Regional languages and tiling: a unifying approach to picture grammars*

Alessandra Cherubini[1], Stefano Crespi Reghizzi[1], and Matteo Pradella[2]

[1] Politecnico di Milano
[2] CNR IEIT-MI
P.zza L. da Vinci, 32, 20133 Milano, Italy
{alessandra.cherubini, stefano.crespireghizzi, matteo.pradella}@polimi.it

Abstract. Several classical models of picture grammars based on array rewriting rules can be unified and extended by a tiling based approach. The right part of a rewriting rule is formalized by a finite set of permitted tiles. We focus on a simple type of tiling, named *regional*, and define the corresponding regional tile grammars. They include both Siromoney's (or Matz's) Kolam grammars, and their generalization by Průša. Regionally defined pictures can be recognized with polynomial time complexity by an algorithm extending the CKY one for strings. Regional tile grammars and languages are strictly included into the tile grammars and languages, and are incomparable with Giammarresi-Restivo tiling systems (or Wang's tilings).

Keywords: picture language, tiling, picture grammar, 2D language, CKY algorithm, syntactic pattern recognition.

1 Introduction

Several classical models of picture grammars based on array rewriting rules can be unified by a tiling based approach. The right part of a rewriting rule can be specified by a finite set of permitted two by two tiles. We focus on a simple type of tiling, named regional and define the corresponding regional tile grammars. The new class generalizes some classical models, yet it permits efficient, polynomial-time recognition of pictures by an approach extending the classical CKY algorithm [12] of context-free (CF) string languages.

Regional tile grammars can be viewed from the standpoint of less, or of more powerful models. First, regional tile grammars are a generalization of the classical Kolam grammars of Siromoney [11] (which are equivalent to the grammars of Matz [7]), where the right parts of grammar rules are tiled in ways than cannot be obtained by 2D regular expressions.

From the standpoint of more powerful grammar models, regional tile grammars correspond to a natural restriction of the recently introduced tile (rewriting) grammars (TG). Such grammars have rewriting rules that replace a homogeneous non-terminal

* Work partially supported by ESF *Automata: from Mathematics to Applications (AutoMathA).*

rectangular area with a picture belonging to a local language defined by tiles. It is known that the TG family dominates the family of languages defined by the Tiling Systems of Giammarresi and Restivo [4] (which are equivalent to Wang's tilings [1]), and that the latter are NP-complete with respect to picture recognition time complexity.

The new model can be conveniently defined starting from TG grammars, by imposing the constraint that the local language used in a rule is made by assembling a finite number of homogeneous rectangular pictures. Such tiling is related to Simplot's [9] interesting closure operation on pictures.

The presentation continues in Sect. 2 with preliminary definitions, and in Sect. 3 with the definition of regional tile grammars and relevant examples. In Sect. 4 we present the parsing algorithm and prove its correctness and complexity. In Sect. 5 we compare regional tile grammars and languages with other picture language families.

2 Basic definitions and regional local languages

The following notation and definitions are mostly from [5] and [2].

Definition 1. *Let Σ be a finite alphabet. A two-dimensional array of elements of Σ is a picture over Σ. The set of all pictures over Σ is Σ^{++}. A picture language is a subset of Σ^{++}.*

For $h, k \geq 1$, $\Sigma^{(h,k)}$ denotes the set of pictures of size (h, k) (we will use the notation $|p| = (h, k), |p|_{row} = h, |p|_{col} = k$). $\# \notin \Sigma$ is used when needed as a boundary symbol*; $\hat{p}$ refers to the bordered version of picture p. That is, for $p \in \Sigma^{(h,k)}$, it is*

$$p = \begin{matrix} p(1,1) & \dots & p(1,k) \\ \vdots & \ddots & \vdots \\ p(h,1) & \dots & p(h,k) \end{matrix} \qquad \hat{p} = \begin{matrix} \# & \# & \dots & \# & \# \\ \# & p(1,1) & \dots & p(1,k) & \# \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \# & p(h,1) & \dots & p(h,k) & \# \\ \# & \# & \dots & \# & \# \end{matrix}$$

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

The domain of a picture p is the set $\mathrm{dom}(p) = \{1, \dots, |p|_{row}\} \times \{1, \dots, |p|_{col}\}$.

