Hexagonal Prusa Grammar Model for Context-Free Hexagonal Picture Languages

T. Kamaraj and D. G. Thomas

Abstract Prusa Grammar is a recently introduced rectangular picture languages generating model which exploits the parallel application of two-dimensional context-free rules. We introduce the hexagonal version of Prusa grammar and generate images. We compare this model with other hexagonal array generating devices for the description of its generative power.

Keywords Hexagonal array languages · Hexagonal tiling system · Regional hexagonal tile rewriting grammars · Prusa grammars

1 Introduction

Hexagonal arrays generated by grammars are found in the literature with the insight into computations in pattern recognition, picture processing, and scene analysis [4, 6]. Some of classical formalisms to generate hexagonal arrays are hexagonal kolam array grammars (HKAG) [6] and hexagonal array rewriting grammars (HAG) [7]. 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 rewriting grammars (HAG) are the generalization for HKAG. Hexagonal Wang system (HWS) and hexagonal tiling system (HTS) were an equivalent pair of formal devices for recognizable class of hexagonal arrays (HREC) [1]. Hexagonal tile

T. Kamaraj (✉)
Department of Mathematics, Sathyabama University, Chennai, Tamil Nadu 600119, India
e-mail: kamaraj_mx@yahoo.co.in

D. G. Thomas
Department of Mathematics, Madras Christian College, Chennai, Tamil Nadu 600059, India
e-mail: dgthomasmcc@yahoo.com

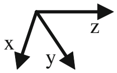
G. S. S. Krishnan et al. (eds.), *Computational Intelligence, Cyber Security and Computational Models*, Advances in Intelligent Systems and Computing 246,
DOI: 10.1007/978-81-322-1680-3_33, © Springer India 2014

rewriting grammars (HTRG) [8] and regional hexagonal tile rewriting grammars (RHTRG) [2] are the recent tiling-based hexagonal array rewriting models, which have more generative capacity. In the generalization process of rectangular grammar models, Prusa [5] has recently defined a grammar device which extended the rectangular context-free kolam array grammar model, attaining some generative capacity. This model generates the rectangular pictures by parallel application of rules and thereby maintains a grid-like structure in each stage of derivation of picture. But hexagonal grammar models with parallel derivation rules are very rare in literature and that too have limited growth patterns. With this quest of generalization of hexagonal models, we propose a hexagonal version of Prusa grammar to generate context-free hexagonal picture languages and study some comparisons with the other existing models for their generative capacity.

The paper is organized in the following manner. In Sect. 2, we recall some basic notions of hexagonal pictures and languages. In Sect. 3, we introduce hexagonal Prusa grammar (HPG) and examples for illustration. In Sect. 4, we present comparison results of HPG with other hexagonal models with respect to the generating capacity.

2 Preliminaries

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

With respect to a triad 

of triangular axes x, y, z, the coordinates of each element of hexagonal picture can be fixed [1]. The origin of reference is the upper left vertex, having co-ordinates (1, 1, 1).

The set of all hexagonal arrays over of the alphabet T is denoted by T^{**H} and set of all non-empty hexagonal arrays over T is denoted by T^{++H}. T^{+} denotes set of all non-empty strings in the three directions parallel to the triangular axes. A non-empty hexagonal picture language L over T is a subset of T^{++H}.

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

Given a picture $p \in T^{++H}$, if ℓ, m, n denote the number of elements in the borders of p, parallel to x-, y-, z-directions, respectively, then the triple (ℓ, m, n) is called the size of the picture p denoted by $|p| = (\ell, m, n)$. Let p_{ijk} denotes the symbol in p with coordinates (i, j, k) where $1 \leq i \leq \ell$, $1 \leq j \leq m$, $1 \leq k \leq n$. Let $T^{(\ell,\ m,\ n)H}$ be the set of hexagonal pictures of size (ℓ, m, n). A typical hexagonal array of size (ℓ, m, n) can be denoted by $[p_{ijk}]^{(\ell,m,n)H}$. A hexagonal picture of size (2, 2, 2) is called a hexagonal tile. We denote the set of all hexagonal tiles contained in a picture $\hat{p}$ by $[[\hat{p}]]$.

The notions of non-convex hexagonal arrays called arrowheads, and arrowhead catenations in six directions are adapted as in [1, 6].

3 Hexagonal Prusa Grammar

In this section, we give a formal definition of hexagonal Prusa grammar and some simple examples of languages generated by these grammars.

