Picture recognizability with automata based on Wang tiles*

Violetta Lonati[1] and Matteo Pradella[2]

[1] Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano
Via Comelico 39/41, 20135 Milano, Italy – lonati@dsi.unimi.it
[2] IEIIT, Consiglio Nazionale delle Ricerche
Via Golgi 40, 20133 Milano, Italy – matteo.pradella@polimi.it

Abstract. We introduce a model of automaton for picture language recognition which is based on tiles and is called Wang automaton, since its description relies on the notation of Wang systems. Wang automata combine features of both online tessellation acceptors and 4-ways automata: as in online tessellation acceptors, computation assigns states to each picture position; as in 4-way automata, the input head visits the picture moving from one pixel to an adjacent one, according to some scanning strategy. We prove that Wang automata recognize the class REC, i.e. they are equivalent to tiling systems or online tessellation acceptors, and hence strictly more powerful than 4-way automata. We also consider a very natural notion of determinism for Wang automata, and study the resulting class, comparing it with other deterministic classes considered in the literature, like DREC and Snake-DREC.

Keywords: picture languages, 2D languages, tiling systems, 4-way automata, online tessellation acceptors, Wang systems, determinism.

1 Introduction

Picture languages are a generalization of string languages to two dimensions: a picture is a two-dimensional array of elements from a finite alphabet. The literature on picture languages is quite rich of models, see e.g. [14,12,17,9,5,7,3,8]. Here we regard class REC, introduced in [12] with the aim to generalize to 2D the class of regular string languages. REC is a robust class that has various characterizations: for instance it is the class of picture languages that can be generated by *online tessellation automata* [13], *tiling systems* [11], or *Wang systems* [10].

In this paper we characterize REC by introducing a new model of 2D automata based on tiles. We call such model *Wang automaton*, since its description is based on the notation of Wang systems. Wang automata combine features of both online tessellation acceptors [13] and 4-ways automata [14]: as in online tessellation acceptors,

* This work has been supported by the MIUR PRIN projects "Mathematical aspects and emerging applications of automata and formal languages", and CNR RSTL 760 "2D grammars for defining pictures".

computation assigns states to each input picture position; as in 4-way automata, the input head visits the input picture following a given *scanning strategy*, that is a method to visit its positions.

The choice of a suitable scanning strategy is a central issue in this context. In particular it has been considered recently in [2,6]. Here we introduce *polite* scanning strategies, that sort all positions in a picture, and visit each of them exactly once, in such a way that the next position to visit is always adjacent to the previous one, and depends only on this information: which neighboring positions have already been visited, and which direction we are moving from. Examples of such scanning strategies are those following the boustrophedonic order, spirals, and many others.

Differently from 4-way automata, Wang automata directed by polite scanning strategies visit each position exactly once; moreover, one can consider various polite scanning strategies, but next position cannot depend on the input symbol (in a sense, like traditional finite state automata on strings). However, we prove that this kind of automata are equivalent to tiling systems, thus they are strictly more powerful than 4-way automata [12].

An interesting aspect of this new model is the possibility to introduce quite naturally the notion of determinism, yielding class Scan-DREC, which is closed under complement and rotation. Determinism is a crucial concept in language theory, whereas in two dimensions it is far from being well understood. Tiling systems are implicitly nondeterministic: REC is not closed under complement, and the membership problem is NP-complete [15].

In the literature several notions of determinism for recognizable languages and automata have been proposed. For 4-way automata the definition of determinism is straightforward [14]. Online tessellation acceptors have a diagonal-based kind of determinism [13] and this notion is extended in [1], with the definition of a deterministic class we denote by Diag-DREC (the original name was DREC). In [16] we introduced the class Snake-DREC which is based on a boustrophedonic scanning strategy, and proved that Snake-DREC properly extends Diag-DREC.

Here we prove that Scan-DREC properly extends Snake-DREC (and hence class Diag-DREC) and is closed under complement and rotation. Several questions concerning the relationship among these classes remain open and are proposed in the conclusions.

