Picture Languages: from Wang tiles to 2D grammars*

Alessandra Cherubini[1] and Matteo Pradella[2]

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

Abstract. The aim of this paper is to collect definitions and results on the main classes of 2D languages introduced with the attempt of generalizing regular and context-free string languages and in same time preserving some of their nice properties. Almost all the models here described are based on tiles. So we also summarize some results on Wang tiles and its applications.

1 Introduction

The interest for a robust theory of two-dimensional (2D) languages (or picture languages) comes from the increasing relevance of pattern recognition and image processing. The main attempt of the research in this area is to generalize the richness of the theory of 1D languages to two dimensions. First focus was on definitions of classes of picture languages that are the analogue of the classes of Chomsky's hierarchy for 1D languages, in sense that, restricting to pictures of size $(1, n)$, picture and string languages at each level of the hierarchy coincide and that the new definitions for pictures inherit as many as possible properties from the corresponding definitions for strings.

Several different approaches were considered in the whole literature on the topic. The generalizations that seem to be the best answers to previous requests for the two lower levels of Chomsky's hierarchy are essentially based on Wang tiles and in this paper we aim to give a survey of classical and new results on these picture languages. Wang tiles, introduced in 1961, are squares whose all edges are colored. A finite set of Wang tiles admits a valid tiling of the plane if copies of the tiles can be arranged one by one, without rotations or reflections, to fill the plane so that all shared edges between tiles have matching colors. In 1966, Berger [8] proved that the problem of determining whether a given finite set of Wang tiles can tile the plane is undecidable, and constructed the first example of an aperiodic set of Wang tiles, i.e. a finite set of tiles whose all valid tilings have no periodic behavior. Several papers are devoted to the problem of determining small aperiodic set of Wang tiles but recently the main interest in Wang tiles was motivated by applications which, besides computer graphics, start to involve appealing areas in the frameworks of nanotechnologies and so called life sciences.

* Work partially supported by ESF *Automata: from Mathematics to Applications (AutoMathA)*, CNR RSTL 760 *Grammatiche 2D per la descrizione di immagini*, and by MIUR PRIN project *Mathematical aspects and emerging applications of automata and formal languages*.

For the ground level of Chomsky's hierarchy a robust definition of recognizable picture languages was proposed in 1991 by Giammarresi and Restivo. They defined the family REC of recognizable picture languages *by projection of local properties*, [31]. This class is considered *the* generalization of the class of regular 1D languages because it unifies several approaches to define the two dimension analogue of regular languages via finite automata, grammars, logic and regular expressions.

In 2005 Crespi Reghizzi and Pradella [18] introduced tile grammars, a model of grammars that extends the context-free (CF) grammars for 1D languages to two dimensions. The right hand part of each rule of a tile grammar is a set of tiles determining a local picture language. A rule is applied to the current picture replacing a rectangular subpicture, completely filled by the left hand side of the rule, with an isometric rectangle belonging to the local picture language determined by the right hand part of the rule. The generative power of these grammars exceeds REC languages. More recently a simplified version of tiling in the right hand part of the rules was considered in [15], giving raise to a new model of grammars called regional tile grammars. The new model includes several models of grammars proposed as generalizations of CF 1D grammars, the membership problem is solved by a polynomial time algorithm that naturally extends the classical CKY algorithm for strings, but it generates a family of languages incomparable with REC.

The first section of the paper contains some basic notions on pictures and picture languages. Then, some information on Wang tiles is given in second section, third and forth sections are devoted to collect results respectively on REC family and on several types of grammars proposed as generalization of CF 1D languages included in the family generated by tile grammars. In the last section, some open problems and some hints on different approaches to picture grammars are given.

2 Basic definitions

In this section some standard definitions of pictures, picture languages and operations on pictures are recalled.

Let Σ be a finite alphabet. A *picture* over Σ is a 2D array of elements of Σ called *pixels*. The *size* $|p|$ of a picture p is the pair $(|p|_{row}, |p|_{col})$ of its number of rows (its height) and columns (width). The indices grow from top to bottom for the rows and from left to right for the columns. The set of all pictures over Σ is denoted by $\Sigma^{+,+}$. $\Sigma^{*,*}$ is $\Sigma^{+,+} \cup \{\lambda\}$, where λ is the empty picture. For $h, k \geq 1$, $\Sigma^{h,k}$ (resp. $\Sigma^{h,+}$, $\Sigma^{+,k}$) is the set of all pictures of size (h, k) (resp. with h rows, with k columns). A *picture language* over Σ is a subset of $\Sigma^{*,*}$. # $\notin \Sigma$ is used when needed as a *boundary symbol*; $\hat{p}$ refers to the bordered version of picture p. That is, for $p \in \Sigma^{h,k}$, $\hat{p}$ is

$$\hat{p} = \begin{array}{ccccc} \# & \# & \cdots & \# & \# \\ \# & p(1,1) & \ldots & p(1,k) & \# \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \# & p(h,1) & \ldots & p(h,k) & \# \\ \# & \# & \cdots & \# & \# \end{array}$$

The *domain* of a picture p is the set $\mathrm{dom}(p) = \{1,\ldots,|p|_{row}\} \times \{1,\ldots,|p|_{col}\}$ and $\mathrm{dom}(\hat{p}) = \{0,\ldots,|p|_{row}+1\} \times \{0,\ldots,|p|_{col}+1\}$ is the domain of the bordered picture $\hat{p}$.

A *subdomain* of $\mathrm{dom}(p)$ is a set d of the form $\{x,\ldots,x'\} \times \{y,\ldots,y'\}$ where $1 \le x \le x' \le |p|_{row},\ 1 \le y \le y' \le |p|_{col}$; the size of d is $(x'-x+1, y'-y+1)$. We will often denote a subdomain by using its top-left and bottom-right coordinates, in the previous case the quadruple $(x,y;x',y')$ [3]. Subdomains of $\mathrm{dom}(\hat{p})$ are defined analogously. Each subdomain of $\mathrm{dom}(\hat{p})$ of size $(1,1)$ is called a *position* of p. 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 will write $d' = \mathrm{transl}_{(a,b)}(d)$. Pairs $(0,i), (|p|_{row}+1, i), (j,0), (j, |p|_{col}+1)$ with $0 \le i \le |p|_{col}+1,\ 0 \le j \le |p|_{row}+1$, are called *external positions* of p, the other are called *internal positions*. Positions in the set $\{(0,0), (0,|p|_{col}+1), (|p|_{row}+1, 0), (|p|_{row}+1, |p|_{col}+1)\}$ are called *corner positions*. Given a position (i,j) with $1 \le i \le |p|_{row}+1$ and $1 \le j \le |p|_{col}+1$ its *top-left-* (tl- for short) contiguous positions are the positions: $(i, j-1), (i-1, j-1), (i-1, j)$. Analogously for tr, bl, br where t, b, l, r are used for top, bottom, left and right respectively. For any internal position, its contiguous positions are all the tl-, tr-, br-, and bl-ones. Since each set $P(n,m) = \{0, 1\ldots, n+1\} \times \{0, 1\ldots, m+1\}$ can be seen as the domain of a bordered picture $\hat{p}$ with p of size (n,m), the elements of $P(n,m)$ are sometimes called positions of $P(n,m)$ as well.

The pixel of the picture p at position (i,j) of $\mathrm{dom}(p)$ is denoted $p(i,j)$. If all pixels of a picture p over Σ belong to an alphabet $\Sigma' \subseteq \Sigma$, p is called Σ'*-homogeneous*, a picture which is $\{a\}$-homogeneous for some $a \in \Sigma$ is called an a*-picture*, or also a homogeneous picture. If $a \in \Sigma$, $a^{h,k}$ stands for the a-picture in $\Sigma^{h,k}$, while $a^{+,+}$ stands for the set of a-pictures in $\Sigma^{+,+}$.

Let p be a picture over Σ and let $d = (x,y;x',y') \subseteq dom(p)$, the *subpicture* $\mathrm{spic}(p,d)$ associated to d is the picture of the same size of d such that, $\forall i \in \{1,\ldots,x'-x+1\}$ and $\forall j \in \{1,\ldots,y'-y+1\}$, $\mathrm{spic}(p,d)(i,j) = p(x+i-1, y+j-1)$. A subpicture q of p, written $q \trianglelefteq p$, is a subpicture $\mathrm{spic}(p,d)$ associated to some subdomain d of p. If $d = (x,y;x+h-1,y+k-1)$, then the subpicture $q = \mathrm{spic}(p,d)$ is also called the subpicture of p of size (h,k) at position (x,y), written $q \trianglelefteq_{(x,y)} p$.The *set of subpictures* of size (h,k) of p is denoted by

$$B_{h,k}(p) = \{q \in \Sigma^{h,k} : q \trianglelefteq p\}.$$

A picture $q \in \Sigma^{m,n}$ is called a *scattered subpicture* [4] of $p \in \Sigma^{+,+}$ if there are strictly monotone functions $f : \{1,2,\ldots,m\} \to \{n \in N \mid n \ge 1\}$, $g : \{1,2,\ldots,n\} \to \{n \in N \mid n \ge 1\}$ such that $p(f(i), g(j)) = q(i,j)$ for all $(i,j) \in \{1,2,\ldots,n\} \times \{1,2,\ldots,m\}$.

Now we shortly present main picture-combining and transforming operators.

The *column concatenation* $\oplus$, for all pictures p,q such that $|p|_{row} = |q|_{row}$, written $p \oplus q$, is defined as:

[3] Notice that the Cartesian coordinate system is clockwise rotated of 90^o with respect to the standard one.

[4] A scattered subpicture is often called a subpicture, and subpictures in our sense are called blocks.

$$p \oplus q = \begin{matrix} p(1,1) & \dots & p(1,|p|_{col}) & q(1,1) & \dots & q(1,|q|_{col}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ p(|p|_{row},1) & \dots & p(|p|_{row},|p|_{col}) & q(|q|_{row},1) & \dots & q(|q|_{row},|q|_{col}) \end{matrix}$$

The row concatenation $\ominus$ for pictures p, q, written $p \ominus q$, is defined analogously (with p on top). The empty picture λ is the neutral element for both concatenation operations. $p^{k\oplus}$ is the horizontal juxtaposition of k copies of p; $p^{*\oplus}$ is the corresponding closure. ${}^{k\ominus}$, and ${}^{*\ominus}$ are the row analogous.

The *projection by mapping* $\pi : \Sigma \to \Delta$ of a picture $p \in \Sigma^{+,+}$ is a picture $p' \in \Delta^{+,+}$ such that $|p| = |p'|$ and $p'(i,j) = \pi(p(i,j))$ for every position (i,j) of p.

The (clockwise) *rotation* of a picture p, $rot(p)$, is informally described as follows:

$$rot(p) = \begin{matrix} p(|p|_{row},1) & \dots & p(1,1) \\ \vdots & \ddots & \vdots \\ p(|p|_{row},|p|_{col}) & \dots & p(1,|p|_{col}) \end{matrix}$$

The *pixel-wise Cartesian product* of two pictures $p \in \Sigma_1^{*,*}, q \in \Sigma_2^{*,*}$ with $|p| = |q|$, is a picture $f \in (\Sigma_1 \times \Sigma_2)^{*,*}$ such that $|f| = |p|$, and $f(i,j) = (p(i,j), q(i,j))$ for all $i, j, 1 \le i \le |p|_{row}, 1 \le j \le |p|_{col}$ [50].

Projection, rotation, row and column concatenation, and pixel-wise Cartesian product can be extended to picture languages as usual. For every language $L \subseteq \Sigma^{*,*}$ we set $L^{0\oplus} = L^{0\ominus} = \lambda$, $L^{i\oplus} = L \oplus L^{(i-1)\oplus}$ and $L^{i\ominus} = L \ominus L^{(i-1)\ominus}$ for every $i \ge 1$. Thus, the row and column closures can be defined as the transitive closures of $\oplus$ and $\ominus$:

$$L^{*\oplus} = \bigcup_{i \ge 0} L^{i\oplus}, \; L^{*\ominus} = \bigcup_{i \ge 0} L^{i\ominus},$$

which can be seen as a sort of 2D Kleene star. In [50] Simplot introduced the closure L^{**}. We omit the detailed definition of Simplot's operator and introduce it quite informally. We say $p \in L^{++}$ iff there exists a partition of $\mathrm{dom}(p)$ where each subpicture associated to a subdomain of the partition is in L. Let L^{**} be the set $L^{++} \cup \{\lambda\}$. For example:

$$\begin{matrix} a & a & b \\ b & e & b \\ b & b & c \end{matrix} \in \left\{ a\ a\,, \begin{matrix} b \\ b \end{matrix}, b\ c\,, \begin{matrix} d \\ d \end{matrix}, e \right\}^{**}$$

If all the pictures of L have the same size, then $(L^{*\oplus})^{*\ominus} = (L^{*\ominus})^{*\oplus} = L^{**}$.

