Tile Rewriting Grammars*

Stefano Crespi Reghizzi** and Matteo Pradella

DEI - Politecnico di Milano and
CNR IEIIT-MI,
Piazza Leonardo da Vinci, 32,
I-20133 Milano, Italy
e-mail: {crespi, pradella}@elet.polimi.it

Abstract

Past proposals for applying to pictures or 2D languages the generative grammar approach do not match in our opinion the elegance and descriptive adequacy that made Context Free grammars so successful for 1D languages. In a renewed attempt, a model named Tile Rewriting Grammar is introduced combining the rewriting rules with the Tiling System of Giammaresi and Restivo which define the family of Recognizable 2D languages. The new grammars have isometric rewriting rules which for string languages are equivalent to CF rules. TRG have the capacity to generate a sort of 2D analogues of Dyck languages. Closure properties of TRG are proved for some basic operations. TRG strictly include TS as well as the context-free picture grammars of Matz.

1 Introduction

In the past several proposals have been made for applying to pictures or 2D languages the generative grammar approach but in our opinion none of them matches the elegance and descriptive adequacy that made Context Free (CF) grammars so successful for 1D i.e. string languages. A picture is a rectangular array of symbols (the pixels) from a finite alphabet.

A survey of formal definition models for picture languages is [3] where different approaches are compared and related: tiling systems, cellular automata, and grammars. The latter had been surveyed in more detail by [6]. Classical 2D grammars can be grouped in two categories[1] called *matrix* and *array* grammars respectively.

The matrix grammars, introduced by A. Rosenfeld, impose the constraint that the left and right parts of a rewriting rule must be isometric arrays; this

* Work partially supported by MIUR, Progetto *Linguaggi formali e automi, teoria e applicazioni.*

** Lecturer of Formal Languages, Università della Svizzera Italiana.

[1] Leaving aside the graph grammar models because they generate graphs, not 2D matrices.

condition overcomes the inherent problem of "shearing" which pops up while substituting a subarray in a host array.

Siromoney's array grammars are parallel-sequential in nature, in the sense that first a horizontal string of nonterminals is derived sequentially, using the horizontal productions; and then the vertical derivations proceed in parallel, applying a set of vertical productions. Several variations have been made, for instance [1]. A particular case are the two dimensional right-linear grammars studied in [3].

More recently Matz [5] has proposed a new approach relying on the notion of row and column concatenation and their closures. A production of his *context-free picture grammars* is similar to a classical 1D Context Free (CF) production, with the difference that the right part is a 2D regular expression. The shearing difficulty does not arise because, say, row concatenation is a partial operation which is only defined on matrices of identical width.

Exploring a different course, our new model is introduced, named *Tile Rewriting Grammar* (TRG), intuitively combining Rosenfeld's isometric rewriting rules with the Tiling System (TS) of Giammaresi and Restivo [2]. The latter defines the family of Recognizable 2D languages (the same as those accepted by so called *on-line tessellation automata* of Inoue and Nakamura [4]).

A TRG production is a schema having to the left a nonterminal symbol and to the right a local 2D language over terminals and nonterminals; that is the right part is specified by a set of fixed size tiles.

Similar to Rosenfeld's matrix grammars, the shearing problem is solved by a isometric constraint on the left and right parts, but a TRG production is not compelled to have a fixed rectangular size. The left part denotes a rectangle of any size filled with the same nonterminal. Whatever size the left part takes, the same size is assigned to the right part. Viewing rewriting from right to left, the effect is to replace each pixel of the right picture by the same nonterminal. To make this simple idea effective, it is necessary to impose a tree partial order on the areas which are rewritten, thus moving from the syntax tree of string languages to some sort of well nested prisms. A simple nonterminal marking device implements the partial ordering.

To our knowledge this approach is novel and is able to generate an interesting gamut of pictures, grids, spirals, and in particular a language of nested frames which is in some way the analogue of a Dyck language.

Some formal properties of the new model are here proved. Derivations can be computed in a canonical order. The family is closed with respect to some basic operations: row/column concatenation and closures, union, and rotation.

Comparing with other families, TRGs' generative capacity is stronger than that of Tiling Systems, as shown by the non-TS language made by the vertical concatenation of two specularly symmetrical squares. TRG grammars are proved more powerful than Matz's CF picture grammars.

In Sect. 2 we gradually and informally introduce the isometric productions and marking device for string languages. Then we list the basic definitions for

picture languages, and we define our model and show the closure properties. In Sect. 3 the family of TRG pictures is compared with the other families.

