# ON SOME COMPARATIVE RESULTS OF REGIONAL HEXAGONAL TILE REWRITING GRAMMARS

T.Kamaraj

Department of Mathematics,
Sathyabama University,
Chennai-119, India
kamarajmx@gmail.com

D.G.Thomas

Department of Mathematics,
Madras Christian College,
Chennai-59, India
dgthomasmcc@yahoo.com

**Abstract**

**Regional hexagonal Tile rewriting grammars(RHTRG) are the recently introduced hexagonal picture generating devices which used a simple type of tiling called regional hexagonal tiling. This model is having isometric rules of derivation and more general than context-free Hexagonal array grammars (HAG) but incomparable with hexagonal tiling systems (HTS). In this paper we compare RHTRG with some parallel generating formalisms like Extended pure 2D hexagonal context-free grammars with regular control and hexagonal array token Petri net Structure with respect to generating capacity..**

***Keywords*: Hexagonal arrays, Petri nets, Pure 2D grammars, Regional hexagonal tiling.**

## 1. Introduction

Hexagonal arrays generated by grammars are found in the literature with the insight of computations in pattern recognition, picture processing and scene analysis [3,5,8,9]. Some of classical formalisms to generate hexagonal arrays are Hexagonal Kolam Array Grammars (HKAG) [9] and Hexagonal Array Rewriting Grammars (HAG) [10], which used sequential and parallel application of rewriting rules. In HKAG model hexagonal arrays on triangular grid were viewed as two dimensional representation of three dimensional blocks and also several scene analytic transformations were discussed. Hexagonal Array Grammars (HAG) is a generalization for HKAG. Hexagonal Tile Rewriting (HTRG) [13] and Regional Hexagonal Tile rewriting grammars (RHTRG) [5] are the recent tiling based isometric families of hexagonal array rewriting grammars, which have more generative capacity than HAG and HKAG. Differently, Pure 2D Hexagonal context free array grammars (P2DHCFG) [10,11] is a simple yet expressive non isometric formalism , where parallel rewriting of arrow head occurs during the process of derivation. On the other hand, Hexagonal array token Petri nets (HATPNS) [6,7] are the evolution of Petri net structure from string generating one, which can generate the hexagonal picture languages of various classes of HAG and P2DHCFG.

Since the rectangular version of RHTRG model included many other array generating formalisms but still maintained the polynomial time complexity, it is natural to explore the further generating power of RHTRG. In this paper we compare RHTRG with an extended form of P2DHCFG and HATPNS and establish relationship results.

## 2. Preliminariess

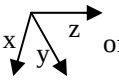
The following notations and definitions are mainly from [5,6,11].

Let Σ be a finite alphabet of symbols. A hexagonal picture p over is a hexagonal array of symbols of Σ.

The set of all hexagonal arrays over of the alphabet Σ is denoted by $\Sigma^{**H}$ and set of all non empty hexagonal arrays over Σ is denoted by $\Sigma^{++H}$. A non empty hexagonal picture language L over Σ is a subset of $\Sigma^{++H}$.

For $p \in \Sigma^{++H}$, let $\hat{p}$ be the hexagonal array obtained by surrounding p with a special boundary symbol $\# \notin \Sigma$. for example,

$$\hat{p} = \begin{array}{ccccccc} & \# & & \# & & \# & \\ \# & & a & & a & & \# \\ \# & & a & & a & & \# \\ & \# & & a & & \# & \\ & & \# & & \# & & \end{array} \quad \text{for} \quad p = \begin{array}{ccc} a & & a \\ & a & \\ a & & a \end{array}.$$

With respect to a triad 

of triangular axes x, y, z, the co-ordinates of each element of hexagonal picture can be fixed [1].

Given a picture $p \in \Sigma^{++H}$, let '$\ell$' denote the number of elements in the border of p from upper left vertex to left most vertex in the direction ↙ called x direction, 'm' denote the number of elements in the border of p from upper right vertex to right most vertex in the direction ↘ called y direction and 'n' denote the number of elements in the border of p from upper left vertex to upper right vertex in the direction → called z direction.