A well-known and widely useful concept in 1D languages is substitution, which assigns languages to letters of the alphabet and naturally extends to strings and languages too. In 2D languages, a substitution can be similarly defined. Given two finite alphabets Σ and Δ, a *substitution* from Δ to Σ is a mapping $\sigma : \Delta \to 2^{\Sigma^{+,+}}$. But a difficulty hinders the extension of the mapping to pictures, because of the so-called shearing problem of picture languages: a pixel in a picture cannot be replaced by a larger picture without disrupting the array structure. To overcome the problem in [15] the notion of replacement was introduced. If p, q, q' are pictures such that

$q \trianglelefteq_{(i,j)} p$ for some position (i,j) of p, and $|q| = |q'|$, then $p[q'/q]_{(i,j)}$ denotes the picture obtained by replacing the occurrence of q at position (i,j) in p with q', i.e., $p[q'/q]_{(i,j)}(i+x-1, j+y-1) = q'(x,y)$ for all $1 \leq x \leq |q|_{row}, 1 \leq y \leq |q|_{col}$. Then the notion of substitution was modified as follows. Let $\sigma : \Delta \to 2^{\Sigma^{+,+}}$ be a substitution. Given a picture $p \in \Delta^{+,+}$, a partition $\Pi(\mathrm{dom}(p)) = \{d_1, \ldots, d_n\}$, with $n \geq 1$, of $\mathrm{dom}(p)$ where each subpicture $\mathrm{spic}(p, d_m)$ associated to a subdomain d_m of the partition is a b_m-picture for some $b_m \in \Delta$ is called a *homogeneous partition* of p. Then the *substitution of* $p \in \Delta^{+,+}$ *induced by* $\Pi(\mathrm{dom}(p))$ is the language $\sigma_{\Pi(\mathrm{dom}(p))}(p) = \{p[r_1/\mathrm{spic}(p,d_1)] \ldots [r_n/\mathrm{spic}(p,d_n)] \mid r_m \in \sigma(b_m), 1 \leq m \leq n\}$. Given $L \subseteq \Sigma^{+,+}$, a set $\Pi = \{(p, \Pi(\mathrm{dom}(p)) \mid p \in L\}$, where each $\Pi(\mathrm{dom}(p))$ is a (homogeneous) partition of $p \in L$, is called a (homogeneous) partition set of L. If $L \subseteq \Delta^{+,+}$ and Π is a homogeneous partition set of L, then the *substitution of* L *induced by the homogeneous partition set* Π is the language $\sigma_\Pi(L) = \{\sigma_{\Pi(\mathrm{dom}(p))}(p) : p \in L\}$.
Roughly speaking a substitution $\sigma : \Delta \to 2^{\Sigma^{+,+}}$ extends to pictures and to picture languages by replacing a-subpictures p_a, at position (i,j), of p with pictures $q \in \sigma(a)$ of the same size. This definition, however, is not equivalent to the traditional notion of substitution when applied to strings.

Now we are in position of introducing families of 2D languages, but since we are mainly presenting languages based on tiling we remind some notions on Wang tiles.

3 Wang tiles

A *Wang tile* is a unit square with colored edges. Let T be a finite set of Wang tiles, which are not allowed to rotate. A map $\tau : \mathbb{Z}^2 \to T$ is called a *valid tiling*, of the Euclidean plane, or equivalently T can tile the Euclidean plane, if common edges of any pair of adjacent tiles have the same color. More formally denote by $N(t),\ S(t),\ W(t),\ E(t)$ the colors of the upper, lower, left and right edges of a tile t respectively, then τ is a valid tiling of the Euclidean plane, if $N(\tau(i,j)) = S(\tau(i,j+1)),\ S(\tau(i,j)) = N(\tau(i,j-1)),\ W(\tau(i,j)) = E(\tau(i-1,j))$, and $E(\tau(i,j)) = W(\tau(i+1,j))$, for each $(i,j) \in \mathbb{Z}^2$. Analogously, T can tile a rectangle of size $n \times m$ if there is a map $\tau : \{1,\ldots,m\} \times \{1,\ldots,n\} \to T$ such that adjacent tiles agree on the colors of contiguous edges. In 1961 Wang [53], analyzing the class of the first order formulas in prenex normal form whose prefix is $\forall x \exists y \forall z$, raised the question
Plane tiling problem *given a finite set of Wang tiles establish whether or not it admits a valid tiling.*

The 1D version of this problem admits an easy solution. Namely, to each finite set T of unary segments with colored left and right end points one can associate a direct graph where the set of colors is the set of vertices, and the edges (i,j) are the colors of left and right endpoints of some segment in T. Obviously T admits a valid tiling if and only if there is a bi-infinite path in the associate graph and then if and only if the graph has a loop. Coming back to the 2-dimensional problem, if the given finite set T of Wang tiles has a valid tiling with some vertical periodicity, the plane is covered by the repetition of some horizontal strip. Then, since this strip has only finitely many different vertical cross sections, the tiling has periodicity along two different directions.

A tiling τ is called *periodic* if there are two integers p, q such that $\tau(i, j) = \tau(i + p, j),\ \tau(i, j) = \tau(i, j + q)$ for all $(i, j) \in \mathbb{Z}^2$. Without loss of generality we can assume $p = q$. By the above argument it follows that if a finite set of Wang tiles has a tiling with a non zero period along one direction then it admits a periodic tiling.

Wang conjectured that any set of tiles which admits a valid tiling of the plane also admits a periodic tiling and under this assumption he gave an algorithm to solve the plane tiling problem, based on a compactness-like theorem.

Proposition 1. *A finite set of Wang tiles can tile the whole plane iff it can tile arbitrarily large finite areas of the plane.*

In particular a given set of tiles can tile the whole plane if and only if it can tile the first quadrant and so several constraints on the tiling of the first quadrant were posed. These problems were a bit easier to settle than the plane tiling problem and were speedily proved to be undecidable, an overview on these results can be found in [54]. The plane tiling problem on the contrary remained unsolved for years. However, from the above discussion it is clear that if the plane tiling problem is undecidable, then there are finite sets of tiles which admit only non-periodic tilings of the plane.

A finite set of Wang tiles which admits only non-periodic valid tiling is said *aperiodic*. In 1966 Berger [8], proved the following

Theorem 1. *The plane tiling problem is undecidable.*

His proof is based on encoding the halting problem of Turing Machine in the valid tiling of an arbitrary large square portion of the plane. Moreover, he constructed an aperiodic set of 20426 Wang tiles that shortly reduced to 104.

Then several well-known scientists from different areas as discrete mathematic, logic and computer science paid attention to the problem of finding smaller aperiodic sets of tiles and simplified proofs of undecidability of plane tiling problem (see for instance [49]). The smallest aperiodic set of Wang tiles obtained by geometrical arguments is composed by 16 tile (for a survey, see Chapters 10 and 11 of [33]). More recently Kari, [37], proposed a different approach based on sequential machines that multiply Beatty sequences of real numbers by rational constants, and produced an aperiodic set of Wang tiles with 14 tiles. His method was improved by Culik, [20], who built an aperiodic set formed by 13 tiles. This is currently the smallest known aperiodic set of Wang tiles. An expository article describing this approach is [27].

Once proved the existence of aperiodic set of Wang tiles, the following problem naturally arises:

Periodic tiling problem *given a finite set of Wang tiles determine whether or not it can tile the plane periodically.*

The problem was first studied in 1972 by Gurevich and Koriakov, who proved its undecidability [34].

Valid tilings have some quite surprising regularities. Let T be a finite set of Wang tiles, a *pattern* is a partial map $\varphi : P \rightarrow T$ from a finite domain P of $\mathbb{Z}^2$ in T. A pattern *appears* in a tiling $\tau : \mathbb{Z}^2 \rightarrow T$ if the tiling is the extension of the image of the pattern by a shift.

A valid tiling $\tau : \mathbb{Z}^2 \to T$ is called *quasi-periodic* if for each pattern M appearing in τ there is an integer n such that M appears in all $n \times n$ squares in τ. A valid quasi periodic tiling that is not periodic is called *strictly quasi-periodic*.

In [24] Durand proved the following

Theorem 2. *Each finite set of Wang tiles admitting a valid tiling admits a quasi-periodic valid tiling.*

The *quasi-periodicity* function for a quasi periodic tiling τ is the function that associate to each integer x the minimal size n of the squares in which one can find all the patterns of size x appearing in the tiling.

This function enables to characterize quasi periodic tilings that are periodic.

Proposition 2. *A quasi periodic tiling is periodic if and only if its quasi-periodicity function is bounded by $x \to x + c$, for some constant c .*

Then, using a counting argument on trees suitably associated to valid tilings, Durand obtains the following

Theorem 3. *If a tile set can be used to form a strictly quasi-periodic tiling of the plane, then it can form an uncountable number of different tilings.*

It is important to note that valid tilings could be defined in several different ways. For instance one could arrange all edge colors in complementary pairs and ask for tilings of the plane where common edges of adjacent tiles have complementary colors. This problem is equivalent to the plane tiling problem. If tile rotation is allowed, the tiling problem with matching colors of contiguous edges is trivially solvable while the problem with complementary colors remain undecidable.

A generalized simple way for describing variants of tiling rules is to consider the given finite set T of Wang tiles as a finite alphabet and a set of local rules $L \subseteq T^4$. A tiling τ satisfies the local rules L if and only if all 2×2 patterns appearing in the tiling are in L. In [26] the authors give via this approach a new short proof of the existence of aperiodic tilings.

Besides the strong connections with first order and description logics [25] yet arising from its original motivation, tiling problems have appeared in many branches of physics and mathematics like group theory, topology, quasicrystals, symbolic dynamics. More recently Winfree et al. [56] have demonstrated the feasibility of creating molecular tiles made from DNA that can act as Wang tiles introducing the *tile assembly model*. As pointed out by Brun [13] a tile assembly model is a highly distributed parallel model of computation that may be implemented using molecules, or a large computer network such as the Internet, and this opens several new prospectives.

In a more applicative and less ambitious context, Wang tiles have been proposed as tool for procedural synthesis of textures, and in general they have also proved to be very useful for the creation of large non-periodic textures, point-distributions and complex 2D scenes, see for instance [1, 17].

4 Recognizable picture languages

The attempt of transferring definitions and properties from string languages to their 2D analogue is quite successful when one considers the first level of Chomsky's hierarchy.

The class of picture languages corresponding to regular one- dimensional languages was intensively studied by several authors with different approaches: finite automata, logical characterizations, regular expressions and so on. An unifying approach to this family of picture language was proposed by Giammarresi and Restivo via local properties and projection. They introduced the so called REC family of picture languages and collected main properties of this family in the nice survey [31]. Here, besides summarizing the results contained in [31], we add some more recent results with the aim of fixing the actual state of art.

4.1 Labeled Wang tiles and Tiling Systems

First, we remind the definition of REC languages based on tiles endowed with labels in a finite alphabet Σ.

Definition 1. *([21]) A* labeled Wang tile*, shortly* LWT*, is a* 5*-tuple* (c_1, c_2, c_3, c_4, a) *where for all* $i,\ 1 \leq i \leq 4$*,* c_i *belongs to a finite set* C *of "colors" and* a *belongs to a finite set* Σ *of labels.*
A Wang system *(*WS*) is a triple* (C, Σ, T) *where* $T \subseteq C^4 \times \Sigma$ *is a finite set of* LWT*'s. Let* $B \in C$ *be a special color and let* r *be a picture of size* (n, m) *on the alphabet* T *,* r *is a* tiling *over* T *if*

- $r(1,1) \in \{(B, B, c_3, c_4, a) \mid c_3, c_4 \in C\backslash\{B\}, a \in \Sigma\}$*,*$r(1, n) \in \{(c_1, B, B, c_4, a) \mid c_1, c_4 \in C \setminus \{B\}, a \in \Sigma\}$*,* $r(m, n) \in \{(c_1, c_2, B, B, a) \mid c_1, c_2 \in C \setminus \{B\}, a \in \Sigma\}$*,*$r(m, 1) \in \{(B, c_2, c_3, B, a) \mid c_2, c_3 \in C \setminus \{B\}, a \in \Sigma\}$*;*
- *for all* $i,\ 1 < i < n$*,* $r(1, i) \in \{(c_1, B, c_3, c_4, a) \mid c_1, c_3, c_4 \in C \setminus \{B\}, a \in \Sigma\}$*,* $r(m, i) \in \{(c_1, c_2, c_3, B, a) \mid c_1, c_2, c_3 \in C \setminus \{B\}, a \in \Sigma\}$*;*
- *for all* $i,\ 1 < i < m$*,* $r(i, 1) \in \{(B, c_2, c_3, c_4, a) \mid c_2, c_3, c_4 \in C \setminus \{B\}, a \in \Sigma\}$*,* $r(i, n) \in \{(c_1, c_2, B, c_4, a) \mid c_1, c_2, c_4 \in C \setminus \{B\}, a \in \Sigma\}$*;*
- *for all* $(i, j),\ 1 \leq i \leq m, 1 \leq j \leq n$*,* $r(j, i) \in \{(c_1, c_2, c_3, c_4, a) \mid c_1, c_2, c_3, c_4 \in C \setminus \{B\}, a \in \Sigma\}$*; moreover let* $r(i, j) = (e, n, w, s, a)$*, then if* $i > 1$*,* $r(i-1, j) \in \{(c_1, c_2, c_3, n, a') \mid c_1, c_2, c_3 \in C, a' \in \Sigma\}$*, if* $j > 1$*,* $r(i, j-1) \in \{(c_1, c_2, e, c_4, a') \mid c_1, c_2, c_4 \in C, a' \in \Sigma\}$*.*

The label $\|r\|$ *of a tiling* r *is a picture over* Σ *of size* $|r|$ *defined by*

$$\|r\|(i, j) = a \Leftrightarrow r(i, j) = (c_1, c_2, c_3, c_4, a)$$

for some $c_1, c_2, c_3, c_4 \in C$*. The set of the labels of all the tilings over* T *is the language* $\mathcal{L}(\mathrm{WS})$ generated by the Wang system WS*. A language* L *generated by a Wang system is called* Wang recognizable.

For each LWT $t = (c_1, c_2, c_3, c_4, a)$ in a Wang system WS, consider the non labeled version $\widetilde{t} = (c_1, c_2, c_3, c_4)$. Roughly speaking the above definition says that the map $\rho : \{1, \ldots, m\} \times \{1, \ldots, n\} \rightarrow T$ defined as $\rho(h, k) = r(n + 1 - h, k)$ is a valid tiling of the region $\{1, \ldots, m\} \times \{1, \ldots, n\}$ by the set $\widetilde{\mathrm{WS}}$ of the non labeled versions of tiles in WS such that the boundary of the tiling r is colored by the special color B that does not occur in inner edges.

The same family of picture languages is also introduced by a formalism based on the local rules introduced in Section 3.

For $p \in \Sigma^{+,+}$ let $[\![p]\!]$ be the set of subpictures of size (2,2) of p.[5] In the sequel the concepts of *tile*, and *local language* are central.

Definition 2. *A* tile *is a square picture of size (2,2). A 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 $\mathrm{LOC}(\Theta)$.*

Notice that $\mathrm{LOC}(\Theta)$ is the set of finite rectangles of Euclidean plane with boundary colored by $\#$ that admit a valid tiling agreeing also with the boundary color. The set of local languages, shortly denoted by LOC, is the natural extension of string local languages and so the following definition extends one of the definitions of regular 1D languages.