Definition 1 A Hexagonal Prusa Grammar (HPG) is a tuple $\langle N, T, P, S\rangle$ where N is a finite set of non-terminals, T is a finite set of terminals, $P \subseteq N \times [(N \cup T)^{++H} \cup (N \cup T)^{+}]$ and $S \in N$ is the start symbol.

Definition 2 Let $G = \langle N, T, P, S\rangle$ be a HPG. We define a hexagonal picture language $L(G, C)$ over T for every $C \in N$, by the following recursive rules.

1. Terminal rule: If $C \to X$ is in P, and $X \in (T^{++H} \cup T^{+})$, then $X \in L(G, C)$.
2. Mixed rule: Let $C \to X$ be a production in P with

$$X \in \cup\ (N \cup T)^{(\ell', m', n')H},\ 1 \leq \ell' \leq \ell,\ 1 \leq m' \leq m \text{ and } 1 \leq n' \leq n$$

and Q_{ijk} $(1 \leq i \leq \ell,\ 1 \leq j \leq m,\ 1 \leq k \leq n)$ be the pictures such that

(a) if $X_{ijk} \in T,$ then $Q_{ijk} = X_{ijk}$.
(b) if $X_{ijk} \in N,$ then $Q_{ijk} \in L[G, X]$.

Then, if $Q = [Q_{ijk}]^{(\ell', m', n')H}$ is defined through string catenation (or) arrowhead catenation, then $Q \in L[G, C]$.

The set $L[G, C]$ contains all and only pictures that can be obtained by applying a finite sequence of rules (1) and (2). The hexagonal language $L[G]$ generated by the grammar G is defined to be the language $L[G, S]$. $\mathcal{L}$(HPG) is the class of all languages generated by these grammars. Languages in $\mathcal{L}$(HPG) are called HPG languages.

Example 1 The language

$$L_1 = \left\{ \begin{array}{ccccc} & a & & a & \\ a & & a & & a \\ & a & & a & \end{array},\ \begin{array}{ccccccc} & a & & a & & a & \\ a & & a & & a & & a \\ & a & & a & & a & \end{array},\ \begin{array}{ccccccccc} & a & & a & & a & & a & \\ a & & a & & a & & a & & a \\ & a & & a & & a & & a & \end{array}, \ldots \right\}$$

is generated by HPG $G_1 = \langle N, T, P, S\rangle$ where $N = \{H, S\}$, $T = \{a\}$ and $P = \left\{ S \to SH,\quad S \to \begin{array}{ccccc} & a & & a & \\ a & & a & & a \\ & a & & a & \end{array},\quad H \to \begin{array}{cc} a & \\ & a \\ a & \end{array} \right\}$. By applying terminal rules, $H \to \begin{array}{cc} a & \\ & a \\ a & \end{array}$, $S \to \begin{array}{ccccc} & a & & a & \\ a & & a & & a \\ & a & & a & \end{array}$ we get $\begin{array}{cc} a & \\ & a \\ a & \end{array} \in L[G, H]$ and $\begin{array}{ccccc} & a & & a & \\ a & & a & & a \\ & a & & a & \end{array} \in L[G, S]$.

Now, by applying mixed rule $S \to SH$, we have $\begin{array}{ccccccc} & a & & a & & a & \\ a & & a & & a & & a \\ & a & & a & & a & \end{array} \in L[G, S]$.

The repeated application of mixed rule $S \to SH$ generates all the members of L_1.

Example 2 The language

$$L_2 = \left\{ \begin{array}{ccccc} & 1 & & 1 & \\ 1 & & 2 & & 1 \\ & 1 & & 1 & \end{array}, \begin{array}{ccccccc} & 1 & & 1 & & 1 & \\ 1 & & 2 & & 2 & & 1 \\ & 1 & & 1 & & 1 & \end{array}, \begin{array}{ccccccccc} & 1 & & 1 & & 1 & & 1 & \\ 1 & & 2 & & 2 & & 2 & & 1 \\ & 1 & & 1 & & 1 & & 1 & \end{array}, \ldots \right\}$$

can be generated by HPG, $G_2 = \langle N, T, P, S \rangle$ where $N = \{S, A, B, H\}$, and $T = \{1, 2\}$.

