit cell filled with letter 'a' over Σ .We denote the set of all hexagonal pictures over Σ by $\Sigma_{h_2}^{**}$.

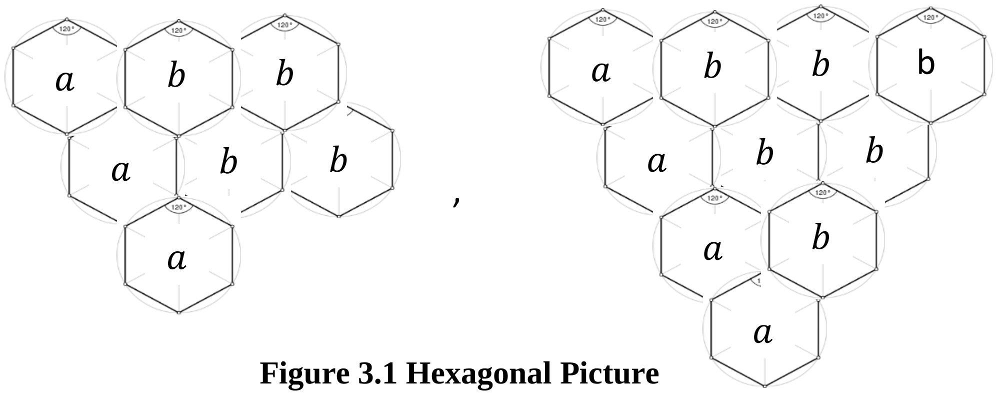


Figure 3.1 Hexagonal Picture

Definition 3.2

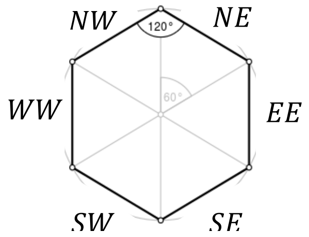
We define a ***general composition operator*** on a two dimensional plane as follows: Given two hexagonal pictures, get new pictures by placing them together such that no interior cell of the first picture may overlap an interior cell of the other.

Definition 3.3

Hexagonal regular expressions are defined by placing the connection points on the borders of the composed pictures. These constraints act on the three types of elements named as Side Borders, Land Corners and Golf Corners. We use the symbol $\mid\mid$ for side border, $\wedge$ for Land Corners and $\vee$ for Golf corners.

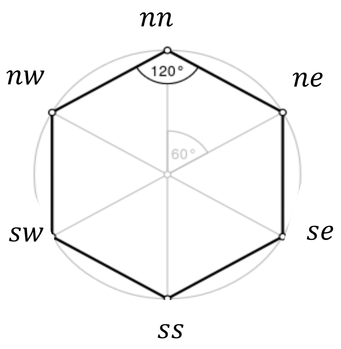
Side Borders:

Elements in $SB = \{NW, NE, EE, SE, SW, WW\}$ where NW stands for "North West border", NE for "North East border", EE for "East border", SE for "South East border", SW for "South West border" and WW for "West border". We denote the Side borders by symbols as $\mid\mid_{NW,}\mid\mid_{NE},\mid\mid_{EE},\mid\mid_{SE},\mid\mid_{SW},\mid\mid_{WW}$.



Land Corners:

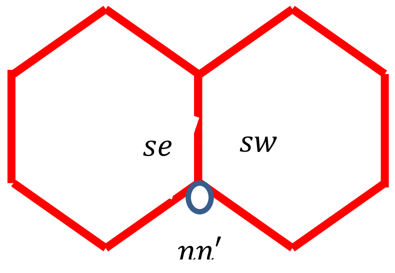
Elements in $LC = \{nw, nn, ne, se, ss, sw\}$, where nw stands for "northwest land corner", nn for "north land corner", ne for "north-east land corner", se for "south-east land corner", ss for "south land corner" and sw for "south-west land corner". We denote the Land Corners by symbols as $\wedge_{nw}, \wedge_{nn}, \wedge_{ne}, \wedge_{se}, \wedge_{ss}, \wedge_{sw}$.