We cite also an interesting and radically different notion of determinism, proposed in [4], which is not based on prefixed scanning strategies. Such notion is built upon a definition of language recognized by a tiling system which is different from the usual one, e.g. of [12], so it is hard to compare it with other approaches. For instance, while it is decidable to check if a tiling system is deterministic in one of the senses presented before, at present we do not know anything about decidability for [4].

The paper is organized as follows. In Section 2 we recall some basic definitions and properties on two-dimensional languages, tiling systems, Wang systems, and determinism. In Section 3 we introduce polite scanning strategies and compare them with scanning strategies already studied in the literature. In Section 4 we present our model of Wang automaton and prove the main theorem characterizing REC as the class of picture languages recognized by Wang automata directed by polite scanning strategies. In

Section 5 we introduce the concept of determinism natural in this framework, and compare the corresponding class with Diag-DREC and Snake-DREC. In the last section we propose some open questions concerning determinism in 2D.

2 Preliminaries

The following definitions are taken and adapted from [12].

Let Σ be a finite alphabet. A two-dimensional array of elements of Σ is a *picture* over Σ. The set of all pictures over Σ is Σ^{++}. A picture language is a subset of Σ^{++}. If C denotes some kind of picture-accepting device, then $\mathcal{L}(\mathrm{C})$ denotes the class of picture languages recognized by such devices.

For $n, m \geq 1$, $\Sigma^{n,m}$ denotes the set of pictures of size (n, m); $\# \notin \Sigma$ is used when needed as a *boundary symbol*; $\hat{p}$ refers to the bordered version of picture p. That is, for $p \in \Sigma^{n,m}$, it is

$$p = \begin{array}{|c|c|c|}\hline p(1,1) & \dots & p(1,m) \\ \hline \vdots & \ddots & \vdots \\ \hline p(n,1) & \dots & p(n,m) \\ \hline \end{array} \qquad \hat{p} = \begin{array}{|c|c|c|c|c|}\hline \# & \# & \dots & \# & \# \\ \hline \# & p(1,1) & \dots & p(1,m) & \# \\ \hline \vdots & \vdots & \ddots & \vdots & \vdots \\ \hline \# & p(n,1) & \dots & p(n,m) & \# \\ \hline \# & \# & \dots & \# & \# \\ \hline \end{array}.$$

A *pixel* is an element $p(i, j)$ of p. We call (i, j) the *position* in p of the pixel. We will sometimes use position (i, j) with i or j equal to 0, or $n + 1$, or $m + 1$ for referring to borders. We use the term *picture domain* (or domain for short) to refer to the set of possible positions in a generic picture of size (n, m), not considering borders, i.e. the set $n \times m = \{1, 2, \dots, n\} \times \{1, 2, \dots, m\}$. Each position has four *edges*, and an edge is identified by a pair of (vertically or horizontally) *adjacent* positions.

We will sometimes consider the 90º clockwise *rotation* of a picture p. E.g. if $p = \begin{array}{|c|c|}\hline a & b \\ \hline c & d \\ \hline \end{array}$, then $\begin{array}{|c|c|}\hline c & a \\ \hline d & b \\ \hline \end{array}$ is its rotation. Naturally, the same operation can be applied to languages, and classes of languages, too.

2.1 Tiling systems

We call *tile* a square picture of size (2,2). We denote by $T(p)$ the set of all tiles contained in a picture p.

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

Let $\pi : \Gamma \to \Sigma$ be a mapping between two alphabets. Given a picture $p \in \Gamma^{++}$, the *projection* of p by π is the picture $\pi(p) \in \Sigma^{++}$ such that $\pi(p)\ (i, j) = \pi(p(i, j))$ for every position (i, j). Analogously, the projection of a language $L \subseteq \Gamma^{++}$ by π is the set $\pi(L) = \{\pi(p) \mid p \in \Gamma^{++}\} \subseteq \Sigma^{++}$.