Definition 3. *([31]) A* tiling system *(TS) is the 4-tuple $T = (\Sigma, \Gamma, \Theta, \pi)$, where:*
Σ and Γ are two finite alphabets,
$\pi : \Gamma \to \Sigma$ is a mapping,
Θ is a finite set of 2×2 tiles over the alphabet $\Gamma \cup \{\#\}$.
The language $L(T) = \pi(LOC(\Theta))$ is the language defined by the TS T.
The languages over finite alphabets defined by tiling systems constitute the family REC *of* TS-recognizable languages *on Σ.*

The family REC is considered the correct answer to the quest of a natural adaptation of the class of regular word languages for pictures. Namely, like in the 1D case, REC languages can be equivalently characterized by several formalisms. We shortly remind some of them, and we mainly refer to [31] for more information.

First, one can modify the size of tiles. In this way the definition of *domino systems* arises where Θ is a finite set of 1×2 and 2×1 pictures over the alphabet $\Gamma \cup \{\#\}$ and $\mathrm{LOC}(\Theta) = \{p \in \Sigma^{**} \mid B_{1,2}(\hat{p}) \cup B_{2,1}(\hat{p}) \subseteq \Theta\}$. A local language of this type is called hv*-local language*. The family of hv-local languages is properly included in LOC.

Moreover, one can consider the connection between Wang tiles and local rules.

Lastly, a characterization of REC in term of regular string languages can be given using the so called *row-column combination* of two string languages R and C, i.e. the languages $R \oplus C$ of the pictures all whose rows, thought as strings, are in R and whose all columns, seen as string from top to bottom, are in C.

Theorem 4. *([50, 21]) Let L be a picture languages. The following are equivalent.*

1. *L is TS-recognizable,*
2. *L is recognizable by some domino system,*
3. *L is Wang recognizable,*
4. *there exist two regular string languages R and C and a projection π such that $L = R \oplus C$.*

[5] In the rest of the paper, we will use this notation instead of $B_{2,2}(p)$ for brevity.

Other generalizations of local languages given in 1D case can be extended to picture languages.

Let h, k be two positive integers. Two pictures $p, r \in \Sigma^{*,*}$ are related in the equivalence relation $\cong_{h,k}$ if and only if their corresponding bordered versions have the same set of subpictures of size (h, k). A picture language is *locally testable* if it is union of $\cong_{h,k}$-equivalence classes for some positive integers $h,\ k$.

Let p be a picture. For h, k, t positive integer and for a picture $q \in (\Sigma \cup \{\perp\})^{*,*}$ of size (h, k) let $occ_p(q)$ the number of subdomains d of $\mathrm{dom}(p)$, such that $spic(p, d)$ is a translation of q and let $occ^t_p(q) = min(t, occ_p(q))$. Let $\cong^t_{h,k}$ be the equivalence relation on $\Sigma^{*,*}$ defined by $p \cong^t_{h,k} r$ if and only if $occ^t_p(q) = occ^t_r(q)$ for all $q \in (\Sigma \cup \{\perp\})^{*,*}$ of size (h, k) .

A picture language is *locally threshold testable* if it is union of $\cong^t_{h,k}$-equivalence classes for some positive integers $h,\ k$ and t.

Above picture languages are proper subclasses of REC.

Proposition 3. *The family LT of locally testable languages is properly included in the family LTT of locally threshold testable languages, which in turn is properly contained in* REC. *Moreover every language in LTT is a projection of a locally testable language.*

The family REC inherits several closure properties of regular string languages. Namely REC is closed under intersection, union, projection, row and column concatenation, closure operations, Cartesian product, and Simplot closure operator **. Moreover REC is closed under substitution of languages in REC induced by homogeneous partition sets, and also under by substitutions of languages in REC induced by the set of all homogeneous partitions of each picture [15].

However, fundamental properties of regular string languages fail in REC.

Proposition 4. REC *is not closed under complement.*
The membership problem for each language L in REC *is NP-complete.*
The emptiness and universe problems for REC *are undecidable.*

It is important to remark that in spite of its NP-completeness, the parsing problem for REC languages can be successfully tackled encoding the problem into SAT. Namely, in [45] a recognizer/generator for pictures defined by a tiling system is implemented in a very attractive, unconventional way, by considering for a picture p and each $a \in \Sigma$ the statement $p(i, j) = a$ as a propositional variable of the SAT problem and transforming the tiling problem into a Boolean satisfiability one, then using an efficient off-the-shelf SAT-solver. The prototype is fast enough to experiment on reasonably sized samples, and has the bonus of being able to complete a partial picture, by assigning to unknown pixels some values which satisfy the picture specification.

Another difference between regular string languages and REC arises considering the following modified definition of local testability. Let h, k be two positive integers. Two pictures are related in the equivalence relation $\sim_{h,k}$ if and only if they have the same set of scattered subpictures of size (h, k).

A picture language is *piecewise locally testable* if it is union of $\sim_{h,k}$-equivalence classes for some positive integers $h,\ k$. The language $\mathrm{CORNERS}$ of pictures p over

$\{a, b\}$ such that whenever $p(i, j) = p(i', j) = p(i, j') = b$ then also $p(i', j') = b$ is piecewise testable, but does not belong to REC.

4.2 Unambiguous and deterministic classes of recognizable picture languages

The definition of recognizability in terms of local languages and projections is implicitly nondeterministic, moreover since REC family is not closed under complement, each attempt to overcome its non-determinism gives smaller families of languages, differently of what happens for regular string languages.

We remind the definition of unambiguous REC languages given in [30].

Definition 4. *A quadruple* $(\Sigma, \Gamma, \Theta, \pi)$ *is an* unambiguous tiling system *for a 2D language* $L \subseteq \Sigma^{*,*}$ *if and only if for any picture* $p \in L$ *there exists a unique local picture* $q \in \mathrm{LOC}(\Theta)$ *such that* $p = \pi(q)$*, i.e. the extension of* π *to a map from* $\Gamma^{*,*}$ *to* $\Sigma^{*,*}$ *is injective on* $\mathrm{LOC}(\Theta)$.
$L \in REC$ *is an* unambiguous picture language *if and only if it admits an unambiguous tiling system* $(\Sigma, \Gamma, \Theta, \pi)$.

The family of all unambiguous REC picture languages is denoted by UREC.

The language of pictures with at least two equal columns is in REC, but not in UREC. Hence

Theorem 5. *([5])* UREC *is strictly included in* REC.

The notion of determinism for tiling systems has to be referred to a direction, like in 1D case. The considered direction is one of the four main directions from a corner to another ($c2c$).

Definition 5. *A tiling system* $(\Sigma, \Gamma, \Theta, \pi)$ *is* tl2br-deterministic [6] *if for any* $\gamma_1, \gamma_2, \gamma_3 \in \Gamma \cup \{\#\}$ *and* $\sigma \in \Sigma$ *there exists at most one tile* $t \in \Theta$ *with* $t = \begin{smallmatrix} \gamma_1 & \gamma_2 \\ \gamma_3 & \gamma_4 \end{smallmatrix}$*, and* $\pi(\gamma_4) = \sigma$. *Similarly* d*-deterministic tiling systems for any direction* $d \in c2c$ *are defined.*
$L \in REC$ *is a* deterministic picture language *if and only if it admits a deterministic tiling system for some* $d \in c2c$.

The family of all deterministic REC picture languages is denoted by DREC.

DREC is properly included in UREC and there are some classes of languages that strictly separate DREC from UREC. In [3] the classes of row-UREC and col-UREC are introduced (see also [29]) where four side-to-side scanning directions, namely left-to-right ($l2r$) and vice versa ($r2l$), top-to-bottom ($t2b$) and vice versa ($b2t$), are considered.

Definition 6. *A tiling system* $(\Sigma, \Gamma, \Theta, \pi)$ *is* $l2r$-unambiguous *if for any column* $col \in \Gamma^{m,1} \cup \{\#\}^{m,1}$ *, and picture* $p \in \Sigma^{m,1}$*, there exists at most one local column* $col' \in \Gamma^{m,1}$ *such that* $\pi(col') = p$ *and* $[\![\{\#\}^{1,2} \ominus (col \oplus col') \ominus \{\#\}^{1,2}]\!] \subseteq \Theta$*. Similar properties define* d*-unambiguous tiling systems, for any side-to-side direction* d.
A language is column-unambiguous *if it is recognized by a* d*-unambiguous tiling system for some* $d \in \{l2r, r2l\}$ *and it is* row-unambiguous *if it is recognized by a* d*-unambiguous tiling system for some* $d \in \{t2b, b2t\}$. Col-UREC *is the class of column-unambiguous languages and* Row-UREC *the class of row-unambiguous languages.*

[6] tl2br means from the top left to the bottom right corner.

Proposition 5. *([3])* $\mathrm{DREC} \subset (\mathrm{Col\text{-}UREC} \cap \mathrm{Row\text{-}UREC}) \subset$
$\subset (\mathrm{Col\text{-}UREC} \cup \mathrm{Row\text{-}UREC}) \subset \mathrm{UREC}$.

More recently, Lonati and Pradella [38] introduced a new kind of determinism for tiles: given $(\Sigma, \Gamma, \Theta, \pi)$, the pre-image of a picture $p \in \Sigma^{*,*}$ is built by scanning p with a boustrophedonic strategy, that is a natural scanning strategy used by many algorithms on pictures and 2D arrays. More precisely, it starts from the top-left corner, scans the first row of p rightwards, then scans the second row leftwards, and so on.

Definition 7. *A tiling system $(\Sigma, \Gamma, \Theta, \pi)$ is snake-deterministic if Γ and Θ can be partitioned as $\Gamma = \Gamma_1 \cup \Gamma_2,\ \Theta = \Theta_1 \cup \Theta_2$, where*

- $(\Sigma, \Gamma, \Theta_1, \pi)$ *is tl2br-deterministic and for each tile $t \in \Theta_1$, $t(i,j) \in \Gamma_{3-i} \cup \{\#\}$,*
- $(\Sigma, \Gamma, \Theta_2, \pi)$ *is tr2bl-deterministic and for each tile $t \in \Theta_2$, $t(i,j) \in \Gamma_i \cup \{\#\}$ and not both $t(1,1), t(1,2)$ are $\#$.*

The closure under rotation of languages recognized by snake deterministic tiling-systems is denoted $\mathrm{Snake\text{-}DREC}$.

Proposition 6. *([38])* $\mathrm{Snake\text{-}DREC} = \mathrm{Col\text{-}UREC} \cup \mathrm{Row\text{-}UREC}$.

UREC is closed under projection, disjoint union, intersection and rotation, and it is not closed under row and column concatenation and under row and column closures. An open problem is whether UREC family is closed under complementation, it is also conjectured that if a REC language is not in UREC then its complement is not in REC. Some recent results in this direction by Anselmo and Madonia are included in this volume. The family DREC is closed under complement but it is not closed under union and intersection. Moreover by Definition 6 it immediately follows that it is decidable whether a given tiling system is d-deterministic for $d \in c2c$. It is also decidable whether a tiling system is column- or row-unambiguous while it is undecidable whether it is unambiguous.

We would like also remark that in [6] a new model of recognizable picture languages without frames surrounding the pictures was introduced, and the changes of properties under the framed vs unframed approaches were considered mainly focusing on determinism and unambiguity. It turns out that the frame surrounding the blocks provides additional memory that, besides enforcing size and content of the recognized pictures, produces unframed ambiguous languages that are unambiguous in REC.

4.3 Models of 2-dimensional finite automata

A tile system $(\Sigma, \Gamma, \Theta, \pi)$ is a natural generalization of non deterministic finite automata to the 2D case. To underlying the analogies, Matz in [42] suggested to consider $\Gamma = \Sigma \times Q$ for some finite set Q, and the projection map π as the map $\pi(a,q) = q$ for each $a \in \Sigma,\ q \in Q$. He calls Q decoration set to point out that element of Q do not correspond to the intuition behind the word “state”. Then to see the tile system as an automaton one could imagine to simultaneously “decorate” each pixel of the input picture p and to check the decorated input for local compatibility with the transition relation Θ. Also in [21] some analogies between Wang systems and finite automata were indicated. However neither tile systems nor Wang systems correspond to an effective procedure of recognition, namely when the membership of a picture p to a given REC language has to be checked, no scanning procedure of the picture p is proposed.

Several operational models have been proposed to recognize picture languages. Here we remind only four of them and we refer to [36] for a survey on different models of finite automata recognizing picture languages.
The first model, called *4-way finite automaton*, shortly $4\mathrm{FA}$, was proposed in 1967 by Blum and Hewitt [10]. It is an extension of 2-way finite automata for strings and allows the finite automaton to move in four directions: t, b, l, r (top, bottom, left, and right).

Definition 8. *([31]) A* $4\mathrm{FA}$ *is a* 7*-tuple* $\mathcal{A} = (\Sigma, Q, \{t, b, l, r\}, q_0, q_a, q_r, \delta)$*, where* Σ *is the input alphabet,* Q *is the set of states,* q_0, q_a, q_r *are three distinguished states, called initial, accepting and rejecting states,* $\delta : (Q \setminus \{q_a, q_r\}) \times \Sigma \to 2^{(Q \times \{t,b,l,r\})}$ *is the transition function.*

$\mathcal{A}$ can be seen as a finite control in Q reading the input picture. If $(q', d) \in \delta(q, a)$ for some $d \in \{t, b, l, r\}$, the automaton goes from the actual state q and the actual position (i, j) with $p(i, j) = a$ to the state q', and moves the reading head by one position according to the direction d. The automaton halts when it reaches either the state q_a or the state q_r. It recognizes a picture $p \in \Sigma^{*,*}$ if starting from the position $(1, 1)$ in the state q_0, it eventually reaches the state q_a, it is not needed that it reads all the pixels of p.

The *2-dimensional on-line tessellation automaton* ($2\mathrm{OTA}$) is a restricted type of 2-dimensional cellular automata, i.e. an array of cells all being in some state at any given time and operating in a sequence of discrete time steps. In $2\mathrm{OTA}$ each cell changes its state depending on the top and left neighbors. This model was introduced by Inoue and Nakamura in 1977 [35]. Here we remind the definition given in [31].