2 Tile Rewriting Grammars

2.1 Introducing one dimensional isometric grammars

For a gradual presentation we first sketch the TRG model for string grammars, showing it is tantamount to a notational variant of classical CF grammars. The essential features are that rewriting rules are isometric and of unbounded length. These properties set the ground for exploiting the model in two dimensions in the next section: because isometric rules matching arrays of unbounded size are exempt from the problems (shearing and narrow scope) encountered by early attempts at using phrase structure grammars for pictures.

Consider a CF grammar $G = (\Sigma, N, S, R)$ in Chomsky normal form with rules of the forms $A \to b$ and $A \to BC$, where b is a terminal and A, B, C are nonterminals.

For a derivation

$$S \stackrel{*}{\Rightarrow}_G \ uAv \stackrel{*}{\Rightarrow}_G \ uzv \in \Sigma^+, A \in N$$

we associate to each nonterminal an attribute, the length of the terminal string generated, writing

$$(S, n) \stackrel{*}{\Rightarrow}_G \ u(A, j)v \stackrel{*}{\Rightarrow}_G \ uzv \in \Sigma^+$$

where $n = |uzv|$ and $j = |z|$.

An isometric grammar $G_I = (\Sigma, N, S, R_I)$ consists of a countable set of rules corresponding to the rules of G as specified in the table:

CF rule	*Isometric rule*
$A \to b$	$\underbrace{A} \to b$
$A \to BC$	$\underbrace{AA} \to \underbrace{B}\,\underbrace{C}$
	$\underbrace{AAA} \to \underbrace{BB}\,\underbrace{C}$
	$\underbrace{AAA} \to \underbrace{B}\,\underbrace{CC}$
	$\dots$
	$\underbrace{A^n} \to \underbrace{B^p}\,\underbrace{C^q}, \forall n, p, q : n \geq 2, n = p + q$

An underbraced string must be rewritten all at once, so that the derivations of a isometric grammar are in step by step correspondence with CF derivations. More precisely, it is clear that the CF derivation

$$(S, m) \stackrel{*}{\Rightarrow}_G \ u(A, n)v \Rightarrow_G \ u(B, p)(C, q)v \stackrel{*}{\Rightarrow}_G \ uzv \in \Sigma^m, |z| = n = p + q$$

exists iff the isometric derivation exists

$$\underbrace{S^m} \stackrel{*}{\Rightarrow}_{G_I} \ u\,\underbrace{A^n}\,v \Rightarrow_{G_I} \ u\,\underbrace{B^p}\,\underbrace{C^q}\,v \stackrel{*}{\Rightarrow}_{G_I} \ uzv$$

Therefore the isometric grammar is just a notational variant of the CF grammar and defines the same language.

The semigraphical representation using underbraces has to be abandoned for a symbolic notation, that will be more convenient for the 2D case. Symbols in the right part of a rule are marked by natural indices, qualified as *update* indices and used to denote underbraces, so that the CF rule $A \to BC$ produces the set of isometric rules

$$A^n \to {B_1}^p {C_2}^q,\ \forall n,p,q : n \geq 2, n = p + q \tag{1}$$

A nonterminal previously underbraced is marked by distinct index, called *area* index. An area index updating rule is added to a derivation step.

Initially the axiom has area index 0, and the derivation starts from $(S_0)^m$. Assume a string $\gamma = \alpha(A_i)^n\beta$ deriving from $(S_0)^m$ is such that the nonterminals that would be underbraced carry the same index i, which differs from all other indices. Moreover neither α nor β contain A_i, that is $(A_i)^n$ is the largest substring containing A_i. Let $\mu(\gamma)$ be the maximum area index present in γ. Then when a derivation step is performed, applying the rewriting rule (1), the indices of the right hand side symbols are incremented by $\mu(\gamma)$:

$$\alpha(A_i)^n\beta \Rightarrow\ \alpha(B_r)^p(C_s)^q\beta$$

$$n = p + q,\ r = 1 + \mu(\gamma),\ s = 2 + \mu(\gamma)$$

As a consequence, the area indices of the resulting string preserve the divisions.

Example 1. Consider the following CF grammar: $S \to SS; S \to AB; A \to a; B \to b$. We can write it as follows:

$$S^n \to (S_1)^p(S_2)^q;\ S^n \to (A_1)^p(B_2)^q,\ \forall n,p,q : n = p + q \geq 2$$
$$A \to a_0;\ B \to b_0$$

Notice that we use update index 0 for terminals.