The directions are fixed with origin of reference as the upper left vertex, having co-ordinates (1, 1, 1). The triple ($\ell$, m, n) is called the size of the picture p denoted by $|p| = (\ell, m, n)$. Let $p_{ijk}$ denote the symbol in p, called as pixel, with coordinates (i, j, k) where $1 \le i \le \ell$, $1 \le j \le m$, $1 \le k \le n$. If all pixels are identical to a pixel C then the hexagonal picture is called C-homogenous. The domain of a picture p is the set d(p) consists of positions of all pixels in p. A subdomain $d_s(p)$ of d(p) is a set of positions of pixels correspond to a subpicture s of p. A subdomain is called C-homogenous when its associated sub picture is a C-homogenous. Let $T^{(\ell, m, n)H}$ be the set of hexagonal pictures of size ($\ell$, m, n).

Given a hexagonal picture p of size ($\ell$, m, n), we denote by $B_{g,h,k}(p)$ the set of all hexagonal sub pictures of p of size (g, h, k), where $g \le \ell$, $h \le m$ and $k \le n$. Each member of $B_{2,2,2}(p)$ is called a hexagonal tile. We denote the set of all hexagonal tiles contained in a picture $\hat{p}$ by $[[\hat{p}]]$.

An arrowhead is a concave hexagon which when catenated to a hexagon in one of the six possible directions, results in growth of the hexagon in that direction. We refer [1, 6] for basic notions and definitions of Arrowheads and arrowhead catenations. Fig.1 shows a left arrowhead and Fig.2(b) shows the left arrowhead catenation of the hexagon in Fig.2(a)

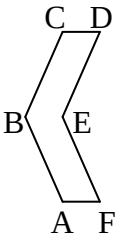



Fig.1. Left arrowhead

.

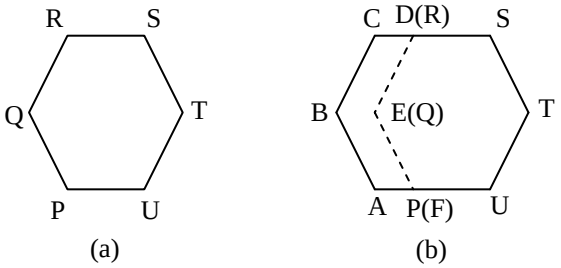



Fig.2. Left arrowhead catenated to hexagon

**Definition 1.** Let Γ be a finite alphabet. A hexagonal picture language $L \subseteq \Gamma^{**H}$ is called local if there exists a finite set Δ of hexagonal tiles over $\Gamma \cup \{\#\}$ such that $L = \{p \in \Gamma^{**H} / B_{2,2,2}(\hat{p}) \subseteq \Delta\}$. The family of hexagonal local picture languages will be denoted by L(HLOC).

We now recall the notions of extended class of Pure 2D Hexagonal context free grammars and Hexagonal array token Petri net structure.

**Definition 2.** A pure 2D hexagonal context-free grammar ( P2DHCFG) [ 11] is a construct

$G = ( T, P_{ur}, P_{ul}, P_{lr}, P_{ll}, P_l, P_r, M_0 )$, where

T is a finite set of symbols,

$P_{ur} = \{ t_{ur_i} / 1 \le i \le m \}$; Each $t_{ur_i}$, $(1 \le i \le m)$ , called a UR table, is a set of context-free rules of the form $a \to \alpha$, $a \in T$, $\alpha \in T^*$ such that any two rules of the form $a \to \alpha$, $b \to \beta$ in $t_{ur_i}$ , we have $|\alpha| = |\beta|$ where $|\alpha|$ denotes the length of $\alpha$,

each of the other five components $P_{ul}$, $P_{lr}$, $P_{ll}$, $P_l$ and $P_r$ is similarly defined,

$M_0 \subseteq T^{++H}$ is a finite set of hexagonal arrays or arrow heads, named axioms.