$$P = \left\{ S \to AH \;/\; \begin{array}{ccccc} & 1 & & 1 & \\ 1 & & 2 & & 1 \\ & 1 & & 1 & \end{array}, \quad A \to AB \;/\; \begin{array}{ccccc} & 1 & & 1 & \\ 1 & & 2 & & 2 \\ & 1 & & 1 & \end{array}, \quad B \to \begin{array}{cc} 1 & \\ & 2 \\ 1 & \end{array}, \quad H \to \begin{array}{cc} 1 & \\ & 1 \\ 1 & \end{array} \right\}$$

From the terminal rules, we have $\begin{array}{ccccc} & 1 & & 1 & \\ 1 & & 2 & & 1 \\ & 1 & & 1 & \end{array} \in L[G,S]$, $\begin{array}{cc} 1 & \\ & 2 \\ 1 & \end{array} \in L[G, B]$.
Parallel application of mixed rules $A \to AB$, $S \to AH$ repeatedly produces the language L_2.

We now introduce non-terminal normal form for hexagonal Prusa grammars.

Definition 3 A hexagonal Prusa grammar $G = \langle N, T, P, S \rangle$ is in non-terminal normal form (NNF) iff every rule in P has the form either $C \to t$ or $C \to X$, where $C \in N$, $X \in (N^{++H} \cup N^{+})$ and $t \in T$.

4 Comparison Results

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

Theorem 1

$$\mathcal{L}[\text{CFHAG}] \subset \mathcal{L}[\text{HPG}].$$

Proof Consider a context-free hexagonal array grammars (CFHAG) [7] G in Chomsky normal form. It may contain six types of rules:

$C \to A \textcircled{$\leftarrow$} B$, $C \to A \textcircled{$\nearrow$} B$, $C \to A \textcircled{$\searrow$} B$ and its duals.

$C \to A \textcircled{$\rightarrow$} B$, $C \to A \textcircled{$\swarrow$} B$ and $C \to A \textcircled{$\nwarrow$} B$.

Now, the rule $C \to A \textcircled{$\leftarrow$} B$ is equivalent to a HPG rule of the form $C \to$ BA.

The rule $C \to A \; \textcircled{\scriptsize$\nearrow$} \; B$ is equivalent to a HPG rule of the form $C \longrightarrow A^B$.

The rule $C \to A \; \textcircled{\scriptsize$\searrow$} \; B$ is equivalent to a HPG rule of the form $C \longrightarrow A_B$.

Similarly, we can have equivalent rules in HPG for the corresponding dual rules in CFHAG. The terminal rules $C \to t$ are identical in both the grammars. This shows that $\mathcal{L}[\mathrm{CFHAG}] \subseteq \mathcal{L}[\mathrm{HPG}]$.

The inclusion is strict as the language L_2 in Example 2 cannot be generated by any CFHAG, which was proved in [2].

Theorem 2 *$\mathcal{L}[\mathrm{HPG}]$ and $\mathcal{L}[\mathrm{HTS}]$ are incomparable but not disjoint.*

Proof It is already proved in [2] that the language L_1 in Example 1 is a hexagonal tiling system [1] recognizable language. So $L_1 \in \mathcal{L}(\mathrm{HPG}) \cap \mathcal{L}(\mathrm{HTS})$.

Now, consider the language L_4, which consists of palindromic left arrowheads over $T = \{a, b\}$.

L_4 can be generated by the HPG, $G = \langle N, T, P, S \rangle$ with $N = \{S\}$, $T = \{a, b\}$,

$$P = \left\{ S \longrightarrow \begin{array}{cc} & a \\ S & \\ & a \end{array}, S \longrightarrow \begin{array}{cc} & b \\ S & \\ & b \end{array}, S \longrightarrow \begin{array}{cc} & a \\ a & \\ & a \end{array} \Big/ \begin{array}{cc} & a \\ b & \\ & a \end{array} \Big/ \begin{array}{cc} & b \\ a & \\ & b \end{array} \Big/ \begin{array}{cc} & b \\ b & \\ & b \end{array} \right\}$$

But there is no local strategy available to remember the characters that appearing in the corresponding positions above and below the arrowhead vertex. So L_4 cannot be generated by any HTS.

In [3], the authors have given a hexagonal tiling system to generate a language $L_{\mathrm{parallel}} \swarrow$ which consists of all hexagonal pictures over $\Sigma = \{a, b\}$ with elements in the corresponding positions of the borders parallel to x-direction are identical. Since the productions of a HPG do not have control in producing such patterns, $L_{\mathrm{parallel}} \swarrow$ cannot be generated by any HPG.

Theorem 3

$$\mathcal{L}(\mathrm{HPG}) \subset \mathcal{L}[\mathrm{RHTRG}].$$

Proof Consider a HPG grammar G in NNF. First, without loss of generality, we assume that the non-terminals in the right part of any rule of G are all different.