Definition 9. *A* $2\mathrm{OTA}$ *is a* 5*-tuple* $\mathcal{A} = (\Sigma, Q, I, F, \delta)$*, where* Σ *is the input alphabet,* Q *is the set of states,* $I \subseteq Q$*,* $F \subseteq Q$ *are the sets of initial and final states,* $\delta : Q \times Q \times \Sigma \to 2^Q$ *is the transition function.*

A run of $\mathcal{A}$ over a picture $p \in \Sigma^{*,*}$ associates a state to each position of p. At time $t = 0$ a state $q_0 \in I$ is associated to all positions of the first row and column of $\hat{p}$, then moving diagonally across the array, at time $t = k$, states are simultaneously associated to each position (i, j) of the picture with $i + j - 1 = k$, according to δ. The picture p is recognized by $\mathcal{A}$ if there is a run of $\mathcal{A}$ associating a final state to the position $(|p|_{row}, |p|_{col})$.

In 2007 Anselmo and al. [4] proposed *tiling automata* (TA for short) as an effective computational device whose transitions are given by a tiling system with a scanning strategy that uses a next-step function and a data structure to remember some of the local symbols associated to the already scanned positions of the input picture. It is evident that to handle the borders, the next-step function depends also from the size of the input picture.

Definition 10. *Let* $n, m \in \mathbb{N}$ *and* $P(n, m) = \{0, 1, \ldots, n + 1\} \times \{0, 1, \ldots, m + 1\}$.
A next-position function *for pictures is a computable partial function* $f : \mathbb{N}^4 \to \mathbb{N}^2$ *associating to a quadruple* (i, j, n, m)*, with* $(i, j) \in P(n, m)$ *a pair* $(i', j') \in P(n, m)$.

Let $v_1(n,m) = (i_0, j_0) \in P(n,m)$ *and put* $v_h(n,m) = f(v_{h-1}(n,m), n, m)$*, then the sequence* $V_{f,k}(n,m) = \{v_1(n,m), v_2(n,m) \dots, v_{k-1}(n,m)\}$ *is called the sequence of* visited positions *by* f *at step* k *with starting position* (i_0, j_0)*.*
A scanning strategy *is a next-position function S such that for any* $(n,m) \in \mathbb{N}^2$ *the sequence* $V_{S,(n+2)(m+2)+1}(n,m) = \{v_1(n,m), v_2(n,m) \dots, v_{(n+2)(m+2)}(n,m)\}$ *of visited positions by S at step* $(n+2)(m+2)+1$ *starting from a corner position of* $P(n,m)$ *satisfies:*

1) $V_{S,(n+2)(m+2)+1}(n,m)$ *is a permutation of* $P(n,m)$*.*
2) *for any* $k = 2, \dots, (n+2)(m+2)$*, the* tl*- (or* tr*-, or* bl*-, or* br*- resp.) contiguous positions of* $v_k(n,m)$ *(when defined) are all in* $V_{S,(n+2)(m+2)+1}(n,m)$*.*

Moreover if S satisfies condition

3) *for any* $k = 2, \dots, (n+2)(m+2)$*,* $v_k(n,m)$ *is a contiguous position of* $v_{k-1}(n,m)$ *provided that* $v_{k-1}(n,m)$ *is an internal position, otherwise if* $v_{k-1}(n,m)$ *is an external position also* $v_k(n,m)$ *is an external position;*

it is called a continuous *scanning strategy; if S satisfies condition*

4) $v_{(n+2)(m+2)}(n,m)$ *is a corner position,*

it is called a normalized *scanning strategy.*

For each next-position function there is at most one starting corner, verifying conditions 1 and 2 of Definition 10. Moreover property 3 avoids that two non-contiguous regions of a picture are both scanned during a scanning process and together with property 4 forbids the existence of holes in the picture during the scanning process. In [4] several examples of continuous normalized scanning strategies are given, showing the richness of possibilities in 2D case, and they produce, for suitable data structures, different definitions of tiling automata. Here we introduce a formal definition of tiling automata with a scanning strategy that follows a main $tl2br$-directed strategy, i.e. a strategy such that for any $(n,m) \in \mathbb{N}^2$ and for any k with $1 \leq k \leq (n+2)(m+2)$ contains the (defined) tl-contiguous positions of $v_k(n,m)$ in the set of visited position at step k starting from position $(0,0)$.

Definition 11. *([4]) A* tiling automaton *of type* $tl2br$ *is a* 4*-tuple* $\mathcal{A} = (\mathcal{T}, S, D_0, \delta)$ *where* $\mathcal{T} = (\Sigma, \Gamma, \Theta, \pi)$ *is a tiling system, S is a* $tl2br$*-directed scanning strategy,* D_0 *is the initial content of a data structure that supports operations* $state_1(D)$*,* $state_2(D)$*,* $state_3(D)$*,* $update(D, \gamma)$*, for* $\gamma \in \Gamma \cup \{\#\}$*, and* $\delta : (\Gamma \cup \{\#\})^3 \times (\Sigma \cup \{\#\}) \rightarrow 2^{(\Gamma \cup \{\#\})}$ *is a relation such that* $\gamma_4 \in \delta(\gamma_1, \gamma_2, \gamma_3, \sigma)$ *if* $\pi(\gamma_4) = \sigma$ *and* $\begin{smallmatrix}\gamma_1 & \gamma_2 \\ \gamma_3 & \gamma_4\end{smallmatrix} \in \Theta$*.*

Tiling automata of type d *for each corner to corner (*$c2c$*) direction* d *are similarly defined.*

The initial configuration of the tiling automaton $\mathcal{A}$ is (p, i, j, D_0), where p is a picture of size (n,m) and $(i,j) = v_1(n,m)$. From a configuration (p, h, k, D), $h, k \in \mathbb{N}$, the automaton moves to the configuration (p, h', k', D) if $S(h,k,n,m)$ is defined, $\gamma_4 \in \delta(state_1(D), state_2(D), state_3(D), p(h,k))$ for some $\gamma_4 \in \Gamma \cup \{\#\}$, $(h', k') = S(h,k,n,m)$ and D' is the content of the data structure after calling $update(D, \gamma_4)$. If

$S(h,k,n,m)$ is defined, and there is no $\gamma_4 \in \Gamma \cup \{\#\}$ such that $\gamma_4 \in \delta(state_1(D), state_2(D), state_3(D), p(h,k))$, $\mathcal{A}$ stops without accepting, while if $S(h,k,n,m)$ is not defined, $\mathcal{A}$ stops accepting p.

It is important to remind that this definition 11 refers to a tiling automaton with a given scanning strategy (of type tl2br), another scanning strategy produces a different type of tiling automaton, nevertheless the class of recognized languages is the same.

Another family of automata for dealing with REC family of languages was introduced in 2005 by Bozapalidis and Grammatikopoulou [12]. Their definition is in terms of *doubly ranked monoids*. A doubly ranked semigroup (DR-semigroup for short) is a doubly ranked set $M = (M_{m,n})$ endowed with two associative operations ⓗ : $M_{m,n} \times M_{m,n'} \rightarrow M_{m,n+n'}$, and ⓥ : $M_{m,n} \times M_{m',n} \rightarrow M_{m+m',n}$, called respectively horizontal and vertical multiplications, that are compatible to each other, i.e. $(a ⓗ a') ⓥ (b ⓗ b') = (a ⓥ b) ⓗ (a' ⓥ b')$, for all a, a', b, b' of suitable ranks. A DR-semigroup M with two sequences $e = (e_m)$ and $f = (f_n)$, with $e_m \in M_{m,0},\ f_n \in M_{0,n}$ such that $e_0 = f_0,\ e_m ⓥ e_n = e_{m+n}$, $f_m ⓗ f_n = f_{m+n}$, and $e_m ⓗ a = a ⓗ e_m = a,\ f_n ⓥ b = b ⓥ f_n = B$ for all a, b of suitable rank is called a doubly ranked monoid; $e,\ f$ are called respectively the horizontal and vertical units of M. Given a doubly ranked alphabet X the free DR-monoid generated by X is called $pict(X)$.

Given a non empty set Q a *quadripolic relation* over Q of rank (m,n) is an element of $2^{Q^m \times Q^n \times Q^m \times Q^n}$ and the set of all quadripolic relations over Q of rank (m,n) is denoted by $4Rel_{m,n}(Q)$. The doubly ranked set $4Rel(Q) = (4Rel_{m,n}(Q))$ can be structured as a DR-monoid, by defining the horizontal multiplication as follows: for each $R \in 4Rel_{m,n}(Q)$ and $S \in 4Rel_{m,n'}(Q)$, $R ⓗ S = \{(w_1, w_2, w_3, w_4) |$ $\exists u \in Q^m, v_2, v_4 \in Q^n,\ z_2, z_4 \in Q^{n'} : w_2 = v_2 z_2, w_4 = v_4 z_4,\ (w_1, v_2, u, v_4) \in R, (u, z_2, w_3, z_4) \in S\}$ and in dual way for the vertical multiplication. Let M and M' be two DR-monoids. A *morphism* from M to M' is a family of functions $\varphi_{m,n} : M_{m,n} \rightarrow M'_{m,n},\ m, n \in \mathbb{N}$, compatible with horizontal and vertical multiplication and units. Now we are in position of remind the following

Definition 12. *([12]) Let X be a finite doubly ranked set. A quadripolic automaton over X is a 5-tuple $\mathcal{A} = (Q, \delta, F_{West}, F_{Sud}, F_{Est}, F_{North})$ where Q is a finite set of states,$F_{West}, F_{Sud}, F_{Est}, F_{North}$ are subsets of Q, called the four poles of acceptance for $\mathcal{A}$, δ is a family of maps $\delta_{m,n} : X_{m,n} \rightarrow 4Rel_{m,n}(Q)$.*

Let $\overline{\delta} : pict(X) \rightarrow 4Rel(Q)$ be the morphism of DR-monoids uniquely extending δ and let $F_{m,n} = F^m_{West} \times F^n_{Sud} \times F^m_{Est} \times F^n_{North}$. A picture $p \in pict_{m,n}(X)$ is accepted by $\mathcal{A}$ if and only if $\overline{\delta}_{m,n}(p) \cap F_{m,n} \neq \emptyset$. $L(QA)$ denotes the family of languages recognized by a quadripolic automaton. It is clear that quadripolic automata are related to the description of REC via labeled Wang tiles. This allows an algebraic approach to recognizable languages that is presented in a paper by Bozapalidis and Grammatikopoulou included in the present volume.

The following theorem clarifies the reason behind the name REC given to the family of TS-recognizable languages.

Theorem 6. *([31, 4, 12]) Let L be a picture language. The following are equivalent:*

1. $L \in \mathrm{REC}$;

2. *$L \in \mathcal{L}(\mathrm{2OTA})$;*
3. *$L \in \mathcal{L}(\mathrm{TA})$;*
4. *$L \in \mathcal{L}(\mathrm{QA})$.*

On the other hand, the family of 4-way automata is not enough powerful to define REC.

Proposition 7. *([31]) $\mathcal{L}(\mathrm{4FA})$ is strictly included in* REC. *Moreover $\mathcal{L}(\mathrm{4FA})$ is not closed under row and column concatenation and closure operations, but it is closed under union and intersection.*

Some attempts of increasing the power of 4-way automata by endowing them with a bounded queue or a bounded stack did not produce satisfactory results [7].

The unambiguous versions of on-line tessellation (2-UOTA, for short) and tilings automata (UTA, for short), i.e. 2-dimensional on-line tessellation and tilings automata such that for any picture there is at most one accepting computation, recognize UREC family.

Automata described in Definitions 8, 9, 11 admit also their deterministic counterparts. In the sequel $\mathrm{4DFA}$, $\mathrm{2DOTA}$, DTA denote the families of deterministic 4-way, 2-dimensional on-line tessellation and tiling automata. They are less powerful than the corresponding non-deterministic automata. In the deterministic case the family of languages recognized by tiling automata depends on the chosen scanning strategy, so $\mathcal{L}(\mathrm{DTA})$ denotes the set of all languages recognized by a deterministic d-tiling automata for each scanning strategy in any direction $d \in c2c$ and $\mathrm{DREC} = \mathcal{L}(\mathrm{DTA})$. Moreover the family $\mathcal{L}(\mathrm{4DFA})$ of languages recognized by a deterministic 4-way automaton and the family $\mathcal{L}(\mathrm{2DOTA})$ recognized by some automaton in $\mathrm{2OTA}$ are not comparable as shown by examples in [35].

4.4 Regular expressions

One of the main results on regular string languages is Kleene's theorem that characterizes the family of languages recognized by finite automata in term of regular expressions. Such expressions can be analogously defined for picture languages.

Definition 13. *([31]) A* regular expression *on the alphabet Σ is defined recursively as follows:*

1. *$\emptyset$ and each $a \in \Sigma$ are regular expressions;*
2. *if α and β are regular expressions, also $\alpha \cup \beta$, $\alpha \cap \beta$, α^C, $\alpha \oplus \beta$, $\alpha \ominus \beta$, $\alpha^{*\oplus}$, $\alpha^{*\ominus}$ are so.*

Each regular expression over Σ denotes a picture language: $\emptyset$ and $a \in \Sigma$ denote respectively the empty language and the language formed by the unique picture of size $(1,1)$ with $p(1,1) = a$, $\alpha \cup \beta$, $\alpha \cap \beta$, $\alpha \oplus \beta$, $\alpha \ominus \beta$, denote the union, intersection, row and column concatenation of languages α and β; α^C, $\alpha^{*\oplus}$, $\alpha^{*\ominus}$ denote the complement, and Kleene's closures of language α.
A language $L \subseteq \Sigma^{*,*}$ is regular *if it is generated by a regular expression over Σ.*

It is an immediate consequence of the non closure of REC under complement that REC does not coincide with the class $\mathcal{L}(\mathrm{RE})$ of the languages denoted by regular expressions. Then it is quite natural to consider restricted sets of operators to be iteratively applied starting from empty language and languages formed by a single picture of size (1,1).

In [31] the following sets of operators are considered: $\mathcal{R}_1 = \{\cup, \cap, \oplus, \ominus, ^{*\oplus}, ^{*\ominus}\}$, $\mathcal{R}_2 = \{\cup, \cap, ^C, \oplus, \ominus\}$ and in [42] the set $\mathcal{R}_3 = \{\cup, \oplus, \ominus, ^{*\oplus}, ^{*\ominus}\}$ was added.

Regular expressions containing only operators in $\mathcal{R}_1$ are called *complement-free* and $\mathcal{L}(\mathrm{CFRE})$ is the class of languages generated by complement-free regular expressions. Regular expressions using only operators in $\mathcal{R}_2$ are called *star-free* and $\mathcal{L}(\mathrm{SFRE})$ is the class of languages they denote. $\mathcal{L}(\mathrm{CFRE})$ properly contains the family of hv-local languages, hence giving a Kleene-like theorem for picture languages modulo projection.