Derivations are defined as follows:

For any two hexagonal arrays H1,H2, we write $H1 \Rightarrow H2$ if H2 obtained from H1 by rewriting all the symbols in an " Arrowhead of thickness one " of H1 by rules of a relevant table in $P_{ur} \cup P_{ul} \cup P_{lr} \cup P_{ll} \cup P_l \cup P_r$. $\overset{*}{\Rightarrow}$ is the reflexive transitive closure of $\Rightarrow$. The hexagonal picture language L(G) generated by G is the set $\{H / H_0 \overset{*}{\Rightarrow} H \in T^{**H}$, for some $H_0 \in M_0\}$

**Definition 3.** A pure 2D hexagonal context-free grammar with regular control (RP2DHCFG) is a tuple

$G_r = (G, \Gamma, C)$ where

1. G is a P2DHCFG

2. Γ is the control alphabet, the set of labels of the rule tables in $P_{ur} \cup P_{ul} \cup P_{lr} \cup P_{ll} \cup P_l \cup P_r$

3. $C \subseteq \Gamma^*$ is the regular control associated with the $G_r$.

If $H \in T^{**H}$ and $H_0 \in M_0$, H is derived from $H_0$ in $G_r$ by means of a control word $w = w_1 w_2 ... \in C$, in symbols $C \Rightarrow_w H$ , if H is obtained from $H_0$ by applying the table rules as in the sequence of tables $w_1 w_2 ...$ . The language L(G) generated by RP2DHCFG $G_r$ is the set of pictures $\{H / H_0 \Rightarrow_w H \in T^{++H}$ for some $w \in C\}$.

To increase the generativity of RP2DHCFG further, we refine the definition of this class of grammars like that of [12] but adapted as in [2].

**Definition 4.** The extended pure 2D hexagonal context-free grammars with regular control (ERP2DHCFG) are RP2DHCFG with $T = T_f \cup T_c$ where $T_f$ is the alphabet of final symbols defining the pictures and $T_c$ is a set of control symbols which are involved only in the process of derivation and they do not appear in the final hexagonal picture.

**Definition 5.** Petri Net is one of the mathematical modeling tools for the description of distributed systems involving concurrency and synchronization. It is a weighted directed bipartite graph consisting of two kinds of nodes called places (represented by circles) and transitions (represented by bars). The places from which a directed arc runs to a transition are called input places of the transition and the places to which directed arcs run from a transition are called output places. Places in Petri nets may contain a discrete number of marks called tokens. In the abstract sense, a transition of a Petri net may fire if it is enabled; when there are sufficient tokens in all of its input places.

**Definition 6.** A Petri net structure is a four tuple $C = <Q, T, I, O>$ where $Q = \{q_1, q_2, ..., q_n\}$ is a finite set of places, $n \ge 1$, $T = \{t_1, t_2, ..., t_m\}$ is a finite set of transitions $m \ge 1$, $Q \cap T = \varnothing$, $I : T \to Q^\infty$ is the input function from transitions to bags of places and $O : T \to Q^\infty$ is the output function from transitions to bags of places.

**Definition 7.** An inhibitor arc from a place $q_l$ to a transition $t_k$ has a small circle in the place of an arrow in regular arcs. This means the transition $t_k$ is enabled only if $q_l$ has no tokens in it. In other words a transition is

enabled only if all its regular arc input places have required number of tokens and all its inhibitor arc (if exists) input places have zero tokens.

In the hexagonal array generating Petri Net structure, hexagonal arrays over an alphabet J are used as tokens in some input places.

**Definition 8.** An Hexagonal Array Token Petri net structure (HATPNS) is a five tuple N = <J, C, $M_0$, ρ, F> where J is a given alphabet, C = <Q, T, I, O> is a Petri net structure with tokens as hexagonal arrays over J, $M_0$ : $Q \rightarrow J^{**}$, is the initial marking of the net, $\rho : T \rightarrow L$, a mapping from the set of transitions to set of labels of transitions and $F \subset Q$, is a finite set of final places.

### 3. Regional hexagonal tile rewriting grammars

In this section we recall the formal definition of Regional Hexagonal Tile Rewriting Grammars [5 ] and give some simple examples on languages generated by these grammars.

**Definition 9.** A homogeneous partition of a picture p is any partition $P = \{d_{s_1}, d_{s_2}, ..., d_{s_n}\}$ of d(p) into homogeneous sub domains $d_{s_1}, d_{s_2}, ..., d_{s_n}$ The unit partition of p, written unit(p), is the homogeneous partition of d(p) defined by single pixels. A homogeneous partition is called strong if adjacent sub domains have different labels. The unique partition given by strong homogenous partition of a picture p is $\pi^*(p)$.