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 2. *Let p be a picture over Σ. A* subdomain *of* $\mathrm{dom}(p)$ *is a set d of the form $\{x, \dots, x'\} \times \{y, \dots, y'\}$ where $1 \leq x \leq x' \leq |p|_{row}$, $1 \leq y \leq y' \leq |p|_{col}$. We will often denote a subdomain by using its top-left and bottom-right coordinates, in the previous case the quadruple $(x, y; x', y')$.*

The set of subdomains of p is denoted $D(p)$. Let $d = \{x, \dots, x'\} \times \{y, \dots, y'\} \in D(p)$, the subpicture $\mathrm{spic}(p, d)$ *associated to d is the picture of size $(x' - x + 1,\ y' - y + 1)$ such that $\forall i \in \{1, \dots, x' - x + 1\}$ and $\forall j \in \{1, \dots, y' - y + 1\}$ $\mathrm{spic}(p, d)(i, j) = p(x + i - 1, y + j - 1)$.*

A subdomain is called C-homogeneous when its associated subpicture is a C-picture. C is called the label *of the subdomain.*

Two subdomains $d_a = (i_a, j_a; k_a, l_a)$ and $d_b = (i_b, j_b; k_b, l_b)$ are horizontally adjacent *(resp.* vertically adjacent*) iff $j_b = l_a + 1$, and $k_b \geq i_a, k_a \geq i_b$ (resp. $i_b = k_a + 1$, and $l_b \geq j_a, l_a \geq j_b$).*

The translation *of a subdomain $d = (x, y; x', y')$ by displacement $(a, b) \in \mathbb{Z}^2$ is the subdomain $d = (x + a, y + b; x' + a, y' + b)$.*

We now introduce the central concepts of *regional language*, *tile*, and *local language*. The adjective "regional" is a metaphor of geographical political maps, such that different regions are filled with different colors.

Definition 3. *A* homogeneous partition *of a picture p is any partition π of $\mathrm{dom}(p)$ into homogeneous subdomains such that adjacent subdomains have different labels.*

A homogeneous partition is regional *(HR) iff distinct subdomains have distinct labels. We will call a picture p* regional *if it admits a HR partition.*

A language is regional *if all its pictures are so.*

We observe that if a picture p admits a homogeneous partition of $\mathrm{dom}(p)$ into subdomains, then the partition is unique and will be denoted by $\Pi(p)$.

Definition 4. *We call* tile *a square picture of size (2,2). We denote by $[\![p]\!]$ the set of all tiles contained in a picture p.*

Let Σ be a finite alphabet. A (two-dimensional) language $L \subseteq \Sigma^{++}$ is local *if there exists a finite set θ of tiles over the alphabet $\Sigma \cup \{\#\}$ such that $L = \{p \in \Sigma^{++} \mid [\![\hat{p}]\!] \subseteq \theta\}$. We will refer to such language as $LOC(\theta)$.*

The right parts of the rules presented in Sect. 3.1 are examples of regional local languages. Next, we characterize the form of tiles occurring in a regional local language.

Consider a tile set θ over the alphabet $\Sigma \cup \{\#\}$. For a tile $t = \begin{pmatrix} x & y \\ z & w \end{pmatrix}$ we define the *horizontal and vertical adjacency relations* $\mathcal{H}_t, \mathcal{V}_t \subseteq (\Sigma \cup \{\#\})^2$ as

$$x\mathcal{H}_t y,\ z\mathcal{H}_t w,\ x\mathcal{V}_t z,\ y\mathcal{V}_t w$$

The adjacency relation is $\mathcal{A}_t = \mathcal{H}_t \cup \mathcal{V}_t$.

The relations can be extended to a tile set θ: $x\mathcal{H}_\theta y$ iff $\exists t \in \theta : x\mathcal{H}_t y$; and similarly for $\mathcal{V}_\theta$ and $\mathcal{A}_\theta$.

Proposition 1. *The local language defined by a tile set θ is regional if*

1. *the (finite) language $\theta \cap \Sigma^{(2,2)}$ is regional, and*
2. *the incidence graph of $\left(\mathcal{A}_\theta \cap \Sigma^2\right) \setminus \mathcal{I}$, where $\mathcal{I}$ is the identity relation, is acyclic.*

3 Regional tile grammars

We are going to introduce and study a grammar model specified by a set of rewriting rules. A typical rule has a left and a right part, both pictures of unspecified but equal size (isometric). The left part is an A-homogeneous picture, where A is a nonterminal symbol. The right part is a picture of a regional local language over nonterminal symbols. Thus a rule is a scheme defining a possibly unbounded number of isometric pairs: left picture, right picture. In addition there are rules whose right part is a single terminal.

Notice that regional tile grammars may be viewed as extending CF grammars from one to two dimensions: see [2] for the argument that such grammars in one dimension are essentially CF grammars allowing a local regular expression in right parts of rules.

The derivation process of a picture starts from a S(axiom)-homogeneous picture. At each step, an A-homogeneous subpicture is replaced with an isometric picture of the regional language, defined by the right part of a rule $A \to \ldots$. The process terminates when all nonterminals have been eliminated from the current picture.[3]

Definition 5. *A* regional tile grammar (RTG) *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*.*
Let $A \in N$*. There are two kinds of rules:*

$$\textit{Fixed size:}\quad A \to t, \quad \textit{where } t \in \Sigma; \tag{1}$$

$$\textit{Variable size:}\quad A \to \omega, \quad \omega \textit{ is a set of tiles over } N \cup \{\#\}, \tag{2}$$

$$LOC(\omega) \textit{ is a regional language.} \tag{3}$$

Picture derivation is defined as a relation between partitioned pictures.