Theorem 7. *A picture language L is in REC if and only if it is the projection of a language in $\mathcal{L}(\mathrm{CFRE})$.*

Also the class $\mathcal{L}(\mathrm{SFRE})$, being closed under complement, does not coincide with REC. In [41] Matz proved that the language $\mathrm{CORNERS}$ belongs to $\mathcal{L}(\mathrm{SFRE})$ whereas it is not in REC so showing that $\mathcal{L}(\mathrm{SFRE})$, and more in general the family of languages denoted by regular expressions, and REC are incomparable. This results answers to some open problems in [31], Section 8.4. In [55] it is proved that the language CROSS of all pictures over $\{a, b\}$ containing $\begin{smallmatrix} a & b & a \\ b & b & b \\ a & b & a \end{smallmatrix}$ as subpicture is piecewise testable but does not belongs to $\mathcal{L}(\mathrm{SFRE})$ and obviously $\mathcal{L}(\mathrm{SFRE})$ is not contained in the family PT of piecewise locally testable languages because the inclusion fails for the analog string languages.

The family of languages denoted by a regular expression containing only operators in $\mathcal{R}_3$, but $\cap$, is called REG in [42]. It is a proper subfamily of $\mathcal{L}(\mathrm{CFRE})$ and, in spite of its low expressive power, some arguments (simplicity, polynomial membership problem, polynomial emptiness problem) suggesting that it could be a better analog of regular string languages, are sketched.

In [39] Matz proposed a more powerful type of regular expressions for picture languages, called *regular expressions with operators*. For instance, he considered the column concatenation of a given picture r to the left and to the right like individual objects: $r\oplus$ and $\oplus r$. He call this kind of objects operators and allows iteration over combinations of operators. If unrestricted, these operators can be combined to generate languages not in REC (e.g $ab((a\oplus)(\oplus b))^*$ denotes the language $\{a^i b^i | i > 0\}$); but under the natural constraints that an operator working on the left (resp. top) is never juxtaposed, united or intersected with an operator working on the right (resp. bottom), he showed that the power of these expressions does not exceed the family REC and is enough to denote the language of square. It remains an open problem whether regular expressions with operators exhaust REC-family.

More recently Anselmo and al. [2] proposed some new operations on pictures and picture languages with the aim of looking for a homogeneous notion of regular expressions that could extend more naturally the concept of regular expression of 1D languages. They focus on regular expressions on one-letter alphabet but, as they remark, this is a necessary and meaningful case to start since it corresponds to study the "shapes" of pictures: if a picture language is in REC then necessarily the language of its shapes is in REC. First they introduced *diagonal concatenation* of pictures, that starting from two pictures p, q over a one-letter alphabet $\{a\}$, respectively of size (n, m) and (n', m'), produces the picture over $\{a\}$ of size $(n+n', m+m')$, so enabling to express some relationship between the dimensions of the pictures. The regular expressions allowing only union, diagonal concatenation and its closure as operators, and the empty set, empty picture, and empty row and column as atomic expressions denote a family of languages over $\{a\}$, called $\mathcal{L}(\mathrm{D})$. It coincides with the languages of a-pictures whose dimensions belongs to some rational relation or equivalently can be recognized by some $4\mathrm{FA}$ automaton that moves only right and down. $\mathcal{L}(\mathrm{D})$ properly contains the class of languages over one letter alphabet belonging to $\mathcal{L}(\mathrm{CFRE})$ and is closed under intersection and complement. Then they consider the family of languages over one letter alphabet denoted by regular expressions whose operator set contains union, column, row and diagonal concatenations and their closures, getting again a family properly included in REC. So, in the attempt of capture all the shapes allowed by 1D REC languages, they defined new types of iteration operations, called *advanced stars*, that result much more powerful than the classical stars and also seem to constitute a more reasonable approach to the general case because the definitions of advanced stars admit obvious generalizations on larger alphabets.

4.5 Logic formulas

Let Σ be a finite set and consider the signature $\{S_1, S_2, \{P_a\}_{a\in\Sigma}\}$, where P_a are unary and $S_i,\ i = 1, 2$ binary relation symbols. Monadic second-order (shortly MSO) formulas on this signature using first-order variables $x, y, z, \ldots$ and second order variables $X, Y, Z \ldots$, are inductively built from atomic formulas $x = y$, $S_1(x, y)$, $S_2(x, y)$, $P_a(x), X(x)$ using Boolean connectives and quantifiers applicable to first and second order variables. A MSO formula where no second order variable is quantified is called a first-order (FO) formula. An existential monadic second order (EMSO) is a formula of the form $\exists X_1 \exists X_2 \ldots \exists X_r \phi$ where ϕ is a first-order formula.

A picture p over Σ can be represented by the structure $\underline{p} = (\mathrm{dom}(p),\ S_{p,1}, S_{p,2}, \{P_{p,a}\}_{a\in\Sigma})$ where $\mathrm{dom}(p) = \{1, \ldots, |p|_{row}\} \times \{1, \ldots, |p|_{col}\}$, $S_{p,1}, S_{p,2} \subset \mathrm{dom}(p) \times \mathrm{dom}(p)$ are two successor relations defined by $(i, j)S_{p,1}(i+1, j)$ for $1 \leq i < |p|_{row}, 1 \leq i \leq |p|_{col}$ and $(i, j)S_{p,2}(i, j + 1)$ for $1 \leq i \leq |p|_{row}, 1 \leq j < |p|_{col}$, $|\Sigma|$ and $P_{p,a} = \{(i, j)|p(i, j) = a\}$, with $a \in \Sigma$ gives the set of positions labeled by a.

Let $\phi(X_1, X_2, ..., X_t)$ be a formula where at most $X_1, X_2, ..., X_t$ are free variables and let $Q_1, Q_2, \ldots, Q_t$ be subsets of $\mathrm{dom}(p)$. Consider the interpretation with domain $\mathrm{dom}(p)$, where first order variables are positions and second order variables are sets of positions in $\mathrm{dom}(p)$, and in particular Q_i is the interpretation of X_i for $1 \leq i \leq t$, the predicates $S_1(x, y), S_2(x, y), P_a(x), X(x)$ are seen as $(x, y) \in S_{p,1},\ (x, y) \in S_{p,2},\ x \in P_{p,a}, x \in X$. Then

$$(\underline{p}, Q_1, Q_2, \ldots, Q_t) \models \phi(X_1, X_2, ..., X_t)$$

means that p satisfies ϕ in the above interpretation.

A sentence is a formula without free variables. Let ϕ a sentence on the signature $\{S_1, S_2, \{P_a\}_{a\in\Sigma}\}$, the picture language L defined by ϕ is the set of all pictures p such that $\underline{p} \models \phi$. A characterization of REC in term of logic formulas is the following

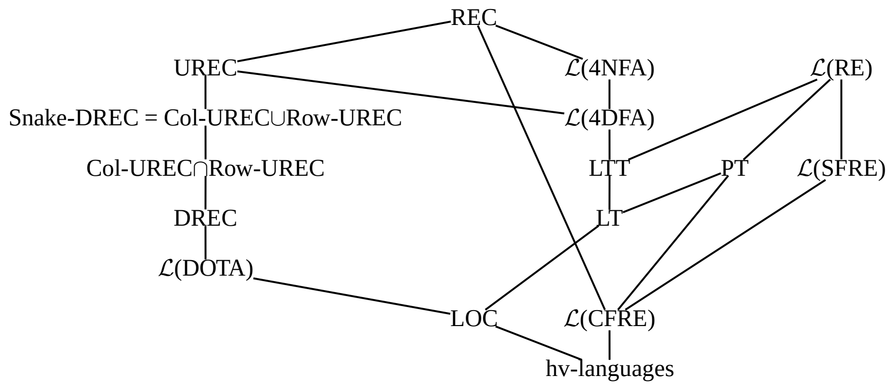
Theorem 8. *A picture language L is in* REC *if and only if it is definable by an* EMSO *formula in the signature* $\{S_1, S_2, \{P_a\}_{a\in\Sigma}\}$.

Matz in [41] enforces the above result showing that every picture language in REC is definable by an EMSO formula of the form $\exists X \phi(X)$ where ϕ is a first order formula.

Also, the families of languages with some kind of local testability admit logical characterization. In fact, a language is locally threshold testable iff it is definable by a first-order formula in the signature $\{S_1, S_2, \{P_a\}_{a\in\Sigma}\}$ ([32]), while is locally testable if and only if it is definable by a first-order formula in the signature $\{S_1, S_2, \{P_a\}_{a\in\Sigma}, left, right, top, bottom\}$, where $left, right, top, bottom$ are unary predicates saying that a position is at the respective border [40].

4.6 Summary

Inclusions of the families introduced in above sections are represented by the following diagram:



4.7 Necessary conditions for recognizability

An useful tool to prove whether a language is recognizable in 1D case is pumping lemma for regular languages. An analog of pumping lemma can be stated for languages in REC provided that they contain pictures whose number of columns (rows) is sufficiently larger than the number of rows (columns).

Lemma 1. *(Horizontal iteration lemma, [31]) Let $L \in REC$. Then there is a function $\varphi : \mathbb{N} \to \mathbb{N}$ such that if $p \in L$ and $|p|_{col} > \varphi(|p|_{row})$, there exist some pictures x, y, q with $|x \oplus q|_{col} \leq \varphi(|p|_{row})$ and $|y|_{col} > 1$ so that $p = x \oplus q \oplus y$ and for all $i \geq 0$ $x \oplus q^{i\oplus} \oplus y \in L$. Moreover, $\varphi(n) \leq |\Gamma|^n$ for any local alphabet used to represent L.*

Analogously can be stated a *vertical iteration lemma*.

Another necessary condition for a language being in REC uses the notion of syntactic equivalence modulo a language L. For a language $L \in \Sigma^{*,*}$ two isometric pictures $p,\ q$ are called *syntactically equivalent modulo* L (in symbols, $p \approx_L q$) if for all $x_1, x_2, y_1, y_2 \in \Sigma^{*,*}$ of suitable sizes, $x_1 \oplus (y_1 \ominus p \ominus y_2) \oplus x_2 \in L$ if and only if $x_1 \oplus (y_1 \ominus q \ominus y_2) \oplus x_2 \in L$. The function $f_L(|p|_{row}, |p|_{col})$ gives the number of $\approx_L$-equivalence classes in $\Sigma^{*,*}$ of size $(|p|_{row}, |p|_{col})$.

Lemma 2. *(Syntactic equivalence lemma, [31]) Let $L \in REC$. Then there exists a positive integer c such that $f_L(n, m) \leq c^{n+m}$ for all positive integers n, m.*

Lemma 3. *([40]) Let $L \in REC$ over Σ. For each positive integer n let $\{M_n\}$ be a sequence such that*

1. $M_n \subseteq \Sigma^{n,+} \times \Sigma^{n,+}$;
2. $\forall (p, q) \in M_n, p \oplus q \in L$;
3. $\forall (p, q), (p', q') \in M_n, \{p \oplus q', p' \oplus q\} \not\subseteq L$.

Then $|M_n|$ is $2^{O(n)}$.

The question of the existence of some language not in REC for which the above lemma fails to prove the non recognizability was posed. The language of squares over $\{a, b\}$ with as many a's as b's was proposed as candidate. However, from a result in [49], it follows that the above language is recognizable.

4.8 Recognizable picture languages on one-letter alphabet

Pictures over a one-letter alphabet, as already remarked in Section 4.4, are a special but meaningful case to consider. Only the shape of the picture is relevant, whence a unary picture is simply identified by a pair of positive integers representing its size. So a picture language over one letter alphabet can be studied looking to the corresponding set of integer pairs, and the definition of recognizability can be extended from languages to functions from $\mathbb{N}$ to $\mathbb{N}$ saying that a function $f : \mathbb{N} \to \mathbb{N}$ is recognizable if its associate language $L_f = \{p \in \{a\}^{*,*} \mid |p_{col}| = f(|p_{row}|)\}$ is recognizable. In [31] it is shown that recognizable functions cannot grow quicker than an exponential function or slower than a logarithmic one.

In 2007 Bertoni and al. [9] presented REC languages over one-letter alphabet via a characterization of strings encoding the pictures of the language. Namely they associate to each picture $p \in \{a\}^{*,*}$ the string $\phi(p) \in \{a, h, v\}^*$ defined as follows:
$\phi(p) = a^{|p|_{row}} h a^{|p|_{col} - |p|_{row} - 1}$, if $|p|_{row} < |p|_{col}$;
$\phi(p) = a^{|p|_{row}}$, if $|p|_{row} = |p|_{col}$;
$\phi(p) = a^{|p|_{col}} v a^{|p|_{row} - |p|_{col} - 1}$, if $|p|_{col} < |p|_{row}$.
Definition of ϕ obviously extends to languages by putting $\phi(L) = \{\phi(p) |\ p \in L\} \subseteq \{a, h, v\}^*$, for $L \subseteq \{a\}^{*,*}$.

Theorem 9. *Let $L \subseteq \{a\}^{*,*}$. L is in REC if and only if $\phi(L)$ is a string language that can be recognized by a 1-tape non-deterministic Turing machines working, for any input $x \in \{a, h, v\}^*$, within $|x|$ space and executing at most ${}_a|x|$ head reversals, where ${}_a|x|$ is the length of the longest prefix of x in a^+.*

Languages on one-letter alphabet were considered also for several of the afore-defined subclasses of REC languages.

5 Grammars for generating pictures

We did not consider generating grammars for REC family: in literature, 2D grammars are mainly considered as a way to introduce an analog of CF string languages, and several different models of grammars were proposed. There are essentially two main categories of picture grammars: one category imposes the constraint that the left and right parts of a rewriting rule must be isometric arrays, so overcoming the inherent problem of shearing (which pops up while substituting a subpicture in a host picture). The other one relies with several variations on notions of operations among pictures. More recently, to overcome the shearing problem and in general problems arising from the non flexibility of pixels in a picture, a picture deformation theory was introduced by Bozapalidis in [11]. A family of pixels $x^{(r,s)}$ is associated to any pixel x, called the (r,s)-deformed pixels of x, where r,s range over a semiring A. The deformation $p^{(r,s)}$ of a picture p is obtained by replacing all pixels of p by their (r,s)-deformations and is a picture where only the dimensions of p are changed.

