Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 12, Number 3 (2016), pp. 1965–1978

http://www.ripublication.com/gjpam.htm

Triangular Tile Rewriting Grammars and Triangular Picture Languages

S. Kuberal
*Department of Applied Mathematics,
Pillai Institute of Information Technology,
New Panvel, Navi Mumbai-410 206, Maharastra.*

Dr. T. Kalyani
*Department of Mathematics,
St. Joseph's Institute of Technology,
Chennai-600 119.*

Dr. D.G. Thomas
*Department of Mathematics,
Madras Christian College, Chennai-600 059.*

Abstract

Tile Rewriting Grammars (TRG) are a new model for defining picture languages introduced in [1]. A rewriting rule changes a homogeneous rectangular sub-picture into an isometric one tiled with specified tiles. Derivation and language generation with TRG rules are similar to context-free grammars. F. Sweety et al. [4] have introduced Hexagonal Tile Rewriting Grammars (HTRG). In this chapter we propose Triangular Tile Rewriting Grammars (TTRG) for generating Triangular Picture Languages. TTRG is the extension of HTRG. We study the derivation of TTRG and also language generation with TTRG rules. Some closure properties are studied. TTRG is also compared with TLOC and TTS.

AMS subject classification:
Keywords: Triangular picture languages, Tile Rewriting Grammars, tiling systems.

1. Introduction

A picture is a rectangular array of terminal symbols (the pixels). A survey of formal models for picture languages is [3] where different approaches are compared and related: tiling systems, cellular automata and grammars. Classical 2D grammars can be grouped into two categories called matrix and array grammars.

The matrix grammars introduced by A. Rosenfeld [7], impose the constraint that the left and right parts of a rewriting rule must be isometric arrays; this condition overcomes the inherent problem of "shearing" which pops up while substituting a sub-array in a host array. Siromoney's array grammars [6] are parallel-sequential in nature, in the sense that first a horizontal string of non-terminals is derived sequentially, using the horizontal production; and then the vertical derivations proceed in parallel, applying a set of vertical productions.

A TRG rule is a schema having to the left a non-terminal symbol and to the right a local 2D language over terminals and non-terminals; that is the right part is specified by a set of fixed size tiles. As in matrix grammars, the shearing problem is avoided by an isometric constraint but the size of a TRG rule needs not to be fixed. The left part denotes any rectangle filled with the same non-terminal. Whatever size the left part takes, the same size is assigned to the right part. To make this idea effective, we impose a tree partial order on the areas which are rewritten.

A progressively refined equivalence relation implements the partial ordering. Derivations can then be visualized in 3D as well nested prisms, the analogue of syntax trees on string grammars.

In this paper we propose triangular tile rewriting grammars using triangular tiles. We also study some closure properties of it. It is compared with other models.

Triangular Pictures

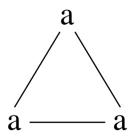
In this section, we recall some basic definition of triangular pictures and the recognizability of the triangular pictures.

Definition 1.1. A triangular picture p over the alphabet Σ is a triangular array of symbols of Σ. The set of all triangular arrays over the alphabet Σ is denoted by Σ_T^{**}. A triangular picture language over Σ is a subset of Σ_T^{**}. Given a triangular picture p, the number of rows (counting from the bottom to top), denoted by $r(p)$, is the size of a triangular picture. The empty picture is denoted by Λ.

Definition 1.2. If $p \in \Sigma_T^{**}$, then $\hat{p}$ is a triangular array obtained by surrounding p with a special boundary symbol #. Here $\# \notin \Sigma$. A triangular picture over the alphabet $\{a\}$ of size 4 surrounded by # is shown in Figure 1.

Definition 1.3. Let $p \in \Sigma_T^{**}$. Let Σ and Γ be two finite alphabets and $\pi : \Sigma \to \Gamma$ be a mapping, which we call a projection. The projection by mapping π of a triangular picture p is the picture $p' \in \Gamma_T^{**}$, such that $\pi(p(i, j, k)) = p'(i, j, k)$.

Definition 1.4.



A triangular picture of the above form is called a triangular tile over an alphabet $\{a\}$.

Definition 1.5. Given a triangular picture p of size k, for $i \leq k$, we denoted by $B_i(p)$, the set of all triangular subpictures of p of size i.

Definition 1.6. Let $L \subseteq \Sigma_T^{**}$ be a triangular picture language. The projection by mapping π of L is the language $\pi(L) = \{p'/p' = \pi(p),\ \forall\ p \in L\} \subseteq \Gamma_T^{**}$.

Definition 1.7. Let Σ be a finite alphabet. A triangular picture language $L \subseteq \Sigma_T^{**}$ is called local if there exist a finite set Δ of triangular tiles over $\Sigma \cup \{\#\}$ such that $L = \{p \in \Sigma_T^{**}/B_2(\hat{p}) \subseteq \Delta\}$. The family of Triangular Local Picture Language is denoted by TLOC.