The CF derivation (with sizes):

$$(S,4) \Rightarrow (S,2)(S,2) \Rightarrow (A,1)(B,1)(S,2) \overset{2}{\Rightarrow} abS \Rightarrow ab(A,1)(B,1) \overset{2}{\Rightarrow} abab.$$

corresponds to the derivation with indexed symbols:

$$(S_0)^4 \Rightarrow (S_1)^2(S_2)^2 \Rightarrow A_3B_4(S_2)^2 \overset{2}{\Rightarrow} a_3b_4(S_2)^2 \Rightarrow a_3b_4A_5B_6 \overset{2}{\Rightarrow} a_3b_4a_5b_6.$$

Since the right hand side of (1) is a locally testable language over $\{B_1, C_1\}$, the set of rules will be defined in closed form by listing the substrings of length two (or tiles) which may occur:

$$A \to \{B_1C_2\};\ A \to \{B_1B_1, B_1C_2\};\ A \to \{B_1C_2, C_2C_2\};$$
$$A \to \{B_1B_1, B_1C_2, C_2C_2\}$$

Notice that by the isometric hypothesis the left hand side is assumed to match the length of the right hand side, and its length is not specified.

Although this technique is clumsy for string languages, it permits to do without 2D bracketing in picture languages.

2.2 Basic Definitions

We list the preliminary notation and definitions, most of them as well as the omitted ones are in [3].

Definition 2. For a terminal alphabet Σ, the set of pictures is Σ^{**}. For $h, k \geq 1$, $\Sigma^{(h,k)}$ denotes the set of pictures of size (h,k) (i.e. $|p| = (h,k)$):

$$p \in \Sigma^{(h,k)} \iff p = \begin{matrix} p(1,1) & \dots & p(1,k) \\ \vdots & \ddots & \vdots \\ p(h,1) & \dots & p(h,k) \end{matrix}$$

Row and column concatenations are denoted $\ominus$ and $\oplus$, respectively. $p \ominus q$ is defined iff p and q have the same number of columns; the resulting picture is the vertical juxtaposition of p over q. $p^{k\ominus}$ is the vertical juxtaposition of k copies of p; $p^{*\ominus}$ is the corresponding closure. $\oplus$,$^{k\oplus}$,$^{*\oplus}$, are the column analogous.

Definition 3. The (vertical) *mirror image* and the (clockwise) *rotation* of a picture p (with $|p| = (h,k)$), respectively, are defined as follows:

$$Mirror(p) = \begin{matrix} p(h,1) & \dots & p(h,k) \\ \vdots & \ddots & \vdots \\ p(1,1) & \dots & p(1,k) \end{matrix} \; ; \; p^R = \begin{matrix} p(h,1) & \dots & p(1,1) \\ \vdots & \ddots & \vdots \\ p(h,k) & \dots & p(1,k) \end{matrix}$$

Definition 4. Let $p \in \Sigma^{(h,k)}, q \in \Sigma^{(h',k')}, h' \leq h,\ k' \leq k$. q is called a *subpicture* (or *window*) of p at position (i,j) (also written $q \trianglelefteq_{(i,j)} p$) iff:

$$\forall i', j' (i' \leq h' \wedge j' \leq k' \wedge i < h - h' \wedge j < k - k' \Rightarrow p(i+i', j+j') = q(i', j'))$$

We will use the shortcut $q \trianglelefteq p$, for $\exists i, j (q \trianglelefteq_{(i,j)} p)$.

Definition 5. For a picture $p \in H^{**}$ the *set of subpictures* with size (h,k) is:

$$B_{h,k}(p) = \{q \in H^{(h,k)} \mid q \trianglelefteq p\}$$

We assume $B_{1,k}$ to be only defined on $\Sigma^{(1,*)}$ (horizontal string), and $B_{h,1}$ on $\Sigma^{(*,1)}$ (vertical string).

Definition 6. Consider a set ω of pictures $t \in \Sigma^{(i,j)}$. The *locally testable language* defined by ω (called $LOC(\omega)$) is the set of pictures $p \in \Sigma^{**}$ such that $B_{i,j}(p) = \omega$.

Moreover the *locally testable language in the strict sense* defined by ω (written $LOCss(\omega)$) is the set of pictures $p \in \Sigma^{**}$ such that $B_{i,j}(p) \subseteq \omega$

Definition 7. *Substitution.* If p, q, q' are pictures, $q \trianglelefteq_{(i,j)} p$, and q, q' have the same size, then $p[q'/q_{(i,j)}]$ denotes the picture obtained by replacing the occurrence of q at position (i,j) in p with q'.