**Definition 10.** A homogeneous partition is regional (RH) iff distinct sub domains have distinct labels. A picture p is regional if it admits a RH partition. A language is regional if all its pictures are regional.

**Definition 11.** A Regional hexagonal tile rewriting grammar (RHTRG) is a tuple (Σ, N, S, R), where Σ is the terminal alphabet, N is a set of non terminal symbols, S ∈ N is the starting symbol, R is a set of rules of the form

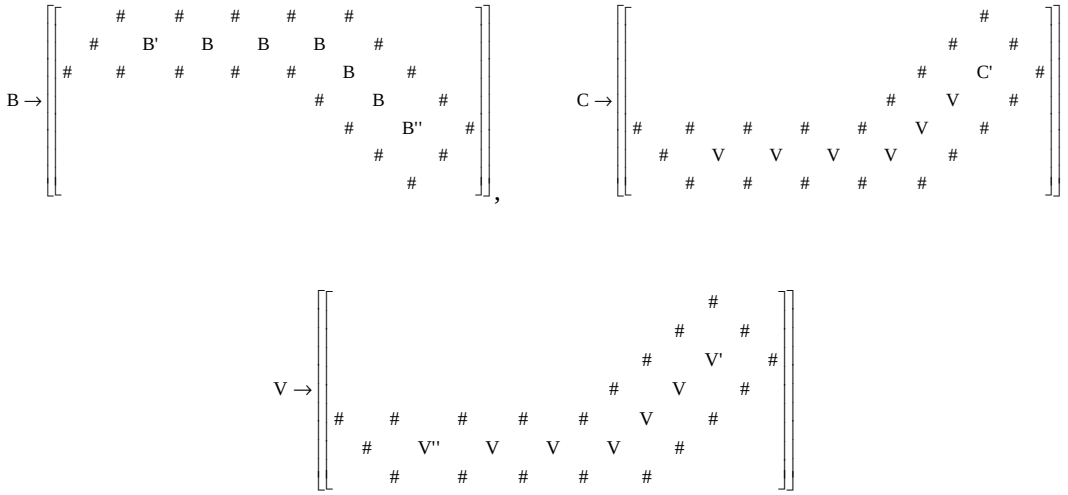
Fixed size : $A \rightarrow t$, where $t \in \Sigma$.

Variable size : $A \rightarrow w$, where w is a set of hexagonal tiles over $N \cup \{\#\}$, HLOC(w) is a regional language.

**Example 1.** The language of hexagonal pictures over {a} with equal sides is generated by the RHTRG G = < Σ, N, S, R > where Σ= {a}, N = {S,A,B,C,A',A'',B',B'',C',U,V,U',V',U'',V'',$X_1$,$X_2$,$X_3$,$X_4$,$X_5$,$X_6$,$X_7$} and R consists of the following rules

$$S \rightarrow \begin{bmatrix} & & \# & \# & \# & \# & \# & \\ & \# & A & B & B & B & \# & \\ \# & A & S & S & S & S & B & \# \\ \# & A & S & S & S & S & S & C & \# \\ \# & A & S & S & S & S & S & C & \# \\ & \# & A & S & S & S & C & \# & \\ & & \# & A & C & C & C & \# & \\ & & & \# & \# & \# & \# & \# & \end{bmatrix} / \begin{bmatrix} & \# & & \# & & \# & \\ \# & & X_1 & & X_2 & & \# \\ & \# & & X_6 & & X_7 & & X_3 & \# \\ \# & & X_5 & & X_4 & & \# \\ & \# & & \# & & \# & \end{bmatrix}$$

$$A \rightarrow \begin{bmatrix} & & \# & \# & \# & \\ & \# & A' & & \# & \\ \# & U & & & \# & \\ \# & U & & & \# & \\ \# & U & & & \# & \\ \# & U & & & \# & \\ & \# & A'' & & \# & \\ & & \# & \# & \# & \end{bmatrix}, \quad U \rightarrow \begin{bmatrix} & & \# & \# & \# & \\ & \# & U' & & \# & \\ \# & U & & & \# & \\ \# & U & & & \# & \\ \# & U & & & \# & \\ \# & U & & & \# & \\ & \# & U'' & & \# & \\ & & \# & \# & \# & \end{bmatrix}$$