Land Corner $\wedge_{nn}$ is a north land corner seen from inside the picture/word. i.e. the cell at the bottom of the point is inside the word while the cell at the south west and south east are outside the word.

Golf Corners:

Elements in $GC = \{nw', nn', ne', se', ss', sw'\}$, where nw' stands for "north-west golf corner",$nn'$for "north golf corner", ne' for "north-east golf corner", se' for "south-east golf corner", $ss'$for "south golf corner" and sw' for "south-west golf corner".We denote the Golf Corners by symbols as $\vee_{nw'}, \vee_{nn'}, \vee_{ne'}, \vee_{se'}, \vee_{ss'}, \vee_{sw'}$.



Golf Corner$\vee_{nn'}$ is a north land corner seen from outside the word. i.e. the cell at the bottom of the point is outside the word while the cell at the south east and south west are inside the word.

Definition 3.4

A cell is an ***extreme* cell** if it touches at most one other cell in the interior area. A side $(NW, NE, EE, SE, SW\ and\ WW)$ or a corner on the border of a picture is *extreme* if it belongs to an extreme cell and does not touch another cell (side elements with an end point touching another cell are also excluded – they are not extreme).

Definition 3.5

The constraints on glueing the borders are independently allowed on one or more of the following ***Gluing combinations*** $(\boldsymbol{x}, \boldsymbol{y})$**:** We use x and y are different and either they are both in $\{||_{EE}, ||_{WW}\}$ or both in $\{||_{NE}, ||_{SW}\}$ or both in $\{||_{SE}, ||_{NW}\}$ or both are land corners in$\{\wedge_{nw}, \wedge_{nn}, \wedge_{ne}, \wedge_{se}, \wedge_{ss}, and\ \wedge_{sw}\}$ or both are combinations of golf - land corners for the same directions { e.g. $(\wedge_{ss'}, \wedge_{ss})$}.The following are the possible *Gluing* combinations:

Linking Side Borders:

$$L_1 = \{ (||_{EE}, ||_{WW}), (||_{WW}, ||_{EE}), (||_{SE}, ||_{NW}), (||_{NW}, ||_{SE}), (||_{SW}, ||_{NE}), (||_{NE}, ||_{SW})\}$$

Linking Land Corners:

$$L_2 = \{(\wedge_{nn}, \wedge_{sw}), (\wedge_{nn}, \wedge_{se}), (\wedge_{ne}, \wedge_{nw}), (\wedge_{ne}, \wedge_{ss}), (\wedge_{se}, \wedge_{sw}), (\wedge_{se}, \wedge_{nn}) (\wedge_{ss}, \wedge_{nw}), (\wedge_{ss}, \wedge_{ne}), (\wedge_{sw}, \wedge_{se}) (\wedge_{sw}, \wedge_{nn}), (\wedge_{nw}, \wedge_{ne}), (\wedge_{nw}, \wedge_{ss})\}$$

Linking Golf Corners:

$$L_3 = \{(\vee_{ss'}, \wedge_{ss}), ((\vee_{nn'}, \wedge_{nn}), (\vee_{sw'}, \wedge_{sw}), (\vee_{nw'}, \wedge_{nw}), (\vee_{se'}, \wedge_{se}), (\vee_{ne'}, \wedge_{ne}) \}$$

Definition 3.6

On each of the above eligible gluing combinations (x, y) we give a constraint consisting of a propositional logic formula (**Constricting formulae**) $F \in PL(\varphi_1, \varphi_2, \varphi_3, \varphi_4)$ i.e., a Boolean formula built up by starting with the following atomic formulae: $\varphi_1(x,y) = x < y,\ \varphi_2(x,y) = x = y,\ \varphi_3(x,y) = x > y,\ \varphi_4(x,y) = "x\#y"$.

The meaning of the connectors is the following: " $<$ " - left is included into the right; " = " - left is equal to the right; " $>$ "- left includes the right; "#" - left and right overlaps, but no one is included in the other.