Definition 6. *Consider a grammar* $G = (\Sigma, N, S, R)$*, let* $p, p' \in (\Sigma \cup N)^{(h,k)}$ *be pictures of identical size. Let* π*,* π' *be homogeneous partitions of* $\mathrm{dom}(p)$*, with* $\pi = \{d_1, \ldots, d_n\}$*. We say that* (p', π') derives in one step *from* (p, π)*, written*

$$(p, \pi) \Rightarrow_G (p', \pi')$$

iff, for some $A \in N$ *and for some rule* $\rho \in R$ *with left part* A*, there exists in* π *an* A*-homogeneous subdomain* $d_i = (x, y; x', y')$*, called* application area*, such that:*

- p' *is obtained substituting* $\mathrm{spic}(p, d_i)$ *in* p *with a picture* s*, defined as follows:*
  - *if* ρ *is of type (1), then* $s = t$*;*
  - *if* ρ *is of type (2), then* $s \in LOC(\omega)$*.*
- π' *is a homogeneous partition of* $\mathrm{dom}(p)$ *into the subdomains*

$$(\pi \setminus \{d_i\}) \cup \mathrm{transl}_{d_i}(\Pi(s))$$

[3] For brevity, this presentation focuses on nonterminal rules, thus excluding for instance that both terminal and nonterminal symbols are in the same right part. More concise and readable forms of rules should be used in applications.

where $\mathrm{transl}_{d_i}(\Pi(s))$ *is the translation by displacement* $(x-1, y-1)$ *(intuitively the position of* d_i *in* p*) of the subdomains of* $\Pi(s)$*, the homogeneous partition of* s*.*

We say that (q, π') derives from (p, π) *in* n *steps, written* $(p, \pi) \overset{n}{\Rightarrow}_G (q, \pi')$*, iff* $p = q$ *and* $\pi = \pi'$*, when* $n = 0$*, or there are a picture* r *and a homogeneous partition* π'' *such that* $(p, \pi) \overset{n-1}{\Longrightarrow}_G (r, \pi'')$ *and* $(r, \pi'') \Rightarrow_G (q, \pi')$*. We use the abbreviation* $(p, \pi) \overset{*}{\Rightarrow}_G (q, \pi')$ *for a derivation with a finite number of steps.*

Definition 7. *The* picture language *defined by a grammar* G *(written* $L(G)$*) is the set of* $p \in \Sigma^{++}$ *such that*

$$\Big(S^{|p|}, \mathrm{dom}(p)\Big) \overset{*}{\Rightarrow}_G (p, \mathcal{I})$$

where $\mathcal{I}$ *denotes the partition of* $\mathrm{dom}(p)$ *defined by single pixels. For short we also write* $S \overset{*}{\Rightarrow}_G p$*.*

If we drop the constraint (3), we obtain the more general model of *tile grammars* [2].

Definition 8. *A* tile grammar *(TG) is a tuple* (Σ, N, S, R) *as in Definition 5, but condition (3) is omitted.*

Clearly, the picture derivation process for TG and RTG is the same. Notice that a derivation is defined iff the picture admits a homogeneous partition (see [2] for details). What makes the difference between Definition 5 and Definition 8 is that in the former the homogeneous partition is regional.

To illustrate, we now list some examples that will be reconsidered in Sect. 5 to separate the family RTG from other ones.

3.1 Regional tile grammars examples

Example 1. One row and one column of b*'s.*

The set of all pictures such that there is one row and one column (both not at the border) that hold b's, and the remainder of the picture is filled with a's.

$$S \to \begin{bmatrix} \# & \# & \# & \# & \# & \# & \# \\ \# & A_1 & A_1 & V_1 & A_2 & A_2 & \# \\ \# & A_1 & A_1 & V_1 & A_2 & A_2 & \# \\ \# & H_1 & H_1 & V_1 & H_2 & H_2 & \# \\ \# & A_3 & A_3 & V_2 & A_4 & A_4 & \# \\ \# & A_3 & A_3 & V_2 & A_4 & A_4 & \# \\ \# & \# & \# & \# & \# & \# & \# \end{bmatrix} ; \quad A_i \to \begin{bmatrix} \# & \# & \# & \# \\ \# & X & X & \# \\ \# & A_i & A_i & \# \\ \# & A_i & A_i & \# \\ \# & \# & \# & \# \end{bmatrix} \mid \begin{bmatrix} \# & \# & \# & \# \\ \# & X & X & \# \\ \# & \# & \# & \# \end{bmatrix}, \text{ for } 1 \le i \le 4$$