Example 1.8. Let $\Sigma = \{0, 1\}$ be a finite alphabet.

$$\Delta = \left\{ \begin{smallmatrix} & \# & \\ \# & & \# \end{smallmatrix}, \begin{smallmatrix} & \# & \\ \# & & 1 \end{smallmatrix}, \begin{smallmatrix} & \# & \\ \# & & 0 \end{smallmatrix}, \begin{smallmatrix} & \# & \\ 1 & & \# \end{smallmatrix}, \begin{smallmatrix} & \# & \\ 0 & & \# \end{smallmatrix}, \begin{smallmatrix} & 0 & \\ \# & & \# \end{smallmatrix}, \begin{smallmatrix} & 1 & \\ \# & & \# \end{smallmatrix}, \begin{smallmatrix} & 1 & \\ 0 & & 0 \end{smallmatrix}, \begin{smallmatrix} & 0 & \\ 1 & & 1 \end{smallmatrix} \right\}$$

Then

$$L_1 = L(\Delta) = \left\{ \begin{matrix} & 0 & \\ 1 & & 1 \end{matrix}, \begin{matrix} & & 0 & & \\ & 1 & & 1 & \\ 0 & & 0 & & 0 \end{matrix}, \ldots, \begin{matrix} & & 1 & & \\ & 0 & & 0 & \\ 1 & & 1 & & 1 \end{matrix}, \ldots \right\}$$

The Language $L(G)$ is the set of all triangles of size $k \geq 2$ with alternative 0 and 1 in the rows. Clearly $L(\Delta)$ is local.

Definition 1.9. A triangular picture language $L \subseteq \Gamma_T^{**}$ is called recognizable if there exists a triangular local picture language L' (given by a set Δ of triangular tiles) over an alphabet Γ and a projection $\pi : \Gamma \to \Sigma$ such that $\pi(L') = L$. The family of recognizable triangular picture languages will be denoted by TREC.

Definition 1.10. A triangular tiling system T is a 4-tuple $(\Sigma, \Gamma, \pi, \theta)$ where Σ and Γ are two finite set of symbols, $\pi : \Gamma \to \Sigma$ is a projection and θ is a set of triangular tiles over the alphabets $\Gamma \cup \{\#\}$. The triangular picture language $L \subseteq \Gamma_T^{**}$ is tiling recognizable if there exists a tiling system $T = (\Sigma, \Gamma, \pi, \theta)$, such that $L = \pi(L(\theta))$. It is denoted by $L(T)$. The family of triangular picture language recognizable by triangular tiling system is denoted by $\mathcal{L}(TTS)$.

2. Triangular Tile Rewriting Grammars (TTRG)

In this section we introduce TTRG and study its properties.

Definition 2.1. For a finite alphabet Σ, the set of triangular picture is Σ_T^{**}. For $k \geq 1$, Σ_T^k denotes the set of triangular pictures of size k. (We will use the notation $|p| = k$). # is used when needed as a boundary symbol. $\hat{p}$ refers to the bordered version of picture p.

A pixel is an element $p(i, j, k)$. If all pixels are identical to $C \in \Sigma$ the picture is called homogeneous and denoted as C-picture.

Let X, Y, Z be the triangular axes. If the co-ordinates of a point is (a_1, a_2, a_3) then the co-ordinates of its neighbour in the direction x be $(a_1 + 1, a_2, a_3 - 1)$ and in the opposite direction be $(a_1 - 1, a_2, a_3 + 1)$. In the direction of y be $(a_1 - 1, a_2 + 1, a_3)$ and opposite direction of y be $(a_1 + 1, a_2 - 1, a_3)$. Similarly for z, the co-ordinates are $(a_1, a_2 - 1, a_3 + 1)$ and to the opposite direction $(a_1, a_2 + 1, a_3 - 1)$.

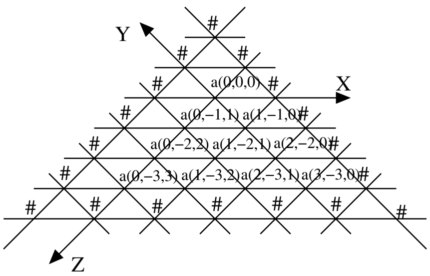


Figure 1:

Definition 2.2. Let p be a triangular picture of size t. A subpicture of p at position (i, j, k) is a picture q such that if t' is the size of q then $t' \leq t$ and there exist integers i, j, k such that,

$$p(i, j, k) = q[i - l, \{(j - m) + (t - t'), k - n], l, m, n \geq 0$$

here, $l \rightarrow$ position of a subpicture from the left
$m \rightarrow$ position of a subpicture from the bottom
$n \rightarrow$ position of a subpicture from the right

We will also write $q \trianglelefteq_{(i,j,k)} p$, or the shortcut $q \triangleleft p \equiv \exists_{i,j,k}(q \trianglelefteq_{(i,j,k)} p)$. Moreover, if $q \trianglelefteq_{(i,j,k)} p$, we define $coor_{(i,j,k)}(q, p)$ as the set of coordinates of p where q is located. Conventionally, $coor(i, j, k)(q, p) = \phi$, if q is not a subpicture of p. If q coincides with p we write $coor(p)$ instead of $coor_{(0,0,0)}(p, p)$.

Example 2.3. Let p be a triangular picture of size 5 and let q be a subpicture of p at position (1,-2,1) of size 2 then the $coor_{(1,-2,1)}(q, p) = \{(1, -2, 1), (1, -3, 2), (2, -3, 1)\}$.