In the following section a grammar model specified by a set of rewriting rules is presented with isometric rules. Then some properties of the model that seem to support the claim that the model is a good generalization of CF 1D languages are stated, and some relations with other well-known models of picture grammars are discussed.

5.1 Tile grammars

Tile grammars were defined in [18] with the name of tile rewriting grammars, then a normal form for those grammars was given in [14]. Here we use the normal form as basic definition because it is simpler to handle.

First we need to introduce the notion of *strong homogeneous partition*. We say that the domain of a picture p admits a strong homogeneous partition if there is a homogeneous partition of $dom(p)$ so that subpictures of p associated to contiguous subdomains have different labels. It is clear that each picture admits at most one strong homogeneous partition.

Definition 14. *A* Tile grammar (TG) *is a 4-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:*

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

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

The nonterminal symbol A in the left part of a variable size rule denotes an A-homogeneous picture. The right part of a variable size rule is a picture of a local language over nonterminal symbols. Thus a variable size 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 tile grammars may be viewed as extending CF grammars from one to two dimensions: the argument that such grammars in one dimension are essentially CF grammars allowing a local regular expression in right parts of rules is in [18].

The derivation process of a picture starts from a S-picture. Picture derivation is a relation between partitioned pictures.

Definition 15. *Consider a grammar $G = (\Sigma, N, S, R)$, let $p, p' \in (\Sigma \cup N)^{h,k}$ be pictures of identical size. Let $\pi = \{d_1, \ldots, d_n\}$ be homogeneous partition of $\mathrm{dom}(p)$. 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 $s \in LOC(\omega)$ and admits a strong homogeneous partition $\Pi(s)$*
- *π' is a homogeneous partition of $\mathrm{dom}(p)$ into the subdomains*

$$(\pi \setminus \{d_i\}) \cup \mathrm{transl}_{(x-1,y-1)}(\Pi(s))$$

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

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

Roughly speaking at each step of the derivation, an A-homogeneous subpicture is replaced with an isometric picture of the local language, defined by the right part of a rule $A \to \ldots$ that admits a strong homogeneous partition. The process terminates when all nonterminals have been eliminated from the current picture.

Definition 16. *The* picture language *defined by a grammar G (written $L(G)$) is the set of $p \in \Sigma^{+,+}$ such that*

$$\left(S^{|p|}, \mathrm{dom}(p)\right) \stackrel{*}{\Rightarrow}_G (p, \mathcal{I}),$$

where $\mathcal{I}$ denotes the partition of $\mathrm{dom}(p)$ defined by single pixels. For short we also write $S \stackrel{*}{\Rightarrow}_G p$. $L(TG)$ denote the family of languages generated by some tile grammar.

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

The set of 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 is defined by the tile grammar (we remind the reader that $[\![p]\!]$ stands for the set of all subpictures of size (2,2) of p):

$$S \to \left[\!\!\left[\begin{array}{ccccccc} \# & \# & \# & \# & \# & \# & \# \\ \# & A_1 & A_1 & V_1 & A_2 & A_2 & \# \\ \# & A_1 & A_1 & V_1 & A_2 & A_2 & \# \\ \# & H_1 & H_1 & V_1 & H_2 & H_2 & \# \\ \# & A_3 & A_3 & V_2 & A_4 & A_4 & \# \\ \# & A_3 & A_3 & V_2 & A_4 & A_4 & \# \\ \# & \# & \# & \# & \# & \# & \# \end{array}\right]\!\!\right]$$

$$A_i \to \left[\!\!\left[\begin{array}{cccc} \# & \# & \# & \# \\ \# & X & X & \# \\ \# & A_i & A_i & \# \\ \# & A_i & A_i & \# \\ \# & \# & \# & \# \end{array}\right]\!\!\right] \mid \left[\!\!\left[\begin{array}{cccc} \# & \# & \# & \# \\ \# & X & X & \# \\ \# & \# & \# & \# \end{array}\right]\!\!\right], \text{ for } 1 \leq i \leq 4$$

$$X \to \left[\!\!\left[\begin{array}{ccccc} \# & \# & \# & \# & \# \\ \# & A & X & X & \# \\ \# & \# & \# & \# & \# \end{array}\right]\!\!\right] \mid a; \quad H_i \to \left[\!\!\left[\begin{array}{ccccc} \# & \# & \# & \# & \# \\ \# & B & H_i & H_i & \# \\ \# & \# & \# & \# & \# \end{array}\right]\!\!\right] \mid b, \text{ for } 1 \leq i \leq 2$$

$$A \to a; \quad B \to b; \quad V_i \to \left[\!\!\left[\begin{array}{ccc} \# & \# & \# \\ \# & B & \# \\ \# & V_i & \# \\ \# & V_i & \# \\ \# & \# & \# \end{array}\right]\!\!\right] \mid b, \text{ for } 1 \leq i \leq 2.$$

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

$$\begin{array}{|ccccc|} \hline S & S & S & S & S \\ S & S & S & S & S \\ S & S & S & S & S \\ S & S & S & S & S \\ \hline \end{array} \Rightarrow \begin{array}{|cc|c|cc|} \hline A_1 & A_1 & V_1 & A_2 & A_2 \\ \hline H_1 & H_1 & V_1 & H_2 & H_2 \\ \hline A_3 & A_3 & V_2 & A_4 & A_4 \\ A_3 & A_3 & V_2 & A_4 & A_4 \\ \hline \end{array} \Rightarrow$$

$$\Rightarrow \begin{array}{|cc|c|cc|} \hline A_1 & A_1 & V_1 & A_2 & A_2 \\ \hline H_1 & H_1 & V_1 & H_2 & H_2 \\ \hline X & X & V_2 & A_4 & A_4 \\ \hline A_3 & A_3 & V_2 & A_4 & A_4 \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|c|cc|} \hline A_1 & A_1 & V_1 & A_2 & A_2 \\ \hline H_1 & H_1 & V_1 & H_2 & H_2 \\ \hline A & X & V_2 & A_4 & A_4 \\ \hline A_3 & A_3 & V_2 & A_4 & A_4 \\ \hline \end{array} \Rightarrow$$

$$\Rightarrow \begin{array}{|c|c|c|cc|} \hline A_1 & A_1 & V_1 & A_2 & A_2 \\ \hline H_1 & H_1 & V_1 & H_2 & H_2 \\ \hline A & a & V_2 & A_4 & A_4 \\ \hline A_3 & A_3 & V_2 & A_4 & A_4 \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|c|cc|} \hline A_1 & A_1 & V_1 & A_2 & A_2 \\ \hline H_1 & H_1 & V_1 & H_2 & H_2 \\ \hline a & a & V_2 & A_4 & A_4 \\ \hline A_3 & A_3 & V_2 & A_4 & A_4 \\ \hline \end{array} \stackrel{+}{\Rightarrow} \begin{array}{|c|c|c|c|c|} \hline a & a & b & a & a \\ \hline b & b & b & b & b \\ \hline a & a & b & a & a \\ \hline a & a & b & a & a \\ \hline \end{array}$$

The family $\mathcal{L}(\mathrm{TG})$ of TG-languages is closed w.r.t. union, column/row concatenation, column/row closure operations, rotation, alphabetic mapping ([18]).

We remark that this family as well as all families presented in the sequel, which exactly define CF string languages if restricted to one dimension, are not closed w.r.t. intersection and complement. Namely, since they are all closed w.r.t. union, the same arguments as string CF grammars can be used to prove these properties.

5.2 Tile grammars and tiling systems

Proposition 8. *([18])* $\mathrm{REC} \subset \mathcal{L}(\mathrm{TG})$.

In fact, for a tiling system $T = (\Sigma, \Gamma, \Theta, \pi)$, it is quite easy to define a TG T' such that $L(T') = L(T)$. Informally, the idea is to take the tile-set Θ and add two markers, e.g. $\{b, w\}$ in a "chequerboard-like" fashion to build up a tile-set suitable for the right part of the variable size starting rule; other straightforward fixed size rules are used to encode the projection π. We show the construction on a simple example. The interested reader may refer to [18] for details.

Example 2. The following TS defines square pictures of a's.

$$\Theta = \left[\!\!\left[\begin{array}{cccccc} \# & \# & \# & \# & \# & \# \\ \# & 1 & 0 & 0 & 0 & \# \\ \# & 0 & 1 & 0 & 0 & \# \\ \# & 0 & 0 & 1 & 0 & \# \\ \# & 0 & 0 & 0 & 1 & \# \\ \# & \# & \# & \# & \# & \# \end{array} \right]\!\!\right], \ \pi(0) = a, \ \pi(1) = a$$

An equivalent tile grammar is the following:

$$S \to \left[\!\!\left[\begin{array}{cccccc} \# & \# & \# & \# & \# & \# \\ \# & 1_b & 0_w & 0_b & 0_w & \# \\ \# & 0_w & 1_b & 0_w & 0_b & \# \\ \# & 0_b & 0_w & 1_b & 0_w & \# \\ \# & 0_w & 0_b & 0_w & 1_b & \# \\ \# & \# & \# & \# & \# & \# \end{array} \right]\!\!\right] \cup \left[\!\!\left[\begin{array}{cccccc} \# & \# & \# & \# & \# & \# \\ \# & 1_w & 0_b & 0_w & 0_b & \# \\ \# & 0_b & 1_w & 0_b & 0_w & \# \\ \# & 0_w & 0_b & 1_w & 0_b & \# \\ \# & 0_b & 0_w & 0_b & 1_w & \# \\ \# & \# & \# & \# & \# & \# \end{array} \right]\!\!\right]$$

$$1_w \to a, \ 1_b \to a, \ 0_w \to a, \ 0_b \to a.$$

To see that the inclusion is proper, one can restrict to string languages.

From above it immediately follows that the parsing problem for TG-languages is NP-hard, but in [44] it is proved that it is in NP, so

Proposition 9. *The parsing problem for* $\mathcal{L}(\mathrm{TG})$ *is NP-complete.*

In [15] some restrictions on tile grammars guaranteeing that the generated language is in REC are given. These restrictions are the analog of the restrictions that one dimensional CF grammars have to satisfy in order of defining regular languages.

Let $G = (\Sigma, N, S, R)$ be a tile grammar, a non terminal $A \in N$ is *non recursive* if and only if there is no derivation of the form $(A, \Pi) \Rightarrow^* (q, \Pi')$ with $spic(p, d) \in A^{+,+}$ for some subdomain d of Π'. Two non terminals $A_1, A_2 \in N$ are *mutually recursive* if and only if for each $i = 1, 2$ there are derivations $(A_i, \Pi_i) \Rightarrow^* (q_i, \Pi'_i)$ with $spic(q_i, d_i) \in \{A_{3-i}\}^{*,*}$ for some subdomain d_i of Π'_i. A tile grammar all whose non terminal are non recursive is called *non recursive tile grammar*.

Proposition 10. *([15]) The family of languages generated by non-recursive tile grammars coincides with* REC.

One can define a 2D analogous of a 1D grammar where self-embedding is never allowed.

Definition 17. *A tile grammar $G = (\Sigma, N, S, R)$ is a corner grammar if there exists a partition of N in sets N_1, N_2, N_3, N_4, and $\overline{N}$ such that:*

1. *$\overline{N}$ is the set of non-recursive nonterminals of G;*
2. *for every $i \neq j,\ 1 \leq i, j \leq 4$, for each $A \in N_i$, $B \in N_j$, A and B are not mutually recursive;*
3. *for every i, $1 \leq i \leq 4$, for each $A \in N_i$ if $A \Rightarrow^* p$ then p has a subpicture at i-th corner in $N_i^{*,*}$ and the remainder pixels in $\Sigma \cup (N \setminus N_i)$, where the i-th corner is lt for $i = 1$, rt for $i = 2$, rb for $i = 3$, lb for $i = 4$.*

In other words, in every non-corner position of a picture, only terminals or those nonterminals that cannot give rise to recursions are allowed, while disjoint (possibly empty) nonterminal alphabets are considered for the four corners. Clearly, a non-recursive tile grammar is a special case of corner grammar (with $N_i = \emptyset$ for every $i,\ 1 \leq i \leq 4$). A corner grammar is also a generalization of right-linear or left-linear grammars for the 1D case.

Proposition 11. *([15]) The family of languages generated by a corner grammars coincides with* REC*.*

Notice that checking whether a tile grammar is recursive or if it is a corner grammar is not decidable.

5.3 Regional tile grammars

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

Definition 18. *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.*

Definition 19. *([14]) A* regional tile grammar (RTG) *is a tile grammar (see Definition 14), in which every variable size rule $A \to \omega$ is such that $\mathrm{LOC}(\omega)$ is a* regional language.

We note that Example 1 is regional, while the picture language presented in Example 2 is not.

For languages generated by regional tile grammars a parsing algorithm generalizing the CKY algorithm is given. A subpicture is conveniently identified by its subdomain as in original algorithm a substring is identified by the positions of its first and last characters.

Theorem 10. *([14]) The parsing problem for* RTG *has polynomial time complexity.*

Analyzing the algorithm, one derives that the complexity of parsing for a picture of size (n, m) is $\mathcal{O}(\mu m^4 n^4)$ where constant μ depends on parameters of the grammar. The property of having polynomial time complexity for picture recognition, together with the remark that pictures with palindromic rules are not in REC immediately give the following results:

Proposition 12. *([14])* $\mathcal{L}(\mathrm{RTG}) \subset \mathcal{L}(\mathrm{TG})$. $\mathcal{L}(\mathrm{RTG})$ *is incomparable with* REC.

Moreover, the polynomial parsing 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 the sequel we prove or recall some inclusion relations between grammar models and corresponding language families.

5.4 Průša's grammars

The following definitions are taken and adapted from [46, 47].

Definition 20. *A 2D CF Průša grammar (*PG*) is a tuple* (Σ, N, R, S)*, where* Σ *is the finite set of* terminal *symbols, disjoint from the set* N *of* nonterminal *symbols,* $S \in N$ *is the* start *symbol, and* $R \subseteq N \times (N \cup \Sigma)^{+,+}$ *is the set of* rules.