For instance: $f(||_{EE} = ||_{WW})g$means "restrict the general composition of $f$and $g$such that the east border of f is identified to the west border of g"; $f(||_{EE} > ||_{WW})g$- the east border of $f$includes all the west border of g, but some east borders of $f$may still be not covered by west borders of g; etc.

We also use the notation, $\varphi_0(x,y) =$ "x!y", where "!" means empty intersection. Actually, this is a derived formula

$$\varphi_1(x,y) \vee \varphi_2(x,y) \vee \varphi_3(x,y) \vee \varphi_4(x,y).$$

We are now in a position to introduce the ***particular composition operators*** induced by the above constricting formulae.

Definition 3.7 (*Restricted compositions)*

A *restriction formula* φ is a boolean combination in $PL(F_{1,\ldots\ldots,}F_n)$, where $F_i$are constricting formulas involving certain eligible glueing combinations $(x_i, y_i) \in Connect$. A *restricted composition operation* $-(F)-$ is the restriction of the general composition to composite pictures satisfying F.A hexagonal picture $h \in f.g$ belongs to $f\ (F)\ g$ if for all glueing combinations (x_i, y_i)occurring in F the contact of the $x_i$border of $f$and y_i border of g satisfy F_i.The non-restricted general composition $f.g$ is the same as $f(.)g$.

Proposition 3.1 The restricted composition operations are not always associative.

$$\left((r_1(||_{EE} = ||_{WW})r_1)(||_{EE} > ||_{WW})r_2\right)(||_{EE} > ||_{WW})r_3))$$
$$\neq (r_1(||_{EE} = ||_{WW})r_1)(||_{EE} > ||_{WW})(r_2(||_{EE} > ||_{WW})r_3)$$

Definition 3.8

The set of hexagonal expressions obtained using all the operators defined so far are denoted by $x2HRE$; they represent *Two dimensional Hexagonal Regular Expressions extended with composition operators on extreme cells*. Dropping the operators involving the extreme cells we get $n2HRE$, the basic new type of regular expressions for two dimensional hexagonal pictures.

Basic new type of hexagonal regular expression:

Let Σ be a finite alphabet. The set $REG\ (\Sigma_{h_2})$ of simple hexagonal regular expressions over Σ with typical element r is defined by

$$r ::= h_a \mid (r_1 \mid\mid r_2) \mid (r_1 \wedge r_2) \mid (r_1 \vee r_2) \mid (r_1(F)r_2) \mid (r_1(.)r_2) \mid r^*(F)$$

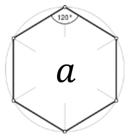
Here h_a stands for a hexagonal picture filled with letter $a \in \Sigma$. r_1 and r_2 are regular expressions.The language generated by such an expression is defined as follows.

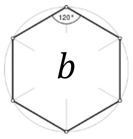
For all $a \in \Sigma,\ L(h_a) = \{h_a\}$.

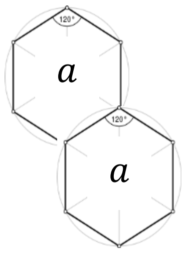
For two expressions $r_1$and r_2 we define $L(r_1||r_2) = L(r_1)||L(r_2)$ and so on. The classes of languages definable by such expressions will be denoted by $REG(\Sigma_{h_2})$.

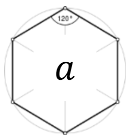
Example 3.1

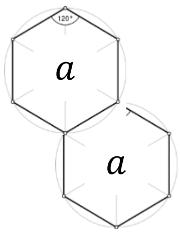
Let $\Sigma = \{a, b, c\}$,

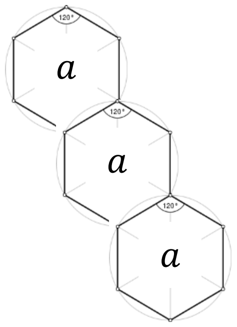
Let h_a = 

h_b = 

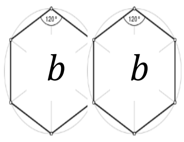
$r_1 = h_a(\mid\mid_{SE} = \mid\mid_{NW})h_a$, 