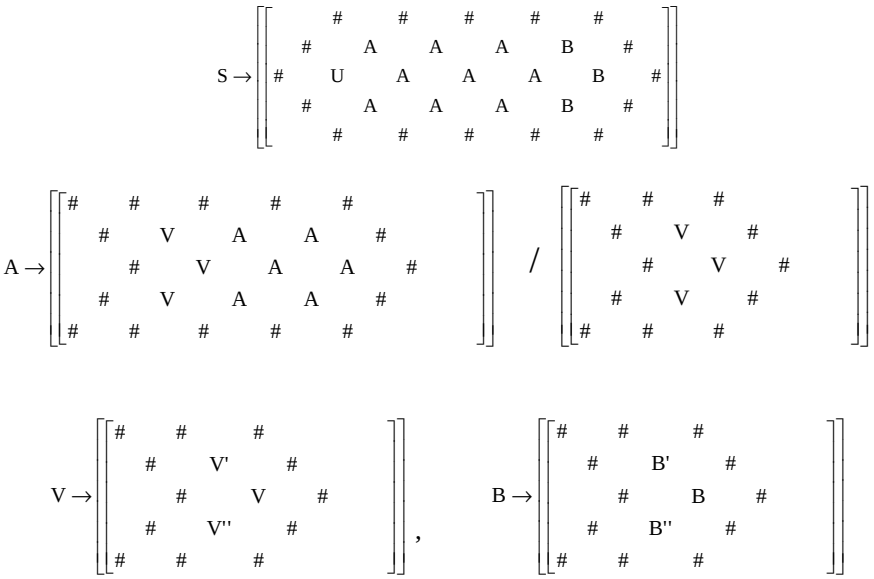


A,B,C,A’,A’’,B’,B’’,C’,U,V,U’,V’,U’’,V’’,$X_1$,$X_2$,$X_3$,$X_4$,$X_5$,$X_6$,$X_7$ $\rightarrow$ a

**Example 2.** The language

$$L_2=\left\{ \begin{matrix} x & x & & x & x & x & & x & x & x & x & \\ x & y & x, & x & y & y & x, & x & y & y & y & x, \ldots \\ x & x & & x & x & x & & x & x & x & x & \end{matrix} \right\}$$

can be generated by the RHTRG G = < Σ, N, S, R > where Σ= {a}, N = {S,V,A,B,V’,V’’,B’,B’’,U} and R consists of



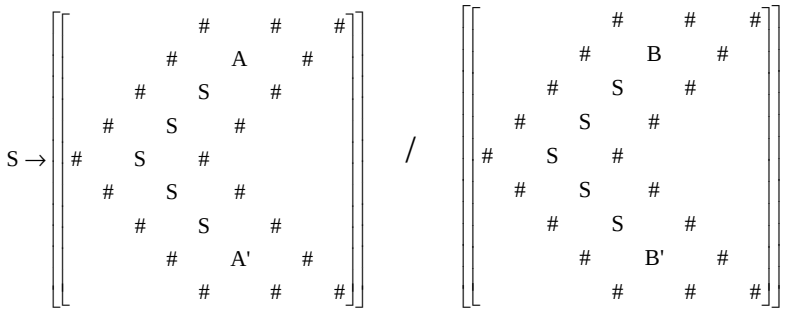
V $\rightarrow$ y, U,V’,V’’,B’,B’’ $\rightarrow$ x

**Example 3.** Consider the language $L_3$, which consists of palindromic left arrowheads of thickness one over T = {a, b}

A typical element of $L_3$ is

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

$L_3$ can be generated by the RHTRG G = < Σ, N, S, R > where Σ= {a,b}, N ={S,A,A',B,B' } and R consists of




S → a,b,
A,A' → a and B,B' → b

## 4. Comparative Results

In this section, we present comparison results of RHTRG with other hexagonal models with respect to its generative power. We use $\mathcal{L}$(X) to denote family of languages generated by the device X