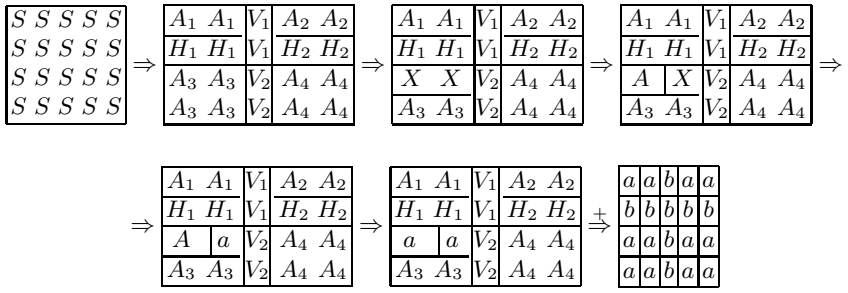
$$X \to \begin{bmatrix} \# & \# & \# & \# & \# \\ \# & A & X & X & \# \\ \# & \# & \# & \# & \# \end{bmatrix} \mid a; \quad H_i \to \begin{bmatrix} \# & \# & \# & \# & \# \\ \# & B & H_i & H_i & \# \\ \# & \# & \# & \# & \# \end{bmatrix} \mid b, \text{ for } 1 \le i \le 2$$

$$A \to a; \quad B \to b; \quad V_i \to \begin{bmatrix} \# & \# & \# \\ \# & B & \# \\ \# & V_i & \# \\ \# & V_i & \# \\ \# & \# & \# \end{bmatrix} \mid b, \text{ for } 1 \le i \le 2.$$

We recall that $[\![\ ]\!]$ denotes the set of tiles contained in the argument picture. This notation is more readable and concise than listing every tile:

$$S \to \left\{ \begin{matrix} \# & \# \\ \# & A_1 \end{matrix}, \begin{matrix} \# & \# \\ A_1 & A_1 \end{matrix}, \ldots, \begin{matrix} A_1 & V_1 \\ H_1 & V_1 \end{matrix}, \begin{matrix} V_1 & A_2 \\ V_1 & H_2 \end{matrix}, \ldots, \begin{matrix} A_4 & A_4 \\ \# & \# \end{matrix}, \begin{matrix} A_4 & \# \\ \# & \# \end{matrix} \right\}.$$

Here is an example of derivation, with partitions outlined for better readability:



Example 2. Picture with palindromic rows. Each row is an even palindrome over $\{a, b\}$.

$$S_P \to \left[\!\!\left[\begin{matrix} \# & \# & \# & \# \\ \# & R & R & \# \\ \# & S_P & S_P & \# \\ \# & S_P & S_P & \# \\ \# & \# & \# & \# \end{matrix} \right]\!\!\right] \mid \left[\!\!\left[\begin{matrix} \# & \# & \# & \# \\ \# & R & R & \# \\ \# & \# & \# & \# \end{matrix} \right]\!\!\right]$$

$$R \to \left[\!\!\left[\begin{matrix} \# & \# & \# & \# & \# & \# \\ \# & A & R & R & A' & \# \\ \# & \# & \# & \# & \# & \# \end{matrix} \right]\!\!\right] \mid \left[\!\!\left[\begin{matrix} \# & \# & \# & \# & \# & \# \\ \# & B & R & R & B' & \# \\ \# & \# & \# & \# & \# & \# \end{matrix} \right]\!\!\right] \mid \left[\!\!\left[\begin{matrix} \# & \# & \# & \# \\ \# & A & A' & \# \\ \# & \# & \# & \# \end{matrix} \right]\!\!\right] \mid \left[\!\!\left[\begin{matrix} \# & \# & \# & \# \\ \# & B & B' & \# \\ \# & \# & \# & \# \end{matrix} \right]\!\!\right]$$

$$A \to a; \quad B \to b; \quad A' \to a; \quad B' \to b.$$

Example 3. Misaligned palindromes.

A picture is a "ribbon" of two rows, divided into four fields: at the top-left and at the bottom right of the picture are palindromes as in Example 2 (where S_p is defined). The other two fields are filled with c's and must not be adjacent.

$$S \to \left[\!\!\left[\begin{matrix} \# & \# & \# & \# & \# & \# & \# & \# \\ \# & P_1 & P_1 & P_1 & P_1 & C_1 & C_1 & \# \\ \# & C_2 & C_2 & P_2 & P_2 & P_2 & P_2 & \# \\ \# & \# & \# & \# & \# & \# & \# & \# \end{matrix} \right]\!\!\right]; \quad P_1 \to S_P; \quad P_2 \to S_P$$

$$C_i \to \left[\!\!\left[\begin{matrix} \# & \# & \# & \# & \# \\ \# & C & C_i & C_i & \# \\ \# & \# & \# & \# & \# \end{matrix} \right]\!\!\right] \mid c, \text{ for } 1 \le i \le 2; \quad C \to c.$$