For simplicity, we shall drop the subscript (i,j) when there is a single occurrence of q in p.

The main definition follows.

Definition 8. A *Tile Rewriting Grammar (TRG)* is a tuple (Σ, N, S, R), where Σ is the terminal alphabet, N is a set of *nonterminal* symbols, $S \in N$ is the *starting symbol*, R is a set of rules.

An *indexed symbol* is an element from $I = (\Sigma \times \{0\}) \cup (N \times \mathbb{N}_{\geq 1})$.

R may contain two kinds of rules:

Type 1: $A \to t$, where $A \in N$, $t \in I^{(h,k)}$, with $h, k > 0$;
Type 2: $A \to \omega$, where $A \in N$ and $\omega \subseteq \{t \mid t \in I^{(i,j)}\}$, with $i, j > 0$.

Notice that type 1 is not a special case of type 2. Moreover, the update index used for terminals is 0, while indices for nonterminals are greater than 0.

Intuitively a rule of type 1 is intended to match a subpicture of small bounded size, identical to the right part t. A rule of type 2 matches any subpicture of any size which can be tiled using *all* the elements t of the tile set ω.

We define a procedure to be used in the derivation step.

Definition 9. *Rewrite procedure.* Consider a TRG $G = (\Sigma, N, S, R)$, a rule $\rho \in R$, and a picture $p \in ((\Sigma \cup N) \times \mathbb{N})^{**}$. Then its *maximum index* (written $\mu(p)$) is defined as: $\mu(p) = Max\{k \mid p(i,j) = (\chi, k), \chi \in \Sigma \cup N\}$.

Rewrite(**in** ρ, p; **out** q)
1: Find a maximal[2] $r \in (A, k)^{**}$, where A is the left part of ρ, such that $r \trianglelefteq_{(m,n)} p$, for some m, n. r is called the *application area*.
2: If $\rho = A \to \omega$ (i.e. ρ is a type 2 rule), then choose a picture $s \in LOC(\omega)$, with $|s| = |r|$.
Otherwise if $\rho = A \to t$ (type 1), set $s := t$, the right part of ρ.
3: Let s' be the picture defined by $\forall i, j : s(i,j) = (\chi, l), \chi \in \Sigma \cup N \Rightarrow s'(i,j) = (\chi, l + \mu(p))$. That is, s' is computed incrementing the indices in s by $\mu(p)$.
4: Return $q = p[s'/r_{(m,n)}]$.
end.

Definition 10. A *derivation in one step* is a relation

$$\Rightarrow_G \subseteq ((\Sigma \cup N) \times \mathbb{N})^{(h,k)} \times ((\Sigma \cup N) \times \mathbb{N})^{(h,k)}$$

Then $p \Rightarrow_G q$ iff there exists a rule $\rho \in R$ such that $Rewrite(\rho, p, q)$.
We say that q is *derivable from* p in n step, in symbols $p \overset{n}{\Rightarrow}_G q$, iff $p = q$, when $n = 0$, or there is a picture $r \in ((\Sigma \cup N) \times \mathbb{N})^{(h,k)}$ such that $p \overset{n-1}{\Rightarrow}_G r$ and $r \Rightarrow_G q$. As usual $p \overset{*}{\Rightarrow}_G q$ says that $p \overset{n}{\Rightarrow}_G q$, for some $n \geq 0$.

Definition 11. The *picture language* defined by a TRG G (written $L(G)$) is the set of $p \in \Sigma^{**}$ such that, if $|p| = (h,k)$, then $(S,0)^{(h,k)} \overset{*}{\Rightarrow}_G p|_1$, where $|_1$ denotes the projection on Σ of the elements in $\Sigma \times \mathbb{N}$. For short we write $S \overset{*}{\Rightarrow}_G p$.

[2] i.e. it is not the case that $\exists q \neq r$ such that $q \in (A,k)^{**}$ and $r \trianglelefteq q$.

The following obvious statements may be viewed as a 2D formulation of well known properties of 1D CF derivations.

Let $p_1 \Rightarrow p_2 \Rightarrow \ldots \Rightarrow p_n$ be a derivation, and $\alpha(p_1), \alpha(p_2), \ldots \alpha(p_{n-1})$ the corresponding application areas.

Disjointness of application areas: For any $p_i, p_j, i < j$, either $\alpha(p_i) \trianglelefteq \alpha(p_j)$, or $\alpha(p_i)$, $\alpha(p_j)$ are disjoint.