A *tiling system* (TS) is a 4-tuple $\tau = \langle \Sigma, \Gamma, \Theta, \pi \rangle$ where Σ and Γ are two finite alphabets, Θ is a finite set of tiles over the alphabet $\Gamma \cup \{\#\}$ and $\pi : \Gamma \to \Sigma$ is a projection. A picture language $L \subseteq \Sigma^{++}$ is *tiling recognizable* if there exists a tiling system $\langle \Sigma, \Gamma, \Theta, \pi \rangle$ such that $L = \pi(L(\Theta))$. We say that τ generates L and denote by REC the class of picture languages that are tiling recognizable, i.e, REC $= \mathcal{L}(\mathrm{TS})$. Notice in particular that any local language is tiling recognizable.

Example 1. Consider the language L_{half} of pictures of size $(n, 2n)$ with the first row like $x \cdot \bar{x}$, where $\bar{x}$ is the reverse of x. We show that L_{half} is recognized by a tiling system. Let Γ be the set of letters of the form σ_ρ, with $\sigma, \rho \in \Sigma$. For each picture p in L_{half}, consider the picture $p' \in \Gamma^{++}$ where subscripts are used to connect each letter in x to the corresponding letter in $\bar{x}$, along nested paths following a $\sqcup$-like form. Below there is an example of such pair of pictures p and p' (the colors in p' are used in the figure only to emphasize the $\sqcup$-like form of the resulting paths):

$$p = \begin{array}{|c|c|c|c|c|c|} \hline a & b & c & c & b & a \\ \hline b & b & a & c & b & a \\ \hline c & a & a & b & a & a \\ \hline \end{array}\,, \qquad p' = \begin{array}{|c|c|c|c|c|c|} \hline a_a & b_b & c_c & c_{c'} & b_b & a_a \\ \hline b_a & b_b & a_b & c_b & b_b & a_a \\ \hline c_a & a_a & a_a & b_a & a_a & a_a \\ \hline \end{array}\,.$$

One can show that the language of pictures p' is local, and hence L_{half} is in REC.

2.2 Wang systems

In [10] a model equivalent to tiling systems but based on a variant of Wang tiles was introduced. A Wang tile is a unitary square with colored edges. Color represents *compatibility*: two tiles may be adjacent only if the color of the touching edges is the same. A *labeled Wang tile* is a Wang tile bearing also a *label*; a set of such tiles is called *Wang system*.

More formally, given a finite alphabet $Colrs$ of colors, and a finite alphabet Σ of labels, a labeled Wang tile is a quintuple (n, s, e, w, x), with $n, s, e, w \in Colrs \cup \{\#\}$ (where, as usual, $\#$ is a color representing borders), and $x \in \Sigma$. Intuitively, n, s, e, w represent the colors respectively at the top, bottom, right, and left of the tile. For better readability, we represent the labeled Wang tile (n, s, e, w, x) as $\begin{array}{c} n \\ w\boxed{x}e \\ s \end{array}$.

Given $\Phi \subseteq Colrs^4 \times \Sigma$, a *Wang-tiled picture over Φ* is any picture in Φ^{++} such that adjacent pixels are compatible, also considering borders, as in the following example:

$$\begin{array}{|c|c|} \hline \begin{array}{c} \# \\ \#\boxed{a}4 \\ 1 \end{array} & \begin{array}{c} \# \\ 4\boxed{b}\# \\ 3 \end{array} \\ \hline \begin{array}{c} 1 \\ \#\boxed{b}2 \\ \# \end{array} & \begin{array}{c} 3 \\ 2\boxed{a}\# \\ \# \end{array} \\ \hline \end{array} \in \Phi^{2,2}.$$

The *label* of a Wang-tiled picture P over Φ is the picture over Σ having for pixels the labels of pixels of P. For instance, the label of the example above is $\begin{array}{|c|c|} \hline a & b \\ \hline b & a \\ \hline \end{array}$.