Procedure *Compute*$\mathfrak{D}$(i, j; k, l):
for each size $(v, h) \in \{1, \ldots, k-i+1\} \times \{1, \ldots, l-j+1\}$:

for each coordinate $(i', j') \in \{i, \ldots, k\} \times \{j, \ldots, l\}$:
for each nonterminal $A \in \mathfrak{M}(i', j'; i'+v-1, j'+h-1)$:
put $(i', j'; i'+v-1, j'+h-1)$ into the set $\mathfrak{D}(i, j; k, l)|_A$;

for each nonterminal $A \in N$:

if $\mathfrak{D}(i, j; k, l)|_A = \emptyset$ **then** put $(0, 0; 0, 0)$ into the set $\mathfrak{D}(i, j; k, l)|_A$.

Fig. 1. Compute$\mathfrak{D}$

4 Parsing for regional tile grammars

To present our version of the CKY algorithm, we have to generalize from substrings to subpictures. As a substring is identified by the positions of its first and last characters, a subpicture is conveniently identified by its subdomain.

Let p be a picture, of size (m, n), to be parsed with a grammar $G = (\Sigma, N, R, S)$.

Definition 9. *A* recognition matrix $\mathfrak{M}$ *is a 4-dimensional* $m \times n \times m \times n$ *matrix, whose generic element* $\mathfrak{M}(i, j; h, k)$ *is a set of non-terminals. The meaning of* $A \in \mathfrak{M}(i, j; h, k)$ *is that* A *can derive the subpicture* $\mathrm{spic}(p, (i, j; h, k))$ *of* p.

In fact, only cells $(i, j; h, k)$, with $h \geq i, k \geq j$, are used: these cells are the four-dimensional counterpart of the upper triangular matrix used in classical CKY.

Definition 10. *Consider a recognition matrix* $\mathfrak{M}$*, and a subdomain* $d = (i, j; k, l)$*. Let us order the nonterminal set* N*:* $A_1, A_2, \ldots, A_{|N|}$*. The* subdomains vector $\mathfrak{D}(d, \mathfrak{M})$ *is the cartesian product* $D_1 \times D_2 \times \ldots \times D_{|N|}$*, where every* D_t *is the set of subdomains* d' *such that* $N_t \in \mathfrak{M}(d')$ *and* d' *is a subdomain of* d*; if such set is empty, then* D_t *contains the rectangle* $(0, 0; 0, 0)$.

For any nonterminal A*, we will use the notation* $\mathfrak{D}(d, \mathfrak{M})|_A$ *to denote the component of the vector corresponding to* A.

To simplify the notation, we shall write $\mathfrak{D}(d)$ instead of $\mathfrak{D}(d, \mathfrak{M})$ at no risk of confusion, because the algorithm refers to a unique recognition matrix $\mathfrak{M}$. Figure 1 shows the procedure used to compute $\mathfrak{D}$.

Figure 2 shows the procedure to check if a rule ρ of the grammar can be applied to a given rectangle $(i, j; k, l)$.

The *Main* procedure, presented in Figure 3, is structured as a straightforward generalization to two dimensions of the CKY parsing algorithm. The input picture p is in $L(G)$ iff $S \in \mathfrak{M}(1, 1; m, n)$.

4.1 Correctness and complexity of parsing

We start with a technical lemma, used to prove the correctness of the CheckRule procedure.

Procedure *CheckRule* $(\omega, (i, j; k, l))$:
Compute$\mathfrak{D}(i, j; k, l)$;
for each $(d_1, d_2, \ldots, d_{|N|}) \in \mathfrak{D}(i, j; k, l)$:

$f := True$;
for each $(N_a, N_b) \in \mathcal{H}_\omega$:
if $d_a = (i_a, j_a; k_a, l_a)$ and $d_b = (i_b, j_b; k_b, l_b)$ are not such that
$j_b = l_a + 1$, and $k_b \geq i_a, k_a \geq i_b$,
then $f := False$;
for each $(N_a, N_b) \in \mathcal{V}_\omega$:
if $d_a = (i_a, j_a; k_a, l_a)$ and $d_b = (i_b, j_b; k_b, l_b)$ are not such that
$i_b = k_a + 1$, and $l_b \geq j_a, l_a \geq j_b$,
then $f := False$;
for each $(\#, N_a) \in \mathcal{H}_\omega$:
if $d_a = (i_a, j_a; k_a, l_a)$ and $j_a \neq j$ **then** $f := False$;
for each $(N_a, \#) \in \mathcal{H}_\omega$:
if $d_a = (i_a, j_a; k_a, l_a)$ and $l_a \neq l$ **then** $f := False$;
for each $(\#, N_a) \in \mathcal{V}_\omega$:
if $d_a = (i_a, j_a; k_a, l_a)$ and $i_a \neq i$ **then** $f := False$;
for each $(N_a, \#) \in \mathcal{V}_\omega$:
if $d_a = (i_a, j_a; k_a, l_a)$ and $k_a \neq k$ **then** $f := False$;
if $f = True$ **then return** $True$;

return $False$.