Definition 2.4. Let γ be an equivalence relation on $coor(p)$, written $(x, y, z) \stackrel{\gamma}{\sim} (x', y', z')$. Two subpictures $q \trianglelefteq_{(i,j,k)} p$, $q' \trianglelefteq_{(x',y',z')} p$ are γ-equivalent, written $q \stackrel{\gamma}{\sim} q'$, iff for all pairs $(x, y, z) \in coor_{(i,j,k)}(q, p)$ and $(x', y', z') \in coor_{(x',y',z')}(q', p)$ it holds

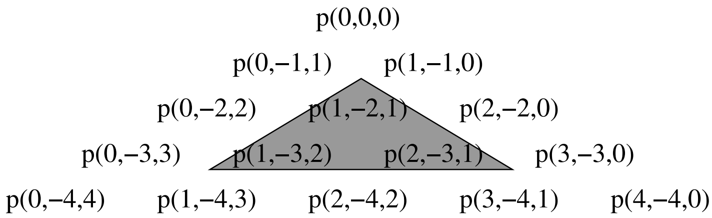


Figure 2:

$(x, y, z) \overset{\gamma}{\sim} (x', y', z')$. A homogenous C-subpicture $q \trianglelefteq p$ is called maximal with respect to relation γ iff for every γ-equivalent C-subpicture q' it is,

$$coor(q, p) \cap coor(q', p) = \phi \vee coor(q', p) \subseteq coor(q, p).$$

In other words, q is maximal if any C-subpicture which is equivalent to q is either a subpicture of q or it is not overlapping.

Definition 2.5. For a triangular picture $p \in \Sigma_T^{**}$ the set of subpictures (or tiles) with size k is: $B_k(p) = \{q \in \Sigma_T^k / q \trianglelefteq p\}$.

Definition 2.6. Consider a set of tiles $\omega \subseteq \Sigma_T^k$. The locally testable language in the strict sense defined by ω (written $Loc_u(\omega)$ 'u' stands for unbordered picture) is the set of pictures $p \in \Sigma_T^{**}$ such that $B_k(p) \subseteq \omega$. The locally testable language defined by a finite set of tiles $Loc_{u,eq}(\{\omega_1, \omega_2, \ldots, \omega_n\})$ (eq stands for the equivality test) is the set of pictures $p \in \Sigma_T^{**}$ such that for some i, $B_k(\hat{p}) = \omega_i$. The bordered locally testable language defined by a finite set of tiles $Loc_{eq}(\{\omega_1, \omega_2, \ldots, \omega_n\})$ is the set of pictures $p \in \Sigma_T^{**}$ such that for some i, $B_k(\hat{p}) = \omega_i$.

Definition 2.7. [Substitution] If p, q, q' are pictures, $q \trianglelefteq_{(i,j,k)} p$ and q, q' have the same size, then $p[q'/q]_{(i,j,k)}$ denotes the picture obtained by replacing the occurrence of q at position (i, j, k) in p with q'.

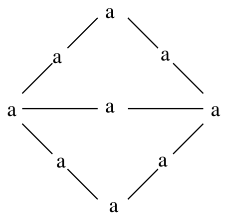
The main definition follows:

Definition 2.8. There are four types of triangular arrays and they are classified as U-array, D-array, L-array and R-array.

$$\begin{array}{ccccc} & & a & & \\ & a & & a & \\ a & & a & & a \end{array} \qquad \begin{array}{ccccc} a & & a & & a \\ & a & & a & \\ & & a & & \end{array} \qquad \begin{array}{ccc} & & a \\ & a & \\ a & & a \\ & a & \\ & & a \end{array} \qquad \begin{array}{ccc} a & & \\ & a & \\ a & & a \\ & a & \\ a & & \end{array}$$

U D L R

Horizontal overlapping
It is defined between U-array and D-array of equal size and it is denoted by the symbol $\ominus_{over}.L(G) = L(G_1) \ominus_{over}.L(G_2)$. Example: $U \ominus_{over} D$



Similarly defined for vertical overlapping $\varobar_{over}$, left overlapping $\oslash_{over}$ and right overlapping $\oslash_{over}$.

Definition 2.9. A Triangular Tile Rewriting Grammar is a tuple (Σ, N, S, R) where Σ is the terminal alphabet, N is a set of non-terminal symbols, $S \in N$ is the starting symbol, R is a set of rules.
R may contain two kinds of rules:
Fixed size: $A \to \{t_1, t_2\}$, where $A \in N$, $t_1, t_2 \in (\Sigma \cup N)^k$ with $k > 0$.
Variable size: $A \to \omega$, where $A \in N$, $\omega \in (\Sigma \cup N)^k$ with $1 \leq k \leq 2$.
Intuitively a fixed size rule is intended to match a subpicture of (small) bounded size, identical to the right part $\{t_1, t_2\}$. A variable size rule matches and subpicture of any size which can be tiled using all the elements $\{t_1, t_2\}$ of the tile set ω. However, fixed size rules are not a special case of variable size rules.