A Wang system ω is a triple $(Colrs, \Sigma, \Phi)$. The language generated by ω is the language over Σ of all labels of Wang-tiled pictures over Φ.

Example 2. A Wang system recognizing L_{half} can be defined using the same idea presented in Example 1. The resulting Wang-tiled pictures have the form

$$P = \begin{array}{|c|c|c|c|c|c|}
\hline
\begin{matrix}\#\\ \#\boxed{a}\cdot\\ a\end{matrix} & \begin{matrix}\#\\ \cdot\boxed{b}\cdot\\ b\end{matrix} & \begin{matrix}\#\\ \cdot\boxed{c}c\\ \cdot\end{matrix} & \begin{matrix}\#\\ c\boxed{c}\cdot\\ \cdot\end{matrix} & \begin{matrix}\#\\ \cdot\boxed{b}\cdot\\ b\end{matrix} & \begin{matrix}\#\\ \cdot\boxed{a}\#\\ a\end{matrix} \\
\hline
\begin{matrix}a\\ \#\boxed{b}\cdot\\ a\end{matrix} & \begin{matrix}b\\ \cdot\boxed{b}b\\ \cdot\end{matrix} & \begin{matrix}\cdot\\ b\boxed{a}b\\ \cdot\end{matrix} & \begin{matrix}\cdot\\ b\boxed{c}b\\ \cdot\end{matrix} & \begin{matrix}b\\ b\boxed{b}\cdot\\ \cdot\end{matrix} & \begin{matrix}a\\ \cdot\boxed{a}\#\\ a\end{matrix} \\
\hline
\begin{matrix}a\\ \#\boxed{c}a\\ \#\end{matrix} & \begin{matrix}\cdot\\ a\boxed{a}a\\ \#\end{matrix} & \begin{matrix}\cdot\\ a\boxed{a}a\\ \#\end{matrix} & \begin{matrix}\cdot\\ a\boxed{b}a\\ \#\end{matrix} & \begin{matrix}\cdot\\ a\boxed{a}a\\ \#\end{matrix} & \begin{matrix}a\\ a\boxed{a}\#\\ \#\end{matrix} \\
\hline
\end{array} . \qquad (1)$$

As before, colors are used only to emphasize the ⊔-form of the paths.

2.3 Diagonal- and snake-deterministic tiling systems

Tiling systems are implicitly nondeterministic: REC is not closed under complement, and the membership problem is NP-complete [15]. Moreover, any notion of deterministic tiling systems seems to require some pre-established "scanning strategy" to read the picture, an important issue we deal with in the following section. Here we recall two notions of determinism recently introduced in the literature. They are both defined using the notation of tiling systems, but it is quite natural translate them from tiling systems to Wang systems.

Diagonal determinism [1] is inspired by the deterministic version of online tessellation acceptors [13], which are directed according to a corner-to-corner direction (namely, from top-left to bottom-right, or *tl2br*). Consider a scanning strategy that follows the tl2br direction: any position (x, y) is read only if all the positions that are above and to the left of (x, y) have already been read. An example of such scanning strategy is depicted in Figure 1(a). Roughly speaking, tl2br determinism means that, given a picture $p \in \Sigma^{++}$, its preimage $p' \in L(\Theta) \subseteq \Gamma^{++}$ can be build deterministically when scanning p with any such strategy: *tl2br-deterministic* tiling systems guarantee this condition (the formal definition can be found in [1]). They are proved to be equivalent to deterministic online tessellation acceptors.

Snake-determinism [16] is based on boustrophedonic scanning strategies. Given a tiling system $\tau = \langle \Sigma, \Gamma, \Theta, \pi \rangle$ and a picture $p \in \Sigma^{++}$, imagine to build one preimage $p' \in L(\Theta)$, $\pi(p') = p$, by scanning p as follows: start from the top-left corner, scan the first row of p rightwards, then scan the second row leftwards, and so on, as in Figure 1(b). This means that we scan odd rows rightwards and even row leftwards, assigning a symbol in Γ to each position. A tiling system is *snake-deterministic* if this choice is guaranteed unique (the formal definition can be found in [16]).