Canonical derivation: Let $c_i = (x_i, y_i)$ be the coordinates of the top left-most corner of the application area $\alpha(p_i)$. The previous derivation is *lexicographic* iff $i < j$ implies $c_i \leq_{lex} c_j$ (where $\leq_{lex}$ is the usual lexicographic order). Then

$$L(G) \equiv \{p \mid S \stackrel{*}{\Rightarrow}_G p \text{ and } \stackrel{*}{\Rightarrow}_G \text{ is a lexicographic derivation}\}$$

To simplify the notation, in the sequel we will use subscripts for indexed symbols (e.g. S_1 instead of $(S, 1)$). Moreover, we will drop the index when possible, for terminals and in rules where all nonterminals have the same index.

To illustrate we present two examples.

Example 12. Chinese boxes. $G = (\Sigma, N, S, R)$, where $\Sigma = \{\ulcorner, \urcorner, \llcorner, \lrcorner, \circ\}$, $N = \{S\}$, and R consists of the rules:

$$S \to \begin{matrix} \ulcorner & \urcorner \\ \llcorner & \lrcorner \end{matrix}; \; S \to \left\{ \begin{matrix} \ulcorner\circ & \circ S & \circ S & S\,S & \circ\circ & S\,S & \circ\urcorner & S\circ & S\circ \\ \circ S\text{'} & \llcorner\circ\text{'} & \circ S\text{'} & \circ\circ\text{'} & S\,S\text{'} & S\,S\text{'} & S\circ\text{'} & S\circ\text{'} & \circ\lrcorner \end{matrix} \right\}$$

(where indices have been dropped).

A picture in $L(G)$ is:

$$\begin{matrix} \ulcorner & \circ & \circ & \circ & \circ & \urcorner \\ \circ & \ulcorner & \circ & \circ & \urcorner & \circ \\ \circ & \circ & \ulcorner & \urcorner & \circ & \circ \\ \circ & \circ & \llcorner & \lrcorner & \circ & \circ \\ \circ & \llcorner & \circ & \circ & \lrcorner & \circ \\ \llcorner & \circ & \circ & \circ & \circ & \lrcorner \end{matrix}$$

For convenience, we will often specify a set of tiles by a sample picture exhibiting the tiles as its subpictures (using $B_{h,k}$ - see Definition 5). We write | to separate alternative right parts of rules with the same left part (analogously to string grammars). Therefore, the previous grammar becomes:

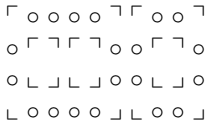
$$S \to \begin{matrix} \ulcorner & \urcorner \\ \llcorner & \lrcorner \end{matrix} \mid B_{2,2} \begin{pmatrix} \ulcorner & \circ & \circ & \urcorner \\ \circ & S & S & \circ \\ \circ & S & S & \circ \\ \llcorner & \circ & \circ & \lrcorner \end{pmatrix}$$

Example 13. 2D Dyck analogue. The next language L_{box}, a superset of Chinese boxes, can be defined by a sort of cancellation rule. But since terminals cannot be cancelled without shearing the picture, we replace them by a character b, the blank or background.

Empty frame: Let $k \geq 0$. An *empty frame* is a picture defined by the regular expression: $(\ulcorner \oplus (\circ)^{k\oplus} \oplus \urcorner) \ominus (\circ \oplus b^{k\oplus} \oplus \circ)^{k\ominus} \ominus (\llcorner \oplus (\circ)^{k\oplus} \oplus \lrcorner)$, i.e. a box containing just b's.

Cancellation: The *cancellation* of an empty frame p is the picture $del(p)$ obtained by applying the projection $del(x) = b, x \in \Sigma \cup \{b\}$.

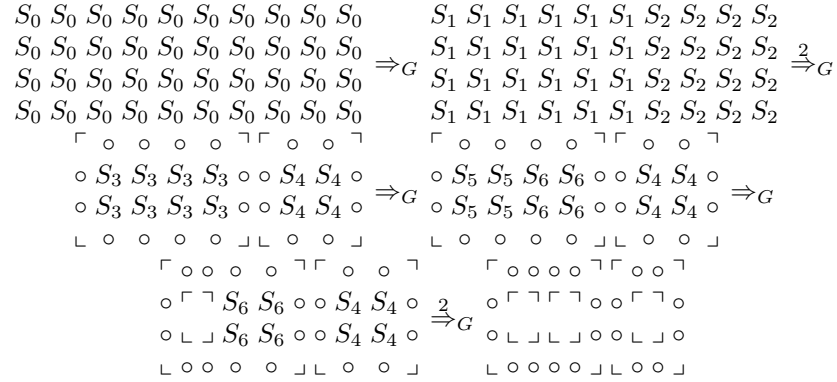
A picture p is in L_{box} iff by repeatedly applying *del* on subpictures which are emply frames, an empty frame is obtained, as in the picture:



This time we need to use indices in the productions of the grammar:

$$S \to B_{2,2} \begin{pmatrix} S_1 & S_1 & S_2 & S_2 \\ S_1 & S_1 & S_2 & S_2 \end{pmatrix} \mid B_{2,2} \begin{pmatrix} S_1 & S_1 \\ S_1 & S_1 \\ S_2 & S_2 \\ S_2 & S_2 \end{pmatrix}$$

To illustrate we list the derivation steps of the previous picture:



Although this language can be viewed as a 2D analogue of a Dyck's string language, variations are possible and we do not claim the same algebraic properties as in 1D.

2.3 Closure Properties

For simplicity, in the following theorem we suppose that $L(G_1), L(G_2)$ contain pictures of size at least (2,2).

Theorem 14. *The family $\mathcal{L}(TRG)$ is closed under union, column/row concatenation, column/row closure operations, rotation, and alphabetical mapping (or 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 that $N_1 \cap N_2 = \emptyset$, and $S \notin N_1 \cup N_2$. 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 $\cup$:

$$R = \left\{ S \to B_{2,2} \begin{pmatrix} A & A \\ A & A \end{pmatrix},\ S \to B_{2,2} \begin{pmatrix} B & B \\ B & B \end{pmatrix} \right\}$$

is such that $L(G) = L(G_1) \cup L(G_2)$.

Concatenation $\oplus/\ominus$:

$$R = \left\{ S \to B_{2,2} \begin{pmatrix} A & A & B & B \\ A & A & B & B \end{pmatrix} \right\}$$

is such that $L(G) = L(G_1) \oplus L(G_2)$. The row concatenation case is analogous.

Closures ${}^{*\oplus}/{}^{*\ominus}$:

$$G = (\Sigma, N_1 \cup \{S\}, S, R_1 \cup R)$$

where

$$R = \left\{ S \to B_{2,2} \begin{pmatrix} S_1 & S_1 & S_2 & S_2 \\ S_1 & S_1 & S_2 & S_2 \end{pmatrix} \right\}$$

is such that $L(G) = L(G_1)^{*\oplus}$. The row closure case is analogous.

Rotation R: Construct the grammar $G = (\Sigma, N, A, R')$, where R' is such that, if $B \to t \in R_1$ is a type 1 rule, then $B \to t^R$ is in R'; if $B \to \omega \in R_1$ is a type 2 rule, then $B \to \omega'$ is in R', with $t \in \omega$ implies $t^R \in \omega'$. It is easy to verify that $L(G) = L(G_1)^R$.

Projection π: Consider a grammar $G = (\Sigma_1, N, S, R)$ and a projection $\pi : \Sigma_1 \to \Sigma_2$. It is possible to build a grammar $G' = (\Sigma_2, N', S, R')$, such that $L(G') = \pi(L(G))$. Indeed, let Σ_1' be a set of new nonterminals corresponding to elements in Σ_1, then $N' = N \cup \Sigma_1'$; $R' = \phi(R) \cup R''$, where $\phi : \Sigma_1 \times \{0\} \to \Sigma_1' \times \{k\}$ is the alphabetical mapping $\phi(a) = a_k'$, where k is a fixed unused updating index, and it is naturally extended to TRG rules. Moreover, we need the additional projection rules R'' (where $v = \pi(a)$):

$$a' \to v_0 \mid \left\{ \begin{matrix} v_0 & v_0 \\ v_0 & v_0 \end{matrix} \right\} \mid \{ v_0\ v_0 \} \mid \left\{ \begin{matrix} v_0 \\ v_0 \end{matrix} \right\}$$

3 Comparison with other models

We first compare with Tiling Systems, then with Matz's CF grammars.

Theorem 15.

$$\mathcal{L}(LOCss) \subseteq \mathcal{L}(TRG)$$

Proof. Consider a local two-dimensional language over Σ defined by the set of allowed blocks Θ.

Let $\Theta_0 = \{ \begin{smallmatrix} x_0 & y_0 \\ z_0 & w_0 \end{smallmatrix} \mid \begin{smallmatrix} x & y \\ z & w \end{smallmatrix} \in \Theta\}$, then an equivalent TRG is $G = (\Sigma, \{S\}, S, R)$, where R is the set $\{S \to \theta \mid \theta \subseteq \Theta_0\}$.