Definition 2.10. Consider a grammar $G = (\Sigma, N, S, R)$, let $p, p' \in (\Sigma \cup N)^k$ be pictures of identical size, and let γ, γ' be equivalence relations over $coor(p)$. We say that (p', γ') derives in one step from (p, γ) written $(p, \gamma) \underset{G}{\Rightarrow} (p', \gamma')$ iff for some $A \in N$ and for some rule $\rho : A \to \cdots \in R$ there exist in p a A-subpicture $r \trianglelefteq_{(i,j,k)} p$, maximal with respect to γ, such that:

- p' is obtained substituting γ with a picture s, that is $p' = p[s/r]_{(i,j,k)}$ where s is defined as follows:
  Fixed size: if $\rho = A \to \{t_1, t_2\}$, then $s = \{t_1, t_2\}$;
  Variable size: if $\rho = A \to \omega$, then $s \in Loc_{u,eq}(\omega)$.
- Let z be $coor_{(i,j,k)}(r, p)$. Let Γ be the γ-equivalence class containing z. Then, γ' is equal to γ, for all the equivalence class not equal to Γ; Γ in γ' is divided in two equivalence classes, z and its complement with respect to Γ (= ϕ if $z = \Gamma$).

More formally: $\gamma' = \gamma \backslash \{(x_1, y_1, z_1), (x_2, y_2, z_2)/(x_1, y_1, z_1) \in z$ or $(x_2, y_2, z_2) \in z\}$
The subpicture r is named the application area of the rule ρ in the derivation step. We say that (q, γ') is derivable from (p, γ) in n steps, written $(p, \gamma) \overset{n}{\Rightarrow}_G (q, \gamma')$ iff $p = q$ and $\gamma = \gamma'$ when $n = 0$, or there are a picture r and an equivalence relation γ'' such that $(p, \gamma) \overset{n-1}{\Rightarrow}_G (q, \gamma'')$ and $(\gamma, \gamma') \Rightarrow_G (q, \gamma')$. We use the abbreviation $(p, \gamma) \overset{*}{\Rightarrow}_G (q, \gamma')$ for a derivation with $n \geq 0$ steps.

Definition 2.11. The picture language defined by a grammar G (written $L(G)$) is the set of $p \in \Sigma_T^{**}$ such that, if $|p| = k$, then

$$(S^k, coor(p) \times coor(p) \times coor(p) \overset{*}{\Rightarrow}_G (p, \gamma) \tag{1}$$

where the relation γ is arbitrary. For short we write $S \overset{*}{\Rightarrow}_G p$. Notice that the derivation starts with a S-picture isometric with the terminal picture to be generated and with the universal equivalence relation over the co-ordinates. The equivalence relation computed by each step of (1) is called germinal relation. When writing examples by hand, it is convenient to visualize the equivalence classes of a germinal relation, by appending the same numerical subscript to the pixels of the application area rewritten by a derivation step. The family of languages generated by TTRG is denoted by $\mathcal{L}(TTRG)$.

To illustrate we present two examples.

Example 2.12. [Chinese triangular arrays] $G = (\Sigma, N, S, R)$, where $\Sigma = \{\wedge, <, >, 0\}$, $N = \{S\}$ and R consists of one fixed size, one variable size rule:

$$S \to \left\{ \begin{array}{ccc} & \wedge & \\ < & & > \end{array}, \begin{array}{ccc} & 0 & \\ 0 & & > \end{array} \right\},$$

$$S \to \left\{ \begin{array}{ccc} & \wedge & \\ S & & S \end{array}, \begin{array}{ccc} & 0 & \\ 0 & & S \end{array}, \begin{array}{ccc} & 0 & \\ S & & 0 \end{array}, \begin{array}{ccc} & 0 & \\ < & & S \end{array}, \begin{array}{ccc} & S & \\ S & & S \end{array}, \begin{array}{ccc} & S & \\ 0 & & S \end{array}, \begin{array}{ccc} & 0 & \\ S & & S \end{array} \right\}$$

A picture in $L(G)$ is:

$$\begin{array}{ccccccccccccccc}
 & & & & & & & \wedge & & & & & & & \\
 & & & & & & 0 & & 0 & & & & & & \\
 & & & & & 0 & & \wedge & & 0 & & & & & \\
 & & & & 0 & & 0 & & 0 & & 0 & & & & \\
 & & & 0 & & 0 & & \wedge & & 0 & & 0 & & & \\
 & & 0 & & 0 & & < & & > & & 0 & & 0 & & \\
 & 0 & & < & & 0 & & 0 & & 0 & & > & & 0 & \\
< & & 0 & & 0 & & 0 & & 0 & & 0 & & 0 & & >
\end{array}$$

$Size$ 8

Obtained by repeated application of fixed size and variable size rules. Similarly, we can get the picture language $L(G)$, for the size of 5, 8, 11, 14, ... In general, the size of the picture languages form the general form is in Arithmetic progression (A.P).

Example 2.13. [Dyck analogue] The next language, a superset of Chinese triangular arrays can be defined by a sort of blanking rule. But since the terminals cannot be deleted without shearing the picture, we replace them with a character '*'. To obtain the grammar, we add the following rules to the Chinese triangular arrays grammar:

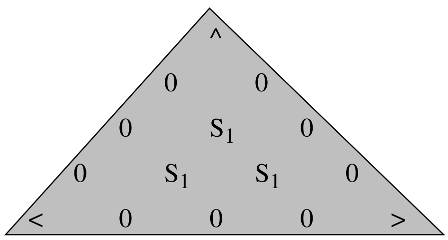
$$S \rightarrow \left[\begin{array}{ccccccccc} & & & & S & & & & \\ & & & S & & S & & & \\ & & S & & S & & S & & \\ & X & & * & & * & & X & \\ X & & X & & * & & X & & X \end{array} \Big/ \begin{array}{ccc} & X & \\ X & & X \end{array} \right] \qquad X \rightarrow \left(\begin{array}{ccc} & S & \\ S & & S \end{array} \right)$$

To illustrate, in Figure 2 we list the derivation steps of a picture. Non-terminals in the same equivalence class are marked with the same subscript.