Fig. 2. CheckRule

Procedure *Main*:
Every set in $\mathfrak{M}$ is empty;
for each pixel $p(i, j) = t$,

if there exists a fixed size rule $A \to t \in R$,
then put A into the set $\mathfrak{M}(i, j; i, j)$;

for each size $(v, h) \in \{1, \ldots, m\} \times \{1, \ldots, n\}$:

for each coordinate $(i, j) \in \{1, \ldots, m\} \times \{1, \ldots, n\}$:
for each variable size rule rule $(A \to \omega) \in R$:
if *CheckRule*$(\omega, (i, j; i + v - 1, j + h - 1))$,
then put A into the set $\mathfrak{M}(i, j; i + v - 1, j + h - 1)$.

Fig. 3. Main

Lemma 1. *Let ω be a regional set of tiles and d a subdomain. CheckRule(ω, d) returns true iff there exists a rule $C \to \omega$, such that $(p_0, \pi_0) \Rightarrow_G (p_1, \pi_1)$, where $d \in \pi_0$, and $\mathrm{spic}(p_0, d)$ is a C-picture.*

Proof. By construction, a true output of CheckRule(ω, d) is equivalent to the fact that there exists a partition of d in the subdomains $d_1, d_2, \ldots, d_r$, and $q \in LOC(\omega)$, such that:

1. every $\mathrm{spic}(q, d_j)$ is an A-picture, for some nonterminal $A \in \mathfrak{M}(d_j)$;
2. if $\mathrm{spic}(q, d_j)$ is an A-picture, then for no $d_k \neq d_j$ the subpicture $\mathrm{spic}(q, d_k)$ is an A-picture.

This means that $\mathrm{transl}_d(\Pi(q))$ is the HR partition $\{d_1, d_2, \ldots, d_r\}$. Moreover, starting from (p_0, π_0), where $\mathrm{spic}(p_0, d)$ is a C-picture, it is possible to apply a rule $C \to \omega$ in a derivation step $(p_0, \pi_0) \Rightarrow_G (p_1, \pi_1)$, where $\pi_0 = \{d, d'_1, d'_2, \ldots, d'_n\}$, $\pi_1 = \{d'_1, d'_2, \ldots, d'_n\} \cup \{d_1, d_2, \ldots, d_r\}$, and $q = \mathrm{spic}(p_1, d) \in LOC(\omega)$. □

After this, the correctness is easy to prove, analogously to the 1D case [12].

Theorem 1. $\mathfrak{M}(d) = \{A \in N \mid A \overset{*}{\Rightarrow}_G \mathrm{spic}(p, d)\}$.

Proof. The proof is by induction over derivation steps.

Base: $d = (i, j; i, j)$. This means that $|\mathrm{spic}(p, d)| = (1, 1)$. Hence, $A \overset{*}{\Rightarrow}_G \mathrm{spic}(p, d)$ iff $A \to \mathrm{spic}(p, d) \in R$. This case is handled by the first loop of procedure Main, the one over each pixel $p(i, j)$. If $\mathrm{spic}(p, d) = t$, and there exists a rule $A \to t$, then the algorithm puts A into $\mathfrak{M}(d)$. Vice versa, $A \in \mathfrak{M}(d)$ means that the algorithm has put A in the set, therefore there must exist a rule $A \to \mathrm{spic}(p, d)$.

Induction: let us consider $d = (i, j; i+v-1, j+h-1)$, $v > 1$, or $h > 1$, or both. We prove that $A \overset{*}{\Rightarrow}_G \mathrm{spic}(p, d)$ implies $A \in \mathfrak{M}(d)$. In this case, the size of the subpicture is not $(1, 1)$, therefore the first rule used in the derivation $A \overset{*}{\Rightarrow}_G \mathrm{spic}(p, d)$ is a variable size rule $A \to \omega$. Thanks to the two nested loops with control variables u and v, when the algorithm considers d, it has already considered all its subdomains $d_1, d_2, \ldots, d_k$. By the induction hypothesis, for every $1 \leq j \leq k$, $B \overset{*}{\Rightarrow}_G \mathrm{spic}(p, d_j)$ implies $B \in \mathfrak{M}(d_j)$. Hence (Lemma 1), CheckRule(ω, d) must be true, and the algorithm puts A in $\mathfrak{M}(d)$.