The following theorem is a consequence of the fact that $\mathcal{L}(TRG)$ is the closure of $\mathcal{L}(LOCss)$ with respect to projection.

Theorem 16.

$$\mathcal{L}(TS) \subseteq \mathcal{L}(TRG)$$

The following strict inclusion is an immediate consequence of the fact that, for 1D languages, $\mathcal{L}(TS) \subset \mathcal{L}(CF)$, and $\mathcal{L}(TRG) = \mathcal{L}(CF)$. But we prefer to prove it by exhibit an interesting picture language, namely a language made by the vertical concatenation of two specularly symmetrical rectangles.

Theorem 17.

$$\mathcal{L}(TS) \neq \mathcal{L}(TRG)$$

Proof. Let $\Sigma = \{a, b\}$. Consider the language $L = \{p \mid p = s \ominus Mirror(s)$ and $s \in \Sigma^{(h,k)}, h > 1, k \geq 1\}$. We prove that $L \notin \mathcal{L}(TS)$ using a technique very similar to that of Theorem 7.5 in [3].

Consider the grammar G:

$$A \rightarrow \begin{matrix} a \\ a \end{matrix} \mid \begin{matrix} b \\ b \end{matrix} \mid B_{2,1} \begin{pmatrix} a \\ A \\ A \\ a \end{pmatrix} \mid B_{2,1} \begin{pmatrix} b \\ A \\ A \\ b \end{pmatrix}$$

$$S \rightarrow B_{2,2} \begin{pmatrix} A & S & S \\ A & S & S \end{pmatrix} \mid B_{2,1} \begin{pmatrix} A \\ A \end{pmatrix}$$

Without proof, it is easy to see that $L(G) = L$.
We prove by contradiction that $L \notin \mathcal{L}(TS)$. Suppose that $L \in \mathcal{L}(TS)$, therefore L is a projection of a local language L' defined over some alphabet Γ. Let $\sigma = |\Sigma|$ and $\gamma = |\Gamma|$, with $\sigma \leq \gamma$. For an integer n, let:

$$L_n = \{p \mid p = s \ominus Mirror(s) \text{ and } |s| = (n, n)\}.$$

Clearly, $|L_n| = \sigma^{n^2}$. Let L'_n be the set of pictures in L' over Γ whose projections are in L_n. By choice of γ and by construction of L_n there are at most γ^n possibilities for the n-th and $(n+1)$-th rows in the pictures of L'_n, because this is the number of mirrored stripe pictures of size $(2, n)$ on Γ.

For n sufficiently large $\sigma^{n^2} \geq \gamma^n$. Therefore, for such n, there will be two different pictures $p = s_p \ominus Mirror(s_p), q = s_q \ominus Mirror(s_q)$ such that the corresponding $p' = s'_p \ominus s''_p, q' = s'_q \ominus s''_q$ have the same n-th and $(n+1)$-th rows. This implies that, by definition of local language, pictures $v' = s'_p \ominus s''_q, w' = s'_q \ominus s''_p$ belong to L'_n, too. Therefore, pictures $\pi(v') = s_p \ominus Mirror(s_q)$, and $\pi(w') = s_q \ominus Mirror(s_p)$ belong to L_n. But this is a contradiction.

The other family of languages to be compared are a different generalisation of CF grammars in two dimensions, Matz's CF Picture Grammars ($CFPG$)[5]. These grammars are syntactically very similar to 1D CFs. The main difference is

that their right parts use $\oplus, \ominus$ operators. Nonterminals denote unbound rectangular pictures. Derivation is analogous to 1D, but the resulting regular expression may or may not define a picture (e.g. $a \oplus (b \ominus b)$ does not generate any picture).

Theorem 18.

$$\mathcal{L}(CFPG) \subseteq \mathcal{L}(TRG)$$

Hint of the proof. Consider now a Matz's CFPG grammar in Chomsky Normal Form. It may contain three types of rules: $A \to B \oplus C$; $A \to B \ominus C$; $A \to a$. Then, $A \to B \oplus C$ corresponds to the following TRG rules:

$$A \to B_{2,2} \begin{pmatrix} B\ B\ C\ C \\ B\ B\ C\ C \end{pmatrix} \mid B_{2,2} \begin{pmatrix} B\ C\ C \\ B\ C\ C \end{pmatrix} \mid B_{2,2} \begin{pmatrix} B\ B\ C \\ B\ B\ C \end{pmatrix} \mid$$

$$B\ C \mid B_{1,2} \begin{pmatrix} B\ B\ C\ C \end{pmatrix} \mid B_{1,2} \begin{pmatrix} B\ C\ C \end{pmatrix} \mid B_{1,2} \begin{pmatrix} B\ B\ C \end{pmatrix}$$