We need to use indices in the production of the grammar:

$$\begin{array}{c} S_0 \\ S_0 \quad S_0 \\ S_0 \quad S_0 \quad S_0 \\ S_0 \quad S_0 \quad S_0 \quad S_0 \\ S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \\ S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \\ S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \\ S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \\ S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \\ S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \quad S_0 \end{array}$$

$Size$ 10



$$\begin{array}{c} X \quad * \quad * \quad * \quad * \quad X \\ X \quad X \quad * \quad * \quad * \quad X \quad X \\ X \quad X \quad X \quad * \quad * \quad X \quad X \quad X \\ X \quad X \quad X \quad X \quad * \quad X \quad X \quad X \quad X \\ X \quad X \quad X \quad X \quad X \quad X \quad X \quad X \quad X \quad X \end{array}$$

$$
\begin{array}{c}
\wedge \\
0 \quad 0 \\
0 \quad S_1 \quad 0 \\
0 \quad S_1 \quad S_1 \quad 0 \\
< \quad 0 \quad 0 \quad 0 \quad > \\
S \quad * \quad * \quad * \quad * \quad S \\
S \quad S \quad * \quad * \quad * \quad S \quad S \\
S \quad S \quad S \quad * \quad * \quad S \quad S \quad S \\
S \quad S \quad S \quad S \quad * \quad S \quad S \quad S \quad S \\
S \quad S \quad S \quad S \quad S \quad S \quad S \quad S \quad S \quad S
\end{array}
$$

A picture in $L(G)$ is:

$$
\begin{array}{c}
\wedge \\
0 \quad 0 \\
0 \quad \wedge \quad 0 \\
0 \quad < \quad > \quad 0 \\
< \quad 0 \quad 0 \quad 0 \quad > \\
\wedge \quad * \quad * \quad * \quad * \quad \wedge \\
0 \quad 0 \quad * \quad * \quad * \quad 0 \quad 0 \\
0 \quad \wedge \quad 0 \quad * \quad * \quad 0 \quad \wedge \quad 0 \\
0 \quad < \quad > \quad 0 \quad * \quad 0 \quad < \quad > \quad 0 \\
< \quad 0 \quad 0 \quad 0 \quad > \quad < \quad 0 \quad 0 \quad 0 \quad >
\end{array}
$$

and is obtained by repeated application of the variable size rule and fixed size rule.

Basic property
The family L(TTRG) is closed under union, overlapping, closures, rotation and projection.

Proof. Consider two grammars $G_1 = (\Sigma, N_1, A, R_1)$ and $G_2 = (\Sigma, N_2, B, R_2)$. Suppose for simplicity we assume that $N_1 \cap N_2 = \phi$, $S \notin N_1 \cup N_2$, and that G_1, G_2 generate pictures having size at least 2. Then it is easy to show that the grammar $G = (\Sigma, N_1 \cup N_2 \cup \{S\}, S, R_1 \cup R_2 \cup R)$, where
Union $\cup$.

$$R = \left\{ S \to \left[\left[\begin{array}{ccc} & A & \\ A & & A \end{array} \right]\right], \; S \to \left[\left[\begin{array}{ccc} & B & \\ B & & B \end{array} \right]\right] \right\}$$

is such that $L[G] = L[G_1] \cup L[G_2]$.

Overlapping.
Triangular arrays of same size can be overlapped using the following overlapping operations.

Horizontal overlapping.
It is defined between U-array and D-array of equal size and it is denoted by the symbol $\ominus_{over}$.

$$R = \left\{ S \to \left[\left[\begin{array}{ccc} & A & \\ A & & A \end{array} \right]\right],\ S \to \left[\left[\begin{array}{ccc} B & & B \\ & B & \end{array} \right]\right] \right\}$$

Vertical overlapping.
It is defined between L and R arrays of same size and it is denoted by the symbol $\obar_{over}$.

$$R = \left\{ S \to \left[\left[\begin{array}{cc} & A \\ A & \\ & A \end{array} \right]\right],\ S \to \left[\left[\begin{array}{cc} B & \\ & B \\ B & \end{array} \right]\right] \right\}$$

Right overlapping.
It is defined between any two gluale iso-triangular arrays of same size and it is denoted by the symbol $\oslash_{over}$.

$$R = \left\{ S \to \left[\left[\begin{array}{ccc} A & & A \\ & A & \end{array} \right]\right],\ S \to \left[\left[\begin{array}{ccc} & B & \\ B & & B \end{array} \right]\right] \right\}$$

Left overlapping.
It is defined between any two gluale iso-triangular arrays of same size and it is denoted by the symbol $\obslash_{over}$.

$$R = \left\{ S \to \left[\left[\begin{array}{ccc} & A & \\ A & & A \end{array} \right]\right],\ S \to \left[\left[\begin{array}{ccc} B & & B \\ & B & \end{array} \right]\right] \right\}$$

Closures: $*\ominus_{over} / * \obar_{over} / * \oslash_{over} / * \obslash_{over}$
$G = (\Sigma, N_1 \cup \{S\}, S, R_1 \cup R)$, where,

$$R = \left\{ S \to \left[\left[\begin{array}{ccc} & A & \\ A & & A \end{array} \right]\right],\ S \to \left[\left[\begin{array}{ccc} B & & B \\ & B & \end{array} \right]\right] \right\}$$

is such that $L(G) = L(G) * \ominus_{over}$. Similarly $*\obar_{over}$, $*\oslash_{over}$ $*\obslash_{over}$ cases are analogous.