Next, we prove that $A \in \mathfrak{M}(d)$ implies $A \overset{*}{\Rightarrow}_G \mathrm{spic}(p, d)$. $A \in \mathfrak{M}(d)$ means that procedure Main has put A in the set. Therefore, CheckRule(ω, d) must be true. Thanks to Lemma 1, this is equivalent to the existence of an applicable variable size rule $A \to \omega$ for the first step of the derivation $A \overset{*}{\Rightarrow}_G \mathrm{spic}(p, d)$. The rest of the derivation holds by induction hypotesis. □

Theorem 2. *The parsing problem for RTG has polynomial time complexity.*

Proof. First, it is straightforward to see the time complexity of procedure *Compute*$\mathfrak{D}$: $T_{\mathrm{Compute}\mathfrak{D}} = O(|N| \cdot m^2 n^2)$. Let us now consider the *CheckRule* procedure. After computing $\mathfrak{D}$, the procedure performs a loop for each element of the subdomains vector, and nested loops on $\mathcal{H}_\omega$ and $\mathcal{V}_\omega$. Therefore, $T_{\mathrm{CheckRule}}(m, n) = O(|N| \cdot m^2 n^2 \cdot \max_{A \to \omega \in R} \{|\mathcal{H}_\omega|, |\mathcal{V}_\omega|\})$.

Coming finally to the *Main* procedure, we note that its core part consists of five nested loops, two on sets of m elements, two on sets of n elements, and the last one on $|R|$ elements. The body is a call to CheckRule. Therefore, $T_{\mathrm{Main}}(m,n) = O\left(|R| \cdot m^2 n^2 \cdot T_{\mathrm{CheckRule}}(m,n)\right)$, i.e.

$$T_{\mathrm{Main}}(m,n) = O\left(|R||N| \cdot \max_{A \to \omega \in R}\{|\mathcal{H}_\omega|, |\mathcal{V}_\omega|\} \cdot m^4 n^4\right).$$

□

For the special case of CF Kolam grammars in Chomsky Normal form (CNF), we note that the parsing time complexity is $O(m^2 n^2 (m+n))$ [3]. Some of the reasons of this significant difference are the following. Kolam grammars in CNF are much simpler, because in the right part of a rule there are at most two distinct nonterminals (see [7] for details). So, checking if a rule is applicable has complexity which is linear with respect to the picture width or height. Moreover, we think that there is room for improvement e.g. in the CheckRule procedure, by using more complex data structures.

The next section will show that CF Kolam grammars are less expressive than RTG.

5 Comparison with other language families

The property of having polynomial time complexity for picture recognition, united with the rather simple and intuitively pleasing form of RTG rules, should make them a worth addition to the series of array rewriting grammar models conceived in past years. In this section we prove or recall some inclusion relations between grammar models and corresponding language families. To this end we rely on the examples of Sect. 3, and on the separation of complexity classes.

Starting with the family of highest generative capacity, we focus on tile grammars.

Proposition 2. *The family of RTG languages is a proper subset of the family of TG languages.*

Proof. We have seen in Sect. 3 that RTG rules are a restricted form of TG rules, characterized by the constraint of regional tiling. To show that inclusion is strict, we observe that the picture recognition problem for tile grammars is NP-complete. This follows from the (strict) inclusion [2] of the tiling systems (or Wang's tiling) [4] family within the TG language family, and the fact that the recognition problem is NP-complete in time for the former [6]. □

We proceed by comparing RTG's and tiling systems.

Proposition 3. *The family of tiling system languages (i.e. Wang's tiling) and the family of RTG languages are incomparable.*

Proof. On one hand, it is easy to see that the language of palindromic columns, used in [2] to prove that tiling systems are strictly included in tile grammars, is also a RTG language, obtained by a 90° rotation of Example 2. On the other hand, we know that parsing tiling systems is NP-complete, and parsing RTG's has polynomial time complexity. □

The remaining models are weaker than RTG, and will be taken in historical order.

Proposition 4. *The family of CF Kolam array grammar [11] (i.e. also [7]) languages is strictly included in the family of RTG languages.*

Proof. In [2] a construction is given to prove that a CF Kolam grammar (in the form defined by Matz [7]) can be transformed into a TG. It is easy to see that the construction used in the proof actually produces rules which satisfy the restriction of RTG's.

More directly, CF Kolam grammars in CNF can be seen as RTG's such that the tile-sets used in the right parts of rules must have one of the following forms:

Either $\left[\!\!\begin{array}{cccccc} \# & \# & \# & \# & \# & \# \\ \# & A & A & B & B & \# \\ \# & A & A & B & B & \# \\ \# & \# & \# & \# & \# & \# \end{array}\!\!\right]$, or $\left[\!\!\begin{array}{cccc} \# & \# & \# & \# \\ \# & A & A & \# \\ \# & A & A & \# \\ \# & B & B & \# \\ \# & B & B & \# \\ \# & \# & \# & \# \end{array}\!\!\right]$, or $\left[\!\!\begin{array}{cccc} \# & \# & \# & \# \\ \# & A & A & \# \\ \# & A & A & \# \\ \# & \# & \# & \# \end{array}\!\!\right]$, with $A \neq B$.