Definition 21. *Let* $G = (\Sigma, N, R, S)$ *be a PG. We define a picture language* $L(G, A)$ *over* Σ *for every* $A \in N$*. The definition is given by the following recursive descriptions:*

(i) If $A \to w$ *is in* R*, and* $w \in \Sigma^{+,+}$*, then* $w \in L(G, A)$*.*
(ii) Let $A \to w$ *be a production in* R*,* $w = (N \cup \Sigma)^{(m,n)}$*, for some* $m, n \geq 1$*. Let* $p_{i,j}$*, with* $1 \leq i \leq m$*,* $1 \leq j \leq n$*, be pictures such that:*
  1. *if* $w(i,j) \in \Sigma$*, then* $p_{i,j} = w(i,j)$*;*
  2. *if* $w(i,j) \in N$*, then* $p_{i,j} \in L(G, w(i,j))$*;*
  3. *for* $1 \leq i < m$*,* $1 \leq j \leq n$*,* $|p_{i,j}|_{col} = |p_{i+1,j}|_{col}$*; let* $P_k = p_{k,1} \oplus p_{k,2} \oplus \cdots \oplus p_{k,n}$*, and* $P = P_1 \ominus P_2 \ominus \cdots \ominus P_m$*.*

  Then $P \in L(G, A)$*.*

The set $L(G, A)$ *contains just all the pictures that can be obtained by applying a finite sequence of rules (i) and (ii). The language* $L(G)$ *generated by the grammar* G *is defined as the language* $L(G, S)$*.*

Informally, rules can either be terminal rules, which are used to generate the pictures which constitute the right parts of rules, or have a picture as right part. In this latter case, the right part is seen as a “grid”, where nonterminals can be replaced by other pictures, but maintaining its grid-like structure.

Example 3. The following grammar generates the language of pictures with one row and one column of b's in a background of a's (see Example 1).

$$S \to \begin{matrix} A & V & A \\ H & b & H \\ A & V & A \end{matrix}, \quad A \to AM \mid M, \quad M \to \begin{matrix} a \\ M \end{matrix} \mid a,$$

$$V \to \begin{matrix} b \\ V \end{matrix} \mid b, \;\; H \to bH \mid b.$$

It is easy to see that Průša grammars admit a Nonterminal Normal Form:

Definition 22. *A Průša grammar $G = (\Sigma, N, R, S)$, is in* Nonterminal Normal Form *iff every rule in R has the form either $A \to t$, or $A \to w$, where $A \in N$, $w \in N^{+,+}$, and $t \in \Sigma$.*

To compare Průša's grammars with tile grammars, we must note that the two models are different in their derivations. Tile grammars start from a picture made of S's having a fixed size, and being every derivation step isometric, the resulting picture, if any, has the same size. On the other hand, PG's start from a single S symbol, and then "grow" the picture derivation step by derivation step, obtaining, if any, a usually larger picture.

Proposition 13. *([14])* $\mathcal{L}(\mathrm{PG}) \subset \mathcal{L}(\mathrm{RTG})$.

Remark 1. Essentially, Průša 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.

5.5 Kolam grammars

Průša introduced his model with the attempt of gaining some generative capacity with respect the class of Kolam grammars. This class of grammars has been introduced by Siromoney et al. [52] under the name "Array grammars", later renamed "Kolam Array grammars" in order to avoid confusion with Rosenfeld's homonymous model. Much later Matz reinvented the same model [39] (considering only CF rules). Here the historical name, CF Kolam grammars (CFKG) is kept, the more succinct definition of Matz is used.

Definition 23. *A* sentential form *over an alphabet V is a non-empty well-parenthesized expression using the two concatenation operators, $\ominus$ and $\oplus$, and symbols taken from V. $\mathcal{SF}(V)$ denotes the set of all sentential forms over V. A sentential form ϕ defines either one picture over V denoted by $(\!|\phi|\!)$, or none.*

For example, $\phi_1 = ((a \oplus b) \ominus (b \oplus a)) \in \mathcal{SF}(\{a, b\})$ and $(\!|\phi_1|\!)$ is the picture $\begin{smallmatrix} a & b \\ b & a \end{smallmatrix}$. On the other hand $\phi_2 = ((a \oplus b) \ominus a)$ denotes no picture, since the two arguments of the $\ominus$ operator have different column numbers.

CF Kolam grammars are defined analogously to CF string grammars. Derivation is similar: a sentential form over terminal and nonterminal symbols results from the preceding one by replacing a nonterminal with some corresponding right hand side of a rule. The end of a derivation is reached when the sentential form does not contain any nonterminal symbols. If this resulting form denotes a picture, then that picture is generated by the grammar.

Definition 24. *A* CF Kolam grammar (CFKG) *is a tuple $G = (\Sigma, N, R, S)$, where Σ is the finite set of* terminal *symbols, disjoint from the set N of* nonterminal *symbols; $S \in N$ is the* starting *symbol; and $R \subseteq N \times \mathcal{SF}(N \cup \Sigma)$ is the set of* rules*. A rule $(A, \phi) \in R$ will be written as $A \to \phi$.*

For a grammar G, we define the *derivation* relation $\Rightarrow_G$ on the sentential forms $\mathcal{SF}(N \cup \Sigma)$ by $\psi_1 \Rightarrow_G \psi_2$ iff there is some rule $A \to \phi$, such that ψ_2 results from ψ_1 by replacing an occurrence of A by ϕ. As usual, $\overset{*}{\Rightarrow}_G$ denotes the reflexive and transitive closure of $\Rightarrow_G$. Notice that the derivation thus defined rewrites strings, not pictures.

From the derived sentential form, one obtains the denoted picture. The picture language generated by G is the set

$$L(G) = \{(\!|\psi|\!) \mid \psi \in \mathcal{SF}(\Sigma), S \overset{*}{\Rightarrow}_G \psi\}.$$

With a slight abuse of notation, we will often write $A \overset{*}{\Rightarrow}_G p$, with $A \in N, p \in \Sigma^{*,*}$, instead of $\exists \phi : A \overset{*}{\Rightarrow}_G \phi, (\!|\phi|\!) = p$.

CF Kolam grammars admit a normal form with exactly two or zero nonterminals in the right part of a rule [39].

Definition 25. *A grammar $G = (\Sigma, N, R, S)$, is in* Chomsky Normal Form *iff every rule in R has the form either $A \to t$, or $A \to B \ominus C$, or $A \to B \oplus C$, where $A, B, C \in N$, and $t \in \Sigma$.*

We know from [39] that for every CFKG G, if $L(G)$ does not contain the empty picture, there exists a CFKG G' in Chomsky Normal Form, such that $L(G) = L(G')$. Also, the classical algorithm to translate a string grammar into Chomsky Normal Form can be easily adapted to CFKGs.

Example 4. The following Chomsky Normal Form grammar G defines the set of pictures such that each column is a palindrome:
$S \to V \oplus S \mid A_1 \ominus A_2 \mid B_1 \ominus B_2 \mid a \mid b$;
$V \to A_1 \ominus A_2 \mid B_1 \ominus B_2 \mid a \mid b$;
$A_2 \to V \ominus A_1 \mid a$;
$B_2 \to V \ominus B_1 \mid b$;
$A_1 \to a$;
$B_1 \to b$.

Proposition 14. *([14])* $\mathcal{L}(\mathrm{CFKG}) \subset \mathcal{L}(\mathrm{PG})$.

Namely, rules $A \to B \oplus C$ of a CF Kolam grammar G in CNF are equivalent to RTG rules:

$$A \to \left[\!\!\left[\begin{array}{cccccc} \# & \# & \# & \# & \# & \# \\ \# & B & B & C & C & \# \\ \# & B & B & C & C & \# \\ \# & \# & \# & \# & \# & \# \end{array} \right]\!\!\right]$$

and similarly an equivalent form can be stated for rules $A \to B \ominus C$. This is compatible with the constraint of Průša grammars given in Remark 1 and so for each CF Kolam grammar there exists an equivalent Průša's grammar. The inclusion is proper because the language of Example 1 cannot be generated by a CF Kolam grammar.

The time complexity of picture recognition problem for CF Kolam grammars in CNF has been recently proved [19] to be $O(m^2n^2(m+n))$. The significant difference with the time complexity of parsing for RTG grammars depends on the fact that in the right part of a rule of a CF Kolam grammars in CNF there are at most two distinct nonterminals. So, checking if a rule is applicable has complexity which is linear with respect to the picture width or height.

5.6 Context-free Matrix grammars

The early model of CF Matrix grammars [51] is a very limited kind of CF Kolam grammars. The following definition is taken and adapted from [48].

Definition 26. *Let $M = (G, G')$ where $G = (N, T, P, S)$ is a string grammar, where N is the set of nonterminals, P is a set of productions, S is the starting symbol, $T = \{A_1, A_2, \cdots, A_k\}$, $G' = \{G_1, G_2, \cdots, G_k\}$ where each A_i is the starting symbol of string grammar G_i. The grammars in G' are defined over an alphabet Σ, which is the alphabet of M. A grammar M is said to be a* CF Matrix Grammar *(*CFMG*) iff G and all G_i are CF grammars.*

Let $p \in \Sigma^{+,+}$, $p = c_1 \oplus c_2 \oplus \cdots \oplus c_n$. $p \in L(M)$ iff there exists a string $A_{x_1}A_{x_n} \cdots A_{x_n} \in L(G)$ such that every column c_j, seen as a string, is in $L(G_{x_j}), 1 \leq j \leq n$. The string $A_{x_1}A_{x_n} \cdots A_{x_n}$ is said to be an intermediate *string deriving p.*

If G and G_i for all $i,\ 1 \leq i \leq k$ are regular grammars then M is called a 2D right linear grammar.

Informally, the grammar G is used to generate an horizontal string of starting symbols for the "vertical grammars" $G_j, 1 \leq j \leq k$. Then, the vertical grammars are used to generate the columns of the picture. If every column has the same height, then the generated picture is defined, and is in $L(M)$.

It is trivial to show that the class of CFMG languages is a proper subset of CF Kolam languages. Intuitively, it is possible to consider the string sub-grammars G, and G_j, of a CF Matrix grammar M, all in Chomsky Normal Form. This means that we can define an equivalent M' CF Kolam grammar, in which rules corresponding to those of G use only the $\oplus$ operator, while rules corresponding to those of G_j use only the $\ominus$ operator.

Also, it is easy to adapt classical string parsing methods to Matrix grammars, see e.g. [48].

It is also well known that the family of languages generated by 2D right linear grammars is strictly included in the family of languages recognized by deterministic 4-way finite automata.

5.7 Grid grammars

Grid grammars are an interesting formalism defined by Drewes [22, 23]. Grid grammars are based on an extension of quadtrees [28], in which the number of "quadrants" is not limited to four, but can be k^2, with $k \geq 2$ (thus forming a square "grid").

Following the tradition of quadtrees, and differently from the other formalisms presented here, grid grammars generate pictures which are seen as set of points on the "unit square" delimited by the points (0,0), (0,1), (1,0), (1,1) of the Cartesian plane.

To compare such model, in which a picture is in the unit square and mono-chromatic (i.e. black and white), with the ones presented in this work, we introduce a different but basically compatible formalization, in which the generated pictures are square arrays of symbols, and the terminal alphabet is not limited to black and white. Our approach ([44]) is similar to the one used for Kolam grammars.

Definition 27. *A* sentential form *over an alphabet V is either a symbol $a \in V$, or $[t_{1,1}, \dots, t_{1,k}, \dots, t_{k,1}, \dots, t_{k,k}]$, with $k \geq 2$, and every $t_{i,j}$ being a sentential form. $\mathcal{SF}(V)$ denotes the set of all sentential forms over V.*

A sentential form ϕ defines a set of pictures $(\!|\phi|\!)$:

- *$(\!|a|\!)$, with $a \in V$, represents the set $\{a\}^{(n,n)}, n \geq 1$ of all a-homogeneous square pictures;*
- *$(\!|[t_{1,1}, \dots, t_{1,k}, \dots, t_{k,1}, \dots, t_{k,k}]|\!)$, represents the set of all square grid pictures where every $(\!|t_{i,j}|\!)$ has the same size $n \times n$, for $n \geq 1$, and $(\!|t_{1,1}|\!)$ is at the bottom-left corner, ..., $(\!|t_{1,k}|\!)$ is at the bottom right corner, ..., and $(\!|t_{k,k}|\!)$ is at the top right corner.*

For example, consider the sentential form $\phi = [[a, b, [a, b, b, a], c], a, B, [b, a, a, b]]$, the smallest picture in $(\!|\phi|\!)$ is

$$\begin{array}{cccccccc} B & B & B & B & a & a & b & b \\ B & B & B & B & a & a & b & b \\ B & B & B & B & b & b & a & a \\ B & B & B & B & b & b & a & a \\ b & a & c & c & a & a & a & a \\ a & b & c & c & a & a & a & a \\ a & a & b & b & a & a & a & a \\ a & a & b & b & a & a & a & a \end{array}$$

Definition 28. *A* Grid grammar (GG) *is a tuple $G = (\Sigma, N, R, S)$, where Σ is the finite set of* terminal *symbols, disjoint from the set N of* nonterminal *symbols; $S \in N$ is the* starting *symbol; and $R \subseteq N \times \mathcal{SF}(N \cup \Sigma)$ is the set of* rules*. A rule $(A, \phi) \in R$ will be written as $A \to \phi$.*

For a grammar G, we define the *derivation* relation $\Rightarrow_G$ on the sentential forms $\mathcal{SF}(N \cup \Sigma)$ by $\psi_1 \Rightarrow_G \psi_2$ iff there is some rule $A \to \phi$, such that ψ_2 results from ψ_1 by replacing an occurrence of A by ϕ.

From the derived sentential form, one then obtains the denoted picture. The picture language generated by G is the set

$$L(G) = \{\text{the smallest picture in } (\!|\psi|\!) \mid \psi \in \mathcal{SF}(\Sigma), S \overset{*}{\Rightarrow}_G \psi\}.$$

With a slight abuse of notation, we will often write $A \overset{*}{\Rightarrow}_G p$, with $A \in N, p \in \Sigma^{*,*}$, instead of $\exists \phi : A \overset{*}{\Rightarrow}_G \phi, (\!|\phi|\!) = p$.