Rotation about $90°$**.**
Construct the grammar $G = (\Sigma, N, A, R')$, where R', is such that, if $B \to t \in R_1$ is a fixed size rule, then $B \to t^R$ is in R'; $B \to \omega \in R_1$ is a variable size rule, then $B \to \omega'$ is in R', with $t \in \omega$ imply $t^R \in \omega'$. It is easy to verify that $L(G) = L(G_1)^R$.

Projection.
Consider a grammar $G_1 = (\Sigma_1, N_1, A, R_1)$ and a projection $\pi : \Sigma_1 \to \Sigma_2$. It is possible to build a grammar $G_2 = (\Sigma_2, N_1, A, R_2)$, such that $L(G_2) = \pi(L(G_1))$. Simply apply π to unitary rules. That is, if $X \to x \in R_1$, then $X \to \pi(x) \in R_2$, while the other rules of G_1 remain in R_2 unchanged. ■

3. Comparison Results

In this section we compare TTRG with TLOC and TTS.

Theorem 3.1. $\mathcal{L}(TLOC) \subseteq \mathcal{L}(TTRG)$.

Proof. Consider a local two-dimensional language over Σ defined by the set of allowed triangular tiles Δ.

Let $\Delta_0 = \left\{ \begin{array}{ccccccc} & & & S_0 & & & \\ & & S_0 & & S_0 & & \\ & S_0 & & S_0 & & S_0 & \\ S_0 & & S_0 & & S_0 & & S_0 \end{array} \Big/ \begin{array}{ccccccc} & & & S & & & \\ & & S & & S & & \\ & S & & S & & S & \\ S & & S & & S & & S \end{array} \right\} \in \Delta$

then an equivalent TTRG is, $G = \{\Sigma, \{S\}, S, R\}$ where R is the set $\{S \to \theta/\theta \subseteq \Delta_0\}$. ■

Lemma 3.2. $\mathcal{L}(TLOC_{u,eq}) \subseteq \mathcal{L}(TTRG)$.

Proof. Consider a Local Triangular picture language over Σ defined (without boundaries) by the sets of allowed tiles $\{\omega_1, \omega_2, \ldots, \omega_n\}$, $\omega_i \subseteq \Sigma_T^2$. An equivalent grammar is $S \to \omega_1/\omega_2/\ldots/\omega_n$. ■

Example 3.3. Consider the language

$$TLOC_{u,eq}.L(\omega) = \left\{ \begin{array}{ccc} & 0 & \\ 1 & & 1 \end{array}, \begin{array}{ccccc} & & 0 & & \\ & 1 & & 1 & \\ 0 & & 0 & & 0 \end{array}, \ldots \right\}$$

generated by

$$\omega = \left\{ \begin{array}{ccc} & 0 & \\ 1 & & 1 \end{array} \right\}.$$

It can be generated by the following TTRG.

Let $G = \{\Sigma, N, S, R\}$, when $\Sigma = \{0, 1\}$, $N = \{S\}$, and R consists of two fixed size rule and one variable size rule:

$$S \to \left\{ \begin{array}{ccc} & 0 & \\ 1 & & 1 \end{array}, \begin{array}{ccc} & 1 & \\ 0 & & 0 \end{array} \right\},$$

$$S \to \left\{ \begin{array}{ccc} & 0 & \\ S & & S \end{array}, \begin{array}{ccc} & 1 & \\ S & & S \end{array} \right\}.$$

Clearly, $\mathcal{L}(TLOC_{u,eq}) \subseteq \mathcal{L}(TTRG)$.

We now consider the notations TTS_{eq} and $TTS_{u,eq}$.

Definition 3.4. The Triangular tiling system TTS_{eq} and $TTS_{u,eq}$ are the same as a TTS, with the following respective changes;

- Replace the local language defined by Definition 1.7 with $TLOC_{eq}(\{\omega_1, \omega_2, \ldots, \omega_n\})$ where ω_i $(1 \leq i \leq n)$ is a finite set of triangular tiles over Γ.

- Replace the local language defined by Definition 1.7 with $TLOC_{u,eq}(\{\omega_1, \omega_2, \ldots, \omega_n\})$ where ω_i $(1 \leq i \leq n)$ is a finite set of triangular tiles over Γ. In $TTS_{u,eq}$ there is no boundary symbol #.

Let $\mathcal{L}(TTS_{eq})$, $\mathcal{L}(TTS_{u,eq})$ be the families of languages generated by TTS_{eq} and $TTS_{u,eq}$ respectively.

Lemma 3.5. $\mathcal{L}(TTS_{eq}) \equiv \mathcal{L}(TTS)$.