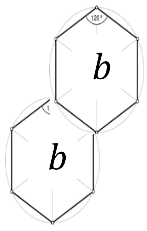
$and\ r_2 = (h_a)^*(\mid\mid_{SE} = \mid\mid_{NW})$: 

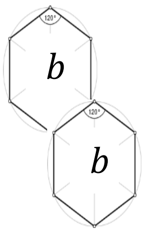




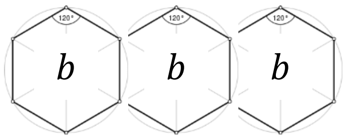
, ,

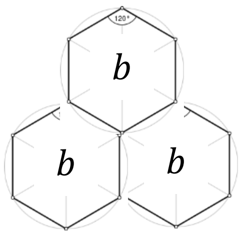
$r_3 = h_b(.)h_b$: 

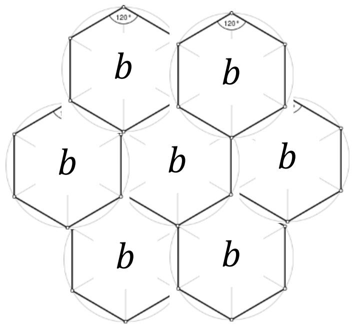
, 

, 

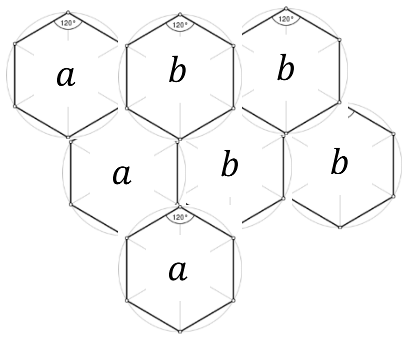
$r_4 = (h_b)^*(.)$:



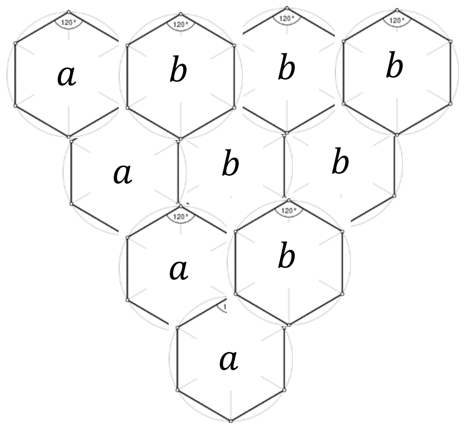
, 

, 

$$r_5 = r_2(\wedge_{ne} > \wedge_{sw} \ \& \ \wedge_{nn} < \wedge_{ss}) r_4$$



,



Theorem 4.1

Pascal triangle on hexagonal grid is generated by the hexagonal regular expression.

Proof:

Pascal triangle is a triangular arrangement of numbers in which each number in any row is the sum of the two numbers immediately above it.