In literature, parameter k is fixed for a Grid grammar G, i.e. all the right parts of rules are either terminal or $k \times k$ grids. This constraint could be relaxed, by allowing different k for different rules: the results that are shown next still hold for this generalization.

It is trivial to see that grid grammars admit a Nonterminal Normal Form:

Definition 29. *A grid grammar $G = (\Sigma, N, R, S)$, is in* Nonterminal Normal Form *(NNF) iff every rule in R has the form either $A \to t$, or $A \to [B_{1,1}, \ldots, B_{1,k}, \ldots, B_{k,1}, \ldots, B_{k,k}]$, where $A, B_{i,j} \in N$, and $t \in \Sigma$.*

Example 5. Here is a simple example of a grid grammar in NNF.

$$S \to [S, B, S, B, B, B, S, B, S], \quad S \to a, \quad B \to b.$$

The generated language is that of "recursive" crosses of b's in a field of a's.
An example picture:

$$\begin{array}{ccccccccc}
a & b & a & b & b & b & a & a & a \\
b & b & b & b & b & b & a & a & a \\
a & b & a & b & b & b & a & a & a \\
b & b & b & b & b & b & b & b & b \\
b & b & b & b & b & b & b & b & b \\
b & b & b & b & b & b & b & b & b \\
a & b & a & b & b & b & a & a & a \\
b & b & b & b & b & b & a & a & a \\
a & b & a & b & b & b & a & a & a
\end{array}$$

First, we note that this is the only 2D grammatical model presented in this paper which cannot generate string languages, since all the generated pictures, if any, have the same number of rows and columns by definition.

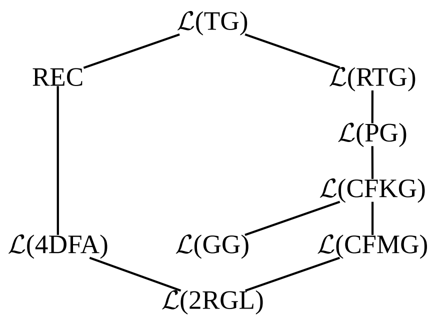
It is easy to see that the class of languages generated by grid grammars are a proper subset of the one of CF Kolam grammars.

Proposition 15. *([44]) $\mathcal{L}(\mathrm{GG}) \subset \mathcal{L}(\mathrm{CFKG})$. $\mathcal{L}(\mathrm{CFMG})$ and $\mathcal{L}(\mathrm{GG})$ are incomparable.*

By definition, grid grammars can generate only square pictures and on the other hand, it is impossible to define CF Matrix grammars generating only square pictures.

5.8 Summary

We finish with a synopsis of the previous language family inclusions.



6 Conclusion

First of all we want to remark that there are several different ways to generate or recognize picture languages that are not considered in this survey, e.g. [16], [43].

Since REC is a robust notion, we believe that it is a necessary starting point for a tutorial on picture languages. If one assumes that REC is the right answer to the quest of a analog for regular string languages then, to maintain hierarchy, TG grammars is the notion corresponding to context-free grammars. This is why we choose to describe this model among the others.

RTG preserves some nice properties of context free string languages and includes several well known models usually introduced as a generalization of context free grammars. So, a question naturally arises: if RTG is the right model for generating context free picture languages, what about the right model for regular string languages? Some criticisms on the fact that REC recognizes a too wide class to be considered the right model in spite of its robustness was posed for instance in [42]. It could be interesting to consider which languages are defined by non recursive RTG grammars in order to verify whether that family can also be proposed as the analog of regular string languages.

Moreover, few attention was paid to study the generalization to two dimensions of push-down automata. For instance how can be defined automata recognizing all the families of languages generated by grammars described in this survey? And finally are there more promising grammatical approaches to "context-free" picture languages?

In conclusion, in our opinion the very idea of defining a Chomsky's hierarchy analogous, moving from one to two dimensions, is probably doomed to partial unsuccess. 2D structures and formalisms, albeit maintaining some similarities with their 1D counterparts, often exhibit very different formal properties and issues which are not present or trivial in string languages.

Acknowledgments. We thank Achille Frigeri for his valuable comments.

References

1. A. Alstes. Wang tiles for image and texture generation. Technical Report 49432R, Helsinki University of Technology, 2004.
2. M. Anselmo, D. Giammarresi, and M. Madonia. New operations and regular expressions for two-dimensional languages over one-letter alphabet. *Theor. Comput. Sci.*, 304:408–431, 2005.
3. M. Anselmo, D. Giammarresi, and M. Madonia. From determinism to non-determinism in recognizable two-dimensional languages. In T. Harju, J. Karhumäki, and A. Lepistö, editors, *DLT 2007*, volume 4588 of *Lecture Notes in Computer Science*, pages 36–47. Springer, 2007.
4. M. Anselmo, D. Giammarresi, and M. Madonia. Tiling automaton: A computational model for recognizable two-dimensional languages. In J. Holub and J. Zdárek, editors, *CIAA 2007*, volume 4783 of *Lecture Notes in Computer Science*, pages 290–302. Springer, 2007.
5. M. Anselmo, D. Giammarresi, M. Madonia, and A. Restivo. Unambiguous recognizable two-dimensional languages. *Theoretical Informatics and Applications*, 40(2):277–293, 2006.
6. M. Anselmo, N. Jonoska, and M. Madonia. Framed versus unframed two-dimensional languages. In M. Nielsen, A. Kucera, P. B. Miltersen, C. Palamidessi, P. Tuma, and F. D. Valencia, editors, *SOFSEM 2009*, volume 5404 of *Lecture Notes in Computer Science*, pages 79–92, 2009.
7. M. Anselmo and M. Madonia. Simulating two-dimensional recognizability by pushdown and queue automata. In J. Farré, I. Litovsky, and S. Schmitz, editors, *CIAA 2005*, volume 3845 of *Lecture Notes in Computer Science*, pages 43–53. Springer, 2006.

8. R. Berger. *The Undecidability of the Domino Problem*. Number 66 in Memoirs of the AMS. AMS, Providence, 1966.
9. A. Bertoni, M. Goldwurm, and V. Lonati. On the complexity of unary tiling-recognizable picture languages. *Fundamenta Informaticae*, 91(2):231–249, 2009.
10. M. Blum and C. Hewitt. Automata on a 2-dimensional tape. In *FOCS 1967*, pages 155–160. IEEE, 1967.
11. S. Bozapalidis. Picture deformation. *Acta Inf.*, 45(1):1–31, 2008.
12. S. Bozapalidis and A. Grammatikopoulou. Recognizable picture series. *Journal of Automata, Languages and Combinatorics*, 10(2-3):159–183, 2005.
13. Y. Brun. Solving NP-complete problems in the tile assembly model. *Theor. Comput. Sci.*, 395(1):31–46, 2008.
14. A. Cherubini, S. Crespi Reghizzi, and M. Pradella. Regional languages and tiling: A unifying approach to picture grammars. In E. Ochmanski and J. Tyszkiewicz, editors, *MFCS 2008*, volume 5162 of *Lecture Notes in Computer Science*, pages 253–264. Springer, 2008.
15. A. Cherubini, S. Crespi Reghizzi, M. Pradella, and P. San Pietro. Picture languages: Tiling systems versus tile rewriting grammars. *Theor. Comput. Sci.*, 356(1-2):90–103, 2006.
16. C. Choffrut and B. Durak. Collage of two-dimensional words. *Theor. Comput. Sci.*, 340(1):364–380, 2005.
17. M. F. Cohen, J. Shade, S. Hiller, and O. Deussen. Wang tiles for image and texture generation. *ACM Trans. Graph.*, 22(3):287–294, 2003.
18. S. Crespi Reghizzi and M. Pradella. Tile rewriting grammars and picture languages. *Theor. Comput. Sci.*, 340(2):257–272, 2005.
19. S. Crespi Reghizzi and M. Pradella. A CKY parser for picture grammars. *Information Processing Letters*, 105(6):213–217, 2008.
20. K. Culik II. An aperiodic set of 13 Wang tiles. *Discrete Mathematics*, 160:245–251, 1996.
21. L. de Prophetis and S. Varricchio. Recognizability of rectangular pictures by Wang systems. *Journal of Automata, Languages and Combinatorics*, 2(4):269–288, 1997.
22. F. Drewes. Language theoretic and algorithmic properties of d-dimensional collages and patterns in a grid. *J. Comput. Syst. Sci.*, 53(1):33–66, 1996.
23. F. Drewes, S. Ewert, R. Klempien-Hinrichs, and H.-J. Kreowski. Computing raster images from grid picture grammars. *Journal of Automata, Languages and Combinatorics*, 8(3):499–519, 2003.
24. B. Durand. Tilings and quasiperiodicity. *Theor. Comput. Sci.*, 221:61–75, 1999.
25. B. Durand. De la logique aux pavages. *Theor. Comput. Sci.*, 281:311–324, 2002.
26. B. Durand, L. A. Levin, and A. Shen. Local rules and global order, or aperiodic tilings. *Math. Intelligencer*, 27(1):64–68, 2005.
27. S. Eigen, J. Navarro, and V. S. Prasad. An aperiodic tiling using a dynamical system and Beatty sequences. In B. Hasselblatt, editor, *Dynamics, Ergodic Theory and Geometry*, number 54 in Mathematical Sciences Research Institute Publications, pages 223–241. Cambridge Univ. Press, 2007.
28. R. A. Finkel and J. L. Bentley. Quad trees: A data structure for retrieval on composite keys. *Acta Inf.*, 4:1–9, 1974.
29. D. Giammarresi. Tiling recognizable two-dimensional languages. In S. Bozapalidis and G. Rahonis, editors, *CAI 2007*, volume 4728 of *Lecture Notes in Computer Science*, pages 75–86. Springer, 2007.
30. D. Giammarresi and A. Restivo. Recognizable picture languages. *International Journal of Pattern Recognition and Artificial Intelligence*, 6(2-3):241–256, 1992.
31. D. Giammarresi and A. Restivo. *Handbook of formal languages*, volume 3: beyond words, chapter Two-dimensional languages, pages 215–267. Springer, Berlin, 1997.
32. D. Giammarresi, A. Restivo, S. Seibert, and W. Thomas. Monadic second-order logic over rectangular pictures and recognizability by tiling systems. *Inf. Comput.*, 125(1):32–45, 1996.

33. B. Grünbaum and G. C. Shephard. *Tilings and Patterns*. W.H. Freeman and Company, New York, 1987.
34. Y. S. Gurevich and I. O. Koriakov. Remarks on Berger's paper on the domino problem. *Siberian Mathematical Journal*, 13(2):319–321, 1972.
35. K. Inoue and A. Nakamura. Some properties of two-dimensional on-line tessellation acceptors. *Information Sciences*, 13:95–121, 1977.
36. K. Inoue and I. Takanami. A survey of two-dimensional automata theory. *Information Sciences*, 55(1-3):99–121, 1991.
37. J. Kari. A small aperiodic set of Wang tiles. *Discrete Mathematics*, 160(1-3):259–264, 1996.
38. V. Lonati and M. Pradella. Snake-deterministic tiling systems. *MFCS 2009*, To appear.
39. O. Matz. Regular expressions and context-free grammars for picture languages. In R. Reischuk and M. Morvan, editors, *STACS 1997*, volume 1200 of *Lecture Notes in Computer Science*, pages 283–294. Springer, 1997.
40. O. Matz. On piecewise testable, starfree, and recognizable picture languages. In M. Nivat, editor, *FoSSaCS 1998*, volume 1378 of *Lecture Notes in Computer Science*, pages 203–210. Springer, 1998.
41. O. Matz. One quantifier will do in existential monadic second-order logic over pictures. In L. Brim, J. Gruska, and J. Zlatuska, editors, *MFCS 1998*, volume 1450 of *Lecture Notes in Computer Science*, pages 751–759. Springer, 1998.
42. O. Matz. Recognizable vs. regular picture languages. In S. Bozapalidis and G. Rahonis, editors, *CAI 2007*, volume 4728 of *Lecture Notes in Computer Science*, pages 112–121. Springer, 2007.
43. M. Nivat, A. Saoudi, K. G. Subramanian, R. Siromoney, and V. Rajkumar Dare. Puzzle grammars and context-free array grammars. *International Journal of Pattern Recognition and Artificial Intelligence*, 5:663–676, 1991.
44. M. Pradella, A. Cherubini, and S. Crespi Reghizzi. A unifying approach to context-free picture languages. Submitted, 2009.
45. M. Pradella and S. Crespi Reghizzi. A SAT-based parser and completer for pictures specified by tiling. *Pattern Recognition*, 41(2):555–566, 2008.
46. D. Průša. Two-dimensional context-free grammars. In G. Andrejková and S. Krajči, editors, *ITAT 2001*, pages 27–40, 2001.
47. D. Průša. *Two-dimensional Languages*. Phd thesis, Charles University, Faculty of Mathematics and Physics, Prague, 2004.
48. V. Radhakrishnan, V. T. Chakaravarthy, and K. Krithivasan. Pattern matching in matrix grammars. *Journal of Automata, Languages and Combinatorics*, 3(1):59–76, 1998.
49. R. M. Robinson. Undecidability and nonperiodicity for tilings of the plane. *Inventiones Mathematicae*, 12:177–209, 1971.
50. D. Simplot. A characterization of recognizable picture languages by tilings by finite sets. *Theor. Comput. Sci.*, 218(2):297–323, 1999.
51. G. Siromoney, R. Siromoney, and K. Krithivasan. Abstract families of matrices and picture languages. *Computer Graphics and Image Processing*, 1(3):284–307, 1972.
52. G. Siromoney, R. Siromoney, and K. Krithivasan. Picture languages with array rewriting rules. *Information and Control*, 23(5):447–470, 1973.
53. H. Wang. Proving theorems by pattern recognition II. *Bell Systems Technical Journal*, 40:1–42, 1961.
54. H. Wang. Dominoes and the AEA case of the decision problems. In *Mathematical theory of automata: Symposium on Mathematical Theory of Automata 1963*, volume 12 of *Brooklyn Polytechnic Institute. Microwave Research Institute. Symposia series*, pages 23–55. Polythechnic Press, 1963.

55. T. Wilke. Star-free picture expressions are strictly weaker than first-order logic. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *ICALP 1997*, volume 1256 of *Lecture Notes in Computer Science*, pages 347–357. Springer, 1997.
56. E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:439–544, 1998.