Proof. We first prove $\mathcal{L}(TTS) \subseteq \mathcal{L}(TTS_{eq})$. This is trivial, because if we consider the triangular tile set ω of a TTS by taking $\{omega_1, \omega_2, \ldots, \omega_n\} = P(\omega)$ (the power set), then we obtain an equivalent TTS_{eq}. Next we have to prove $\mathcal{L}(TTS_{eq}) \subseteq \mathcal{L}(TTS)$. In [1], the family of languages $\mathcal{L}(TLOC_{eq}(\Omega))$ where Ω is the set of tiles, is proved to be a proper subset of $\mathcal{L}(TTS)$, is closed with respect to projection, and $\mathcal{L}(TTS_{eq})$ is the closer with respect to projection of $\mathcal{L}(TLOC_{eq}(\Omega))$. Therefore $\mathcal{L}(TTS_{eq}) \subseteq \mathcal{L}(TTS)$. Hence proved. ■

Lemma 3.6. $\mathcal{L}(TTS_{u,eq}) \equiv \mathcal{L}(TTS_{eq})$.

Proof. First we prove $\mathcal{L}(TTS_{eq}) \subseteq \mathcal{L}(TTS_{u,eq})$. Let $T = (\Sigma, \Gamma, \{\omega_1, \omega_2, \ldots, \omega_n\}, \pi)$ be a TTS_{eq}. For every triangular tile set ω_i separate its tiles containing the boundary symbol # (call this subset ω_i') from the other tiles (ω_i''). This is $\omega_i = \omega_i' \cup \omega_i''$. Introduce a new alphabet Γ' and a bijective mapping $br : \Gamma \to \Gamma'$, we use symbols in Γ' to encode boundary and new tile set δ_i to contain them for every tile t in ω_i'', if there is a tile in ω_i' which overlaps with t, then a encode this boundary in a new tile t' and put it in the set δ_i. For example, suppose $\begin{array}{ccc} & a & \\ b & & c \end{array} \in \omega_i''$ overlaps with

$$\begin{array}{ccc} & a & \\ \# & & \# \end{array}, \begin{array}{ccc} & \# & \\ b & & \# \end{array}, \begin{array}{ccc} & \# & \\ \# & & c \end{array} \in \omega_i',$$

then all

$$\begin{array}{ccc} & a & \\ br(b) & & br(c) \end{array}, \begin{array}{ccc} & br(a) & \\ b & & br(c) \end{array}, \begin{array}{ccc} & br(a) & \\ br(b) & & c \end{array}$$

are in δ_i.

Consider a $TTS_{u,eq}$, $T' = (\Sigma', \Gamma \cup \Gamma', \Omega, \pi')$ where π' extends π to π' as follows:

$\pi'(br(a)) = \pi'(a) = \pi(a), a \in \Gamma$ and $ubr : \Gamma \cup \Gamma' \to \Gamma$ is defined as $ubr(a) = br^{-1}(a)$ if $a \in \Gamma'$ otherwise $= a$, and it is naturally extended to tiles and tile sets.

Ω is the set $\{\omega / \omega \subseteq \omega_i'' \cup \delta_i \wedge ubr(\omega) = \omega_i'' \wedge \omega \cup \delta_i \neq \phi \wedge 1 \leq i \leq n\}$.

The proof of $L(T) = L(T')$ is obvious.

To prove $\mathcal{L}(TTS_{u,eq}) \subseteq \mathcal{L}(TTS_{eq})$

Let $T = (\Sigma, \Gamma, \{\omega_1, \omega_2, \ldots, \omega_n\}, \pi)$ be a $TTS_{u,eq}$. To construct an equivalent TTS_{eq}, we introduce the boundary tile sets δ_i, defined as follows.

For every tile $\begin{array}{ccc} & a & \\ b & & c \end{array} \in \omega_i$, the following tiles are in δ_i.

$$\begin{array}{ccc} & \# & \\ \# & & \# \end{array}, \begin{array}{ccc} & \# & \\ \# & & a \end{array}, \begin{array}{ccc} & \# & \\ a & & \# \end{array}, \begin{array}{ccc} & \# & \\ \# & & b \end{array}, \begin{array}{ccc} & a & \\ b & & c \end{array}, \begin{array}{ccc} & \# & \\ c & & \# \end{array}, \begin{array}{ccc} & b & \\ \# & & \# \end{array}, \begin{array}{ccc} & c & \\ \# & & \# \end{array}.$$

Consider a TTS_{eq}, $T' = (\Sigma, \Gamma, \Omega, \pi)$, where Ω is the set $\{\omega \cup \omega_i / \omega \subseteq \delta_i \wedge \omega \neq \phi \wedge 1 \leq i \leq n\}$. It is easy to show that $L(T) = L(T')$. ■

Theorem 3.7. $\mathcal{L}(TTS) \subseteq \mathcal{L}(TTRG)$.