Notice that a CFPG production $A \to B \oplus B$, with two copies of the same nonterminal B, imposes the use of indices in the corresponding TRG productions:

$$A \to B_{2,2} \begin{pmatrix} B_1\ B_1\ B_2\ B_2 \\ B_1\ B_1\ B_2\ B_2 \end{pmatrix} \mid \dots$$

The $\ominus$ case is analogous, while $A \to a$ is trivial.

Theorem 19.

$$\mathcal{L}(CFPG) \neq \mathcal{L}(TRG)$$

Proof. It is a consequence of Theorems 16, 17, and 18, and the fact that $\mathcal{L}(TS) \not\subseteq \mathcal{L}(CFPG)$, as reported in [5].

An example of a TRG but not CFPG language is the following. We know from [5] that the language which consists of a thin cross of b's on a field of a's is not in $\mathcal{L}(CFPG)$. It is easy to show that the following TRG defines the "cross" language:

$$S \to B_{2,2} \begin{pmatrix} B\ B\ A\ A \\ B\ B\ A\ A \\ C\ C\ D\ D \\ C\ C\ D\ D \end{pmatrix}; \quad \begin{array}{l} B \to B_{2,2} \begin{pmatrix} a\ a \\ a\ a \\ b\ b \end{pmatrix}; \ A \to B_{2,2} \begin{pmatrix} b\ a\ a \\ b\ a\ a \\ b\ b\ b \end{pmatrix} \\ C \to B_{2,2} \begin{pmatrix} a\ a \\ a\ a \end{pmatrix}; \ D \to B_{2,2} \begin{pmatrix} b\ a\ a \\ b\ a\ a \end{pmatrix} \end{array}$$

4 Conclusions

The new TRG grammar model extends the context-free string grammars to two dimensions. As the application area of a rewriting rule is an unbounded subpicture, we use tiling to specify it. In a derivation the application areas rewritten are partially ordered by the subpicture relation, as in a syntax tree. In three dimensions a derivation can be depicted as a set of prisms whose bases are the

application areas of the productions, the analogue of the syntax tree for strings. The Dyck strings then become the picture language of well nested boxes, in one of several possible variations.

The expressive power of TRG is greater than of two previous models: the Tiling Systems of [2] which define 2D recognizable languages; and Matz's context free picture grammars, another generalisation of context free grammars. More work is needed to compare with other grammar families and to assess the suitability of TRG for practical applications.

The analogy with CF string grammars raises to the educated formal linguists many obvious questions, such as the formulation of a pumping lemma, of Chomsky-Schutzenberger theorem, etc.

It would be also interesting to define a type 1 TRG model in the sense of Chomsky's hierarchy.

Acknowledgement

We thank Antonio Restivo for calling our attention to the problem of inventing a class of grammars suitable to define "2D Dyck languages". We thank Alessandra Cherubini and Pierluigi San Pietro for their comments, and the anonymous referees for helpful suggestions.

References

1. H. Fernau and R. Freund. Bounded parallelism in array grammars used for character recognition. In P. Perner, P. Wang, and A. Rosenfeld, editors, *Advances in Structural and Syntactical Pattern Recognition (Proceedings of the SSPR'96)*, volume 1121, pages 40–49. Berlin: Springer, 1996.
2. D. Giammarresi and A. Restivo. Recognizable picture languages. *International Journal Pattern Recognition and Artificial Intelligence*, 6(2-3):241–256, 1992. Special Issue on *Parallel Image Processing*. M. Nivat, A. Saoudi and P.S.P. Wangs (Eds.).
3. D. Giammarresi and A. Restivo. Two-dimensional languages. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, volume 3, Beyond Words, pages 215–267. Springer-Verlag, Berlin, 1997.
4. K. Inoue and A. Nakamura. Some properties of two-dimensional on-line tessellation acceptors. *Information Sciences*, 13:95–121, 1977.
5. Oliver Matz. Regular expressions and context-free grammars for picture languages. In *14th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1200 of *lncs*, pages 283–294, Lübeck, Germany, 27 February–March 1 1997. Springer.
6. Rani Siromoney. 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*, volume 291 of *Lecture Notes in Computer Science*, pages 549–563. Springer-Verlag, 1987.