The coordinate values (uv) on hexagonal pixels are arranged as in Figure 4.1

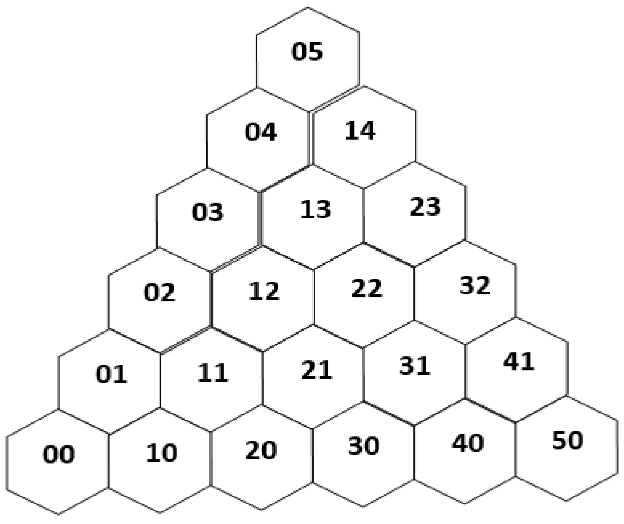


Figure 4.1 : Triangle with coordinate values (uv) on hexagonal pixels

We take hexagonal pixel on coordinate(00) as h_{00} and set

$$h_{00} = 1, h_{0v} = 1, v > 0, h_{u0} = 1, u > 0,$$

$$h_{uv} = h_{u,v-1} + h_{u-1,v}, 1 \le u, v \le n,$$

where n denotes the highest value of u and v.

Now we arrange the hexagonal pixels in the u direction by the regular expression $h_{uv}(\,|\,|_{WW} = \,|\,|_{EE})h_{u+1,v}$ and the hexagonal pixels in the v direction by $h_{uv}(\,|\,|_{NE} = \,|\,|_{SW})h_{u,v+1}$.

The Pascal triangle for n = 5 is generated as in Figure 4.2.

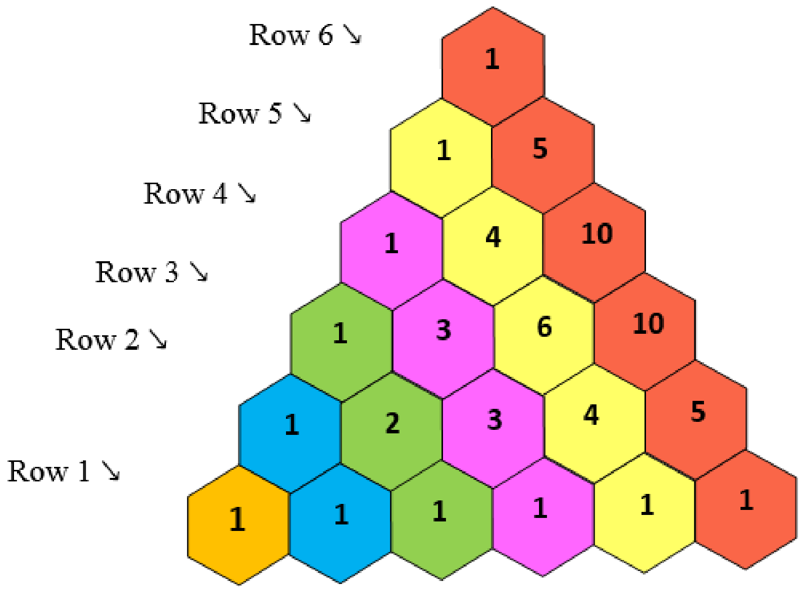


Figure 4.2: Pascal Triangle on Hexagonal Grid

Chains, pretzel-like shapes

A *pretzel picture* is a set of cells such that each cell has precisely 2 touching cells around and no proper subset has this property. The construction of pretzels is in two stages: first, we give a procedure for generating chains and a pretzel is produced by an appropriate connection of two chains.For constructing a chain, we use the strict interpretation of the composition, i.e., the one where the only connecting elements are those specified in the constricting formula. This combined with the restriction to make composition for extreme elements only, leads to a formula for iterative generation of chains: start with a cell and iteratively add one cell at a time, connected via an extreme element.Finally, a pretzel is obtained composing two chains via their extreme elements and asking for equality, to avoid having only one end of a chain connected.