The inclusion is strict, because the language of Example 1 was shown by Matz [7] to trespass the generative capacity of his grammars. □

The fact that the picture recognition problem for CF Kolam grammars has been recently proved [3] to be polynomial in time of course follows from the above inclusion property and from Theorem 2.

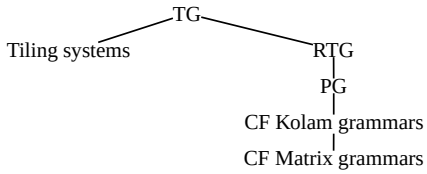
In the quest for generality, D. Průša [8] has recently defined a grammar model that extends CF Kolam rules, gaining some generative capacity. The model is for instance able to generate the language of Example 1.

Essentially, this kind of grammars can be seen as RTG's with the additional constraint that tiles used in the right parts of rules must not have one of these forms: $\begin{pmatrix} A & B \\ C & C \end{pmatrix}$, $\begin{pmatrix} A & C \\ B & C \end{pmatrix}$, $\begin{pmatrix} C & C \\ A & B \end{pmatrix}$, $\begin{pmatrix} C & A \\ C & B \end{pmatrix}$, with A, B, C all different. Therefore the following inclusion holds.

Proposition 5. *A Průša's grammar "with productions in CF form" (PG) [8] is a restricted kind of RTG. The corresponding family of languages is strictly included in the family of RTG languages.*

The inclusion of the language families is strict, because the language of Example 3 cannot be defined by PG's. This fact can be sketchily proved as follows.

Proof. First, an obvious application of CF pumping lemma for strings (over the alphabet $\{a \ominus a, a \ominus b, a \ominus c, b \ominus a, \ldots\}$) excludes that the language can be obtained by horizontal concatenation only. Therefore, it is necessary to generate the pictures either as vertical concatenations of strings of equal length, or using a grid-like rule, such as $\left[\!\!\begin{array}{cccccc} \# & \# & \# & \# & \# & \# \\ \# & A & A & B & B & \# \\ \# & C & C & D & D & \# \\ \# & \# & \# & \# & \# & \# \end{array}\!\!\right]$. By definition of the language, the two palindromes must span at least one common column, therefore we cannot use a simple vertical concatenation. The fact that $\{uu^R \mid u \in \{a, b\}^+\}$ cannot be factorized as a concatenation of CF languages is another simple corollary of the pumping lemma for CF languages. This means that each misaligned palindrome must be generated starting from a single nonterminal. But it is impossible for PG's to define a grid-like rule with single nonterminals partially overlapping on common columns. □



We finish with a synopsis of the previous language family inclusions. The early model of CF Matrix grammars [10] is a very limited kind of CF Kolam grammars.

6 Conclusions

The generalization of array rewriting grammars offered by the new and simple regional tile model is a convenient accomplishment of a series of generalizations, stemming from the early models of Rosenfeld up to the models of Siromoney, Matz and Průša. To our knowledge (but we may be missing something because the literature of picture grammars is rather fragmented), this is the most general family of polynomial time recognizable picture languages based on rewriting rules, which in one dimension collapse to CF string grammars.

Acknowledgement We thank the anonymous referees for helpful suggestions.

References

1. C. Allauzen and B. Durand. Tiling problems. In E. Borger and E. Gradel, editors, *The classical decision problem*. Springer-Verlag, 1997.
2. S. Crespi Reghizzi and M. Pradella. Tile Rewriting Grammars and Picture Languages. *Theoretical Computer Science*, 340(2):257–272, 2005.
3. S. Crespi Reghizzi and M. Pradella. A CKY parser for picture grammars. *Information Processing Letters*, 105(6):213–217, 2008.
4. 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*.
5. D. Giammarresi and A. Restivo. Two-dimensional languages. In Arto Salomaa and Grzegorz Rozenberg, editors, *Handbook of Formal Languages*, volume 3, Beyond Words, pages 215–267. Springer-Verlag, Berlin, 1997.
6. H. Lewis. Complexity of solvable cases of the decision problem for predicate calculus. In *Proc. 19th Symposium on Foundations of Computer Science*, pages 35–47, 1978.
7. O. 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, 1997.
8. D. Průša. Two-dimensional languages (PhD Thesis), 2004.
9. D. Simplot. A characterization of recognizable picture languages by tilings by finite sets. *TCS: Theoretical Computer Science*, 218, 1999.
10. G. Siromoney, R. Siromoney, and K. Krithivasan. Abstract families of matrices and picture languages. *Computer Graphics and Image Processing*, 1, 1972.
11. G. Siromoney, R. Siromoney, and K. Krithivasan. Picture languages with array rewriting rules. *Information and Control*, 23(5):447–470, 1973.
12. D. H. Younger. Recognition of context-free languages in time n^3. *Information and Control*, 10(2):189–208, 1967.