**Theorem 1.** $\mathcal{L}$(RHTRG) and $\mathcal{L}$(ERP2DHCFG) are incomparable but not disjoint

**Proof.** The language $L_3$ in Example 3 can be generated by a ERP2DHCF grammar G = < $T_f \cup T_c$ , {$r_1$,$r_2$},{λ},{λ},{ λ },{ λ }, { λ }, $M_0$ > and $T_f$ = { x, y}, $T_c$ = { d, e }, $M_0$ = $\begin{Bmatrix} & d & & x & \\ x & & e & & x \\ & d & & x & \end{Bmatrix}$ and $r_1$ = { d → *x* d, e → ye } , $r_2$ = { d→x, e→y} with regular control $(r_1)^* r_2$ .

So, $\mathcal{L}$(RHTRG) ∩ $\mathcal{L}$(EP2DHCFG) ≠ ∅.

For incomparability : Consider the language $L_4$ consisting of all hexagonal pictures of sizes (2,2, 2n+1 ) n ≥ 1 , with alternate a-homogenous and b-homogenous symmetrical arrowheads. A typical picture in the language is

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

This language can be generated by a ERP2DHCFG, G = ( T, $P_{ur}$, $P_{ul}$, $P_{lr}$, $P_{ll}$, $P_l$, $P_r$, $\mathcal{M}_0$ ), where T = $T_f \cup T_c$ ; $T_f$ = {a,b}, $T_c$={$v_1$,$v_2$},

$P_r$={$r_1$, $r_2$}; $r_1$={ $v_2$→a$v_1$, $v_1$→b$v_2$ }, $r_2$={ $v_2$→a, $v_1$→b },
$P_l$={$l_1$, $l_2$}; $l_1$={ $v_1$← $v_2$a, $v_2$ ← $v_1$b }, $l_2$={$v_1$← a, $v_2$← b}

$P_{ur}$ = $P_{ul}$ = $P_{lr}$= $P_{ll}$ = ∅ and $\mathcal{M}_0$ = $\begin{Bmatrix} & v_1 & & v_2 & \\ v_1 & & b & & v_2 \\ & v_1 & & v_2 & \end{Bmatrix}$. The regular control language associated with G is $(l_1 r_1)^*$ $l_2 r_2$. But this language cannot be generated by any RHTRG as to form the pictures of $L_4$ we should define RHTRG with a variable size rule as given below

$$S \rightarrow \begin{bmatrix} \begin{matrix} & \# & & \# & & \# & & \# & & \# & \\ \# & & V & & V & & W & & W & & \# \end{matrix} \\ \begin{matrix} \# & & V & & V & & B & & W & & W & & \# \end{matrix} \\ \begin{matrix} & \# & & V & & V & & W & & W & & \# & \\ & & \# & & \# & & \# & & \# & & \# & & \end{matrix} \end{bmatrix}$$

But the regional characterization of partition of pictures generated by an RHTRG, cannot provide the semantic relation between the two distinct blocks.

On the other hand, the language $L_3$ of palindromic left arrow heads in example 3, cannot be generated by any ERP2DHCFG since from the definition of ERP2DHCFG grammar, the effect of applying a table rules guided by a control word to a arrowhead of thickness one is either increasing the thickness or maintaining the same thickness but not on the length of the arrowhead. □

**Theorem 2.** $\mathcal{L}$(RHTRG) and $\mathcal{L}$(HATPNS) are incomparable but not disjoint

**Proof.** The language in $L_1$ in example 1 can also be generated by an HATPNS, $P_1 = < J, C, M_0, \rho, F >$ given by Petri net graph in Fig.3.

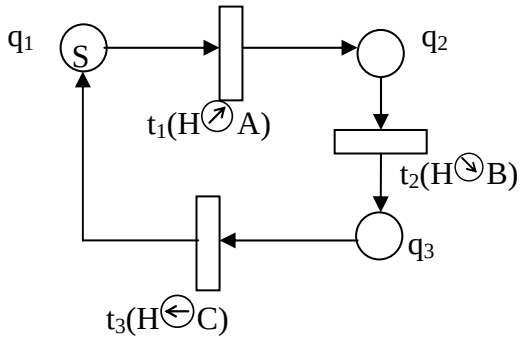



Fig.3 Petri net for generating $L_1$

Where the arrays used are S = $\begin{matrix} & a & & a & \\ a & & a & & a \\ & a & & a & \end{matrix}$ and A = $(a)^{|H|_z-1}\langle a\rangle(a)^{|H|_y-1}$, B = $(a)^{|H|_x-1}\langle a\rangle(a)^{|H|_z-1}$ and C = $(a)^{|H|_y-1}\langle a\rangle(a)^{|H|_x-1}$ and F = $\{q_1\}$.