Proof. It follows the above theorem and lemmas and the fact that $\mathcal{L}(TTS_{u,eq})$ is the closure of $\mathcal{L}(TLOC_{u,eq})$ with respect to projection. ■

Remark 3.8. $\mathcal{L}(TTRG) \neq \mathcal{L}(TTS)$. To show this we give the following example. Let $G = (\Sigma, N, S, R)$ be a TTRG where $\Sigma = \{\bullet, x\}$, $N = \{S\}$, and R consists of the rules

$$S \to \begin{array}{ccc} & x & \\ \bullet & & \bullet \end{array}$$

$$S \to \left\{ \begin{array}{ccc} & S & \\ \bullet & & S \end{array}, \begin{array}{ccc} & x & \\ \bullet & & S \end{array}, \begin{array}{ccc} & \bullet & \\ \bullet & & S \end{array}, \begin{array}{ccc} & S & \\ \bullet & & \bullet \end{array}, \begin{array}{ccc} & S & \\ \bullet & & x \end{array}, \begin{array}{ccc} & \bullet & \\ S & & x \end{array}, \begin{array}{ccc} & S & \\ x & & S \end{array}, \begin{array}{ccc} & S & \\ S & & x \end{array}, \begin{array}{ccc} & x & \\ S & & S \end{array} \right\}$$

A picture in $L(G)$ is

$$\begin{array}{ccccccccccccccc}
 & & & & & & & x & & & & & & & \\
 & & & & & & \bullet & & \bullet & & & & & & \\
 & & & & & \bullet & & x & & \bullet & & & & & \\
 & & & & x & & \bullet & & \bullet & & x & & & & \\
 & & & \bullet & & \bullet & & x & & \bullet & & \bullet & & & \\
 & & \bullet & & x & & \bullet & & \bullet & & x & & \bullet & & \\
 & x & & \bullet & & \bullet & & x & & \bullet & & \bullet & & x & \\
\bullet & & \bullet & & x & & \bullet & & \bullet & & x & & \bullet & & \bullet
\end{array}$$

size 8

It is known that the language $L(G)$ is generated by the grammar $(R : CF)TAG$,

$$G = (N, I, \{\bullet, x\}, P_1 \cup P_2, S_1, S_2, T\}$$
$$N = \{S_1, S_2\}$$
$$I = \{A_1, A_2, A_2\}$$
$$T = \begin{array}{ccc} & x & \\ \bullet & & \bullet \end{array}$$
$$P_1 = \{S_1 \to T \textcircled{$\uparrow$} S_1\}[(\text{ or })P_1 = \{S_1 \to T \textcircled{$\searrow$} S_1\}$$

(or) $P_1 = \{S_1 \to T \swarrow S_1\}]$
$P_2 = \{S_1 \to A_1 \uparrow A_2 \uparrow A_3 \uparrow S_2,\ S_2 \to A_1 \uparrow A_2 \uparrow A_3\}$

$$L_{A_1} = \{< x > \bullet^n < x > \bullet^n < x >, n \geq 2\}$$
$$L_{A_2} = \{\bullet^n < x > \bullet^n < x > \bullet^n < x > \bullet^n, n \geq 2\}$$
$$L_{A_3} = \{\bullet^{n-1} < x > \bullet^n < x > \bullet^n < x > \bullet^n < x > \bullet^n < x > \bullet^{n-1}, n \geq 2\}.$$

But $L(G)$ is note generated by any $(R : R)TAG$. Clearly it is not generated by any TTS. i.e., $L(G)$ is not tiling recognizable. Here TTRG is more powerful.

4. Conclusion

In this paper we introduced Triangular Tile Rewriting Grammars (TTRG). The closure properties of TTRG are proved for some basic operations. The expressive power of TTRG is greater than the previous model TTS. We investigate further interesting properties.

References

[1] Crespi Reghizyi, S., Pradella, M., 2005, "Tile Rewriting Grammars and Picture Languages", Theoretical Comp. Science, 340, pp. 257–272.

[2] Dora Giammarresi and Antonio Restivo, 1992, "Recogniable Picture Languages", International Journal Pattern Recognition and Artificial Intelligence 6(2-3), pp. 241–256. Special issue on Parallel Image Processing.

[3] Dora Giammarresi and Antonio Restivo, 1997, "Two-dimensional Languages", In Arto Salomaa and Grzegorz Rozenberg, editors Handbook of Formal Languages, 3, Beyond Words, pp. 215–267, Springer-Verlag, Berlin.

[4] Sweety, F., Kalyani, T., Thomas, D.G., 2007, "Hexagonal Tile Rewriting Grammars", 7th National Conference on Emerging Trends in Automata (ETA'07), pp. 101–115.

[5] Stefano Crespi Reghizzi and Matteo Pradella, "Tile Rewriting Grammars", DEI-Politecnico di Milano and CNR IEIIT-MI, Piazza Leonardo da Vinci, 32, I-20133 Milano, Italy.

[6] Rani Siromoney, 1987, "Advances in Array Languages", In Hartmut Ehrig, Manfred Nagl, Grzegorz Rozenberg, and Azriel Rosenfeld, editors, Proc. 3rd Int. Workshop on Graph-Grammars and Their Application to Computer Science, 291, Lecture Notes in Computer Science, pp. 549–563. Springer-Verlag.

[7] Henning Fernau and Rudolf Freund, 1996, "Bounded parallelism in array grammars used for character recognition". In Petra Perner, Patrick Wang, and Azriel Rosenfeld, editors, Advances in Structural and Syntactical Pattern Recognition (Proceedings of the SSPR'96), 1121, pp. 40–49, Springer-Verlag.