An expression recognizing *chain pictures* over the alphabet $\Sigma = \{a, b, c, d\}$:

C= $(h_a+h_b+h_c+h_d)$*[$(||_{XEE} > ||_{XWW}) \vee (||_{XWW} > ||_{XEE}) \vee (||_{XNE} > ||_{XSW}) \vee (||_{XSW} > ||_{XNE}) \vee (||_{XSE} > ||_{XNW}) \vee (||_{XNW} > ||_{XSE}) \vee (\wedge_{xnn}>\wedge_{xsw}) \vee (\wedge_{xnn}>\wedge_{xse}) \vee (\wedge_{xne}>\wedge_{xnw}) \vee (\wedge_{xne}>\wedge_{xss}) \vee (\wedge_{xse}>\wedge_{xsw}) \vee (\wedge_{xse}>\wedge_{xnn}) \vee (\wedge_{xss}>\wedge_{xnw}) \vee (\wedge_{xss}>\wedge_{xne}) \vee (\wedge_{xsw}>\wedge_{xse}) \vee (\wedge_{xsw}>\wedge_{xnn}) \vee (\wedge_{xnw}>\wedge_{xne}) \vee (\wedge_{xnw}>\wedge_{xss})_{\_s}$

An expression recognizing pretzel pictures over the alphabet$\Sigma = \{a, b, c, d\}$:

$$P = C[(||_{XEE} = ||_{XWW}) \vee (||_{XWW} = ||_{XEE}) \vee (||_{XNE} = ||_{XSW}) \vee (||_{XSW} = ||_{XNE}) \vee (||_{XSE} = ||_{XNW}) \vee (||_{XNW} = ||_{XSE})]_{\_}\ s\ C$$

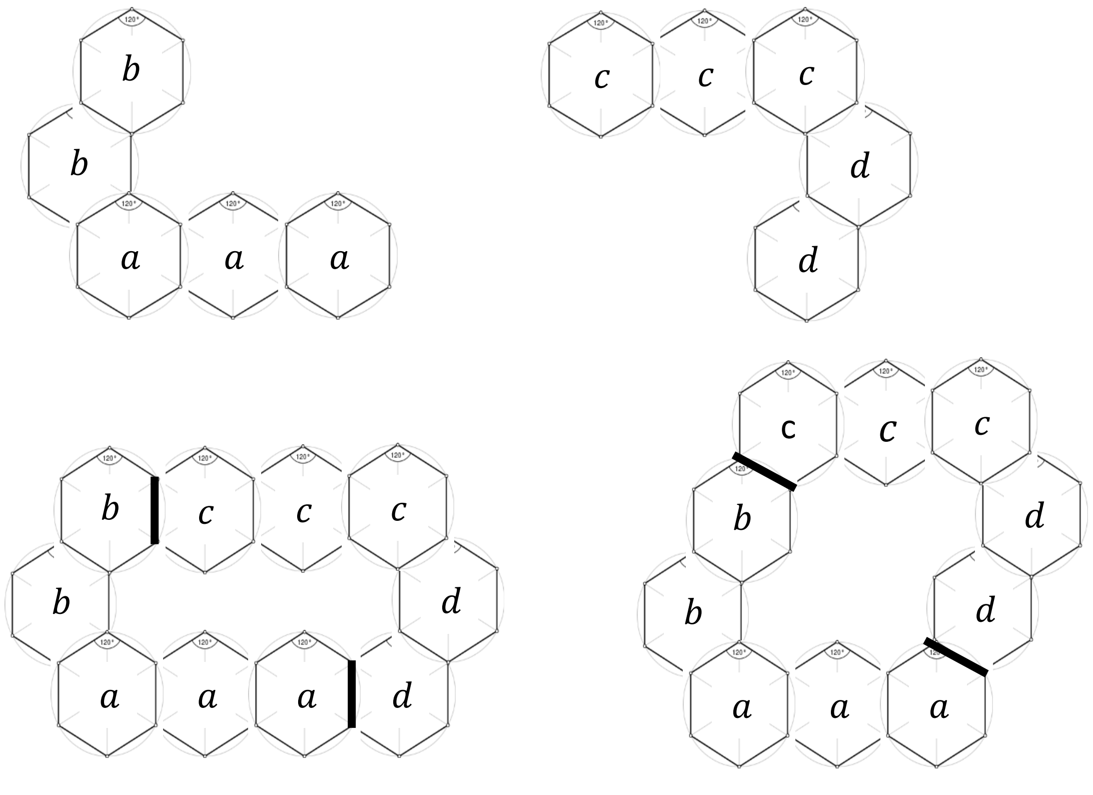


Figure 4.3 Hexagonal *Chains and pretzel-like shapes.*

Figure 4.3 exhibits the pretzel like shape by connecting the chains based on the above expressions.