For incomparability : the language $L_2$ in example 2 cannot be generated by any HATPNS as it is the consequence of the fact that the growth of an array in any HATPNS language can take place only outside the border arrow heads but not in the interior as in $L_2$ .

On the other hand consider the language $L_5$ of regular hexagons over {a} of side $2^n$ which can be generated by an HATPNS [ Example 4.1, 7]. This language cannot be generated by any RHTRG as to build a hexagon of size $2^n$ , we should have the catenation of the upper right, lower right and left arrow head blocks of thickness $2^{n-1}$ with a central hexagon of size $2^{n-1}$ . But a regional hexagonal partition cannot provide a size dependency between the arrowhead blocks and central hexagon block. □

## 5. Conclusion

In this work we have considered regional hexagonal tiling based grammar model recently introduced by Kamaraj et al., for comparing with some of the expressive parallel rewriting hexagonal array generating models. The lack of dependency between the regional hexagonal portioned blocks by production rules make the generative capacity of this model incomparable with that of ERP2DHCFG and HATPNS. But in future such models with control can be developed to have higher generative capacity. Practical applicability of this model to picture processing tasks like pattern recognition , image processing tasks remains to be explored.

## References

[1] Baker H.G., (1972): Petri net languages, *Computation Structures Group Memo 68*, Project MAC, MIT, Cambridge, Massachusetts.
[2] Bersani M.M., Frigeri A. and Cherubini A., (2011): On some classes of 2D languages and their relations. In: *IWCIA* 2011, LNCS, Vol. 6636, Springer, Heidelberg, 222–234
[3] Dersanambika K.S., *et al.*, (2005): Local and recognizable hexagonal picture languages, International Journal of Pattern Recognition and Artificial Intelligence, 19, 853–871.
[4] Hack M., Petri net languages, (1975): *Computation Structures Group Memo 124*, Project MAC, MIT.

[5] Kamaraj T. and Thomas D.G., (2012): Regional hexagonal tile rewriting grammars, In : R.P. Barneva et al (eds.), IWCIA 2012, LNCS, Vol. 7655, Springer, Heidelberg, 181−195.
[6] Lalitha D., Rangarajan K. and Thomas D.G., (2011): Petri net generating hexagonal arrays, In : J.K. Aggarwal et al (eds.), *IWCIA* 2011, LNCS, Vol. 6636, Springer, Heidelberg, 235−247
[7] Lalitha D., Rangarajan K., and Thomas D.G., (2011): Petri net generating context-sensitive hexagonal array languages, In : R. Nadarajan et al (eds.), ICMCM 2011, Narosa Publications, New Delhi, 389−397
[8] Middleton L. and Sivaswamy J., Hexagonal image processing , 2005: A practical approach, Advances in Computer Vision and Pattern Recognition Series, Springer.
[9] Siromoney G. and Siromomey R, (1976): Hexagonal arrays and rectangular blocks, Computer Graphics and Image Processing, 5, 353–381.
[10] Subramanian, K.G. Hexagonal array grammars, (1979): Computer Graphics and Image Processing, 10, 388–394.
[11] Subramanian K.G., *et al*, (2009): Pure 2D picture grammars and languages, Discrete Applied Mathematics, 157(16), 3401−3411.
[12] Subramanian K.G., *et al*, (2008): Two-dimensional picture grammar models, In: Proceedings of the 2nd European Modelling Symposium, EMS2008, IEEE, pp. 263–267.
[13] Thomas D.G., Sweety F. and Kalyani T.,(2008): Results on Hexagonal Tile Rewriting Grammars, In: G. Bebis et al. (Eds.), International Symposium on Visual Computing, Part II, LNCS, 5359, Springer-Verlag, Berlin, Heidelberg, 945–952.