Let us define a RHTRG [2] G' equivalent to G. The terminal rules are easy to handle. For a non-terminal rule of G, $C \longrightarrow A^B$, the equivalent rule in G' is

$$C \longrightarrow \left[\left[\begin{array}{ccccccccccc} & & & & \# & & \# & & \# & & \\ & & & \# & & B & & B & & \# & \\ & & \# & & B & & B & & B & & \# \\ & \# & & A & & A & & B & & \# & \\ \# & & A & & A & & A & & \# & & \\ & \# & & A & & A & & \# & & & \\ & & \# & & \# & & \# & & & & \end{array} \right]\right]$$

Similarly, for the rules $C \to A_B$ and $C \to \text{BA}$ in G, equivalent rules can be formed in G'. Therefore, $\mathcal{L}(\text{HPG}) \subseteq \mathcal{L}(\text{RHTRG})$

For strict inclusion, consider the language L_5 over the alphabet $T = \{a, b, c, x\}$, consists of pictures of size $(2, 2, n)$, $n \geq 4$, with misaligned palindromic borders in the Z-direction, with no (2, 2, 2) hexagonal subpicture contains the symbol c in both the z-directional borders. An RHTRG can be easily defined to generate L_5 with one of the variable size rules as

$$S \to \left[\left[\begin{array}{ccccccccccccccccc} & & \# & & \# & & \# & & \# & & \# & & \# & & \# & & \\ & \# & & A_1 & & A_1 & & A_1 & & A_1 & & C_1 & & C_1 & & \# & \\ \# & & X & & X & & X & & X & & X & & X & & X & & \# \\ & \# & & C_2 & & C_2 & & A_2 & & A_2 & & A_2 & & A_2 & & \# & \\ & & \# & & \# & & \# & & \# & & \# & & \# & & \# & & \end{array}\right]\right]$$

This language cannot be generated by any HPG G', as a production of G' in the form of $S \to \begin{array}{ccccc} & A & & C & \\ X & & X & & X \\ & C & & A & \end{array}$ where A generates the z-direction palindromic string, and C, X generate z-direction strings of c's and x's, respectively, cannot restrict the occurrence of c's so that we may have a hexagonal subpicture of the form $\begin{array}{ccccc} & d & & c & \\ x & & x & & x \\ & c & & d & \end{array}$ where $d \in \{a, b\}$, but which is not a picture in L_5.

5 Conclusions

HPG is the simple type of hexagonal array rewriting models with parallel application of rewriting rules in each derivation, which gives a considerable generative capacity than CFHAG. But lack of control in the production rules make this model less general than that of RHTRG. Since hexagonal tiling patterns and kolam patterns are the applications of CFHAG, HPG can also produce these patterns. Other pattern recognition tasks by this model remain to be explored and which may further depends on the development of good parsing algorithms.

References

1. K.S. Dersanambika, K. Krithivasan, C. Martin-Vide and K.G. Subramanian, Local and recognizable hexagonal picture languages, International Journal of Pattern Recognition and Artificial Intelligence, 19 (2005), 853–871.
2. T. Kamaraj and D.G. Thomas, Regional hexagonal tile rewriting grammars, In : R.P. Barneva et al (eds.), IWCIA 2012, LNCS, Vol. 7655, 181–195, Springer, Heidelberg, 2012.

3. T. Kamaraj, D.G. Thomas and T. Kalyani, Hexagonal picture recognizability by HWA, In Proc. of ICMCM 2011, Narosa Publishing, 378–388, 2012.
4. L. Middleton and J. Sivaswamy, Hexagonal image processing: A practical approach, Advances in Computer Vision and Pattern Recognition Series, Springer, 2005.
5. D. Prusa, Two-dimensional Languages, Ph.D. Thesis, 2004.
6. G. Siromoney and R. Siromomey, Hexagonal arrays and rectangular blocks, Computer Graphics and Image Processing, 5 (1976), 353–381.
7. K.G. Subramanian, Hexagonal array grammars, Computer Graphics and Image Processing, 10 (1979), 388–394.
8. D.G. Thomas, F. Sweety and T. Kalyani, Results on Hexagonal Tile Rewriting Grammars, G. Bebis et al. (Eds.), International Symposium on Visual Computing, Part II, LNCS, 5359, Springer-Verlag, Berlin, Heidelberg (2008), 945–952.