As an application we have generated the pretzel shape floor designs as shown in Figure 4.4.

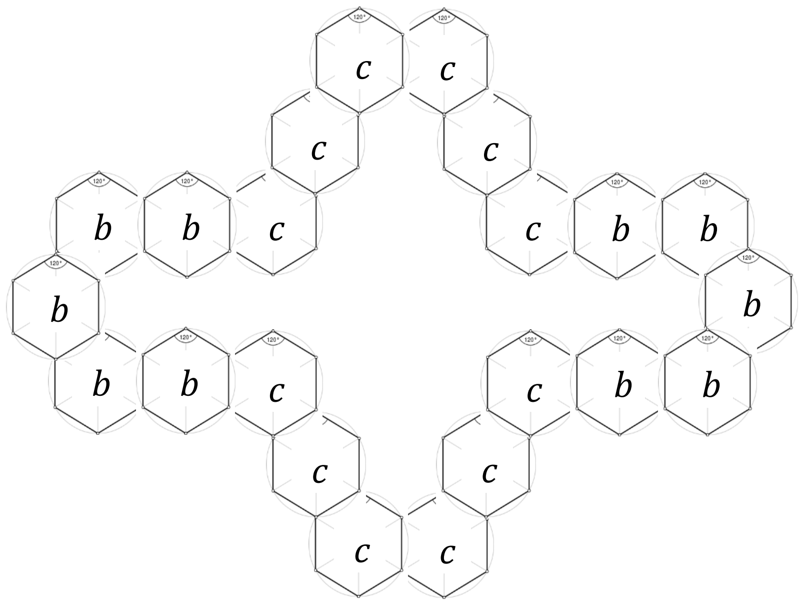


Figure 4.4. Pretzel shape floor designs.

CONCLUSION

In this paper, a new type of regular expressions $n2HRE$ for two-dimensional hexagonal patterns based on their composition is presented. We have initiated to prove some algebraic properties and generated mathematical models like Pascal triangle. As an application, shapes of floral designs are formed. In this approach, we are intended to find the magic way of getting the language by renaming and intersection is replaced by a steady work of tiling shapes to build up the pictures step by step. There are many directions to continue the research presented here. We are particularly interested to develop the interactive system scenario for this type of hexagonal expressions.

REFERENCES

1. Oliver Matz, Regular expressions and context free grammars for picture languages, LNCS 1200, 283-294, Springer, 2005.
2. I.T, Banu-Demergian, G. Stefanescu: The Geometric Membrane Structure of Finite Interactive Systems Scenarios, In Proceedings of the 14th International Conference on Membrane Computing, Chisinau, Moldova, 63-80, 2013.
3. I.T. Banu-Demergian, C.I. Paduraru, G. Stefanescu. A new representation of two-dimensional patterns and applications to interactive programming, In Proceedings of FSEN 2013, LNCS 8161, 183–198. Springer, 2013.
4. I.T. Banu-Demergian, G. Stefanescu. On the contour representation of two-dimensional patterns, arXiv preprint arXiv:1405.3791, 2014
5. D. Giammarresi , A. Restivo, S. Seibert, W. Thomas, Monadic second –order logic and recognizability by tiling systems. Information and Computation, 125, 32-45, 1996.
6. G. Stefanescu, Algebra of networks: Modelling simple network as well as complex interactive system. In: Proofand System- Reliability, 49-78, Springer, 2002
7. G. Stefanescu, Interactive systems: Fromfolkloreto Mathematics. In: Relmics' 01.LNCS 2561, 197-211, Springer, 2002
8. Wesley E. Snyder, Hairong Qi, William Sander, Coordinate systems for hexagonal pixels, In Proceedings. Spiedigitallibrary.org / data/ conferences/ SPIEP/…/716 _ 1 pdf.