Diag-DREC (resp. Snake-DREC) is the family of languages such that one of their rotations is recognized by a *tl2br*-deterministic (resp. snake-deterministic) tiling system. Diag-DREC ⊂ Snake-DREC ⊂ REC with all proper inclusions.

3 Two-dimensional scanning strategies

The notions of determinism in [1,16] are all based on some fixed and pre-established kinds of scanning strategy. This approach can be limiting, so we plan to define and consider here a wider range of possible strategies. We will start by introducing the central concept of *scanning strategy*, and then discussing the two related approaches of [2] and [6].

Definition 1. *A* scanning strategy *is a family*

$$\mu = \{\mu_{n\times m} : \{1, 2, \ldots\} \to n \times m\}_{n,m}$$

and $\mu_{m\times n}$ *is called the* scanning function over domain $n \times m$. *A scanning strategy is said to be* continuous *if* $\mu_{n\times m}(i + 1)$ *is adjacent to* $\mu_{n\times m}(i)$ *for every* n, m, i*; it is said to be* one-pass *if each scanning function* $\mu_{n\times m}$ *restricted to* $\{1, 2, \ldots, nm\}$ *is a bijection.*

Intuitively, a scanning strategy provides a method to visit positions in any picture domain: $\mu_{n\times m}(i)$ is the position visited in domain $n \times m$ at time i. One-pass strategies are those that visit each position in each domain exactly once.

Some one-pass scanning strategies are illustrated in Figure 1. Actually they are not fully defined: only the function $\mu_{3\times4}$ is depicted whereas the other functions should be defined analogously; each position c of domain 3×4 contains the number i such that $c = \mu_{3\times4}(i)$. The strategy (a) is not continuous and visits one row after the other, from left to right and from top to bottom; the other strategies are all continuous.

1	2	3	4
5	6	7	8
9	10	11	12

(a)

1	2	3	4
8	7	6	5
9	10	11	12

(b)

1	2	3	4
10	11	12	5
9	8	7	6

(c)

1	12	9	8
2	11	10	7
3	4	5	6

(d)

1	2	11	10
4	3	12	9
5	6	7	8

(e)

Fig. 1. Some one-pass scanning strategies: the number in each pixel denotes its scanning order. (a) is not continuous (b) has a boustrophedonic behavior, (c) has a spiral behavior, (d) draws nested $\sqcup$-like paths, (e) combines the behavior of (b) in the first half of the picture and a counterclockwise variant of (c) in the second one.

In the literature on 2D languages, two recent works considered the problem of defining scanning strategies for pictures, namely [2] and [6].

In [2] an automata model called *tiling automaton* is introduced, with the aim to define a general computational model for recognizable languages. This approach is centered upon the concept of scanning strategy itself, which directly depends on the size of the picture to be scanned. This definition is very general, and may exploit the size of the picture to perform "jumps", thus allowing complex behaviors. This freedom, together with the potential knowledge of the picture size, may be exploited to exceed REC (in practice, scanning strategies presented and used in [2] are simpler, and do not exhibit this issue). Consider e.g. the 1D non-regular language $x \cdot \bar{x}$, with $x \in \{a, b\}^+$ and $\bar{x}$ equal to the reverse of x: if we are able to jump back and forth, starting from the first character, and then going to the last, then considering the second, the last-but-one and so on, we can easily define an accepting automaton.

The very recent work [6] is also based on the concept of scanning strategy. In it, the considered strategies are "continuous" (hence called "snakes", not to be confused with the homonym we used before), in the sense that the next considered position is adjacent to the current one. The actual definition of such strategies is not formally presented, as the authors preferred more qualitative considerations. This aspect could be source of some problems, since may admit different strategies depending on the picture size or shape (e.g. Peano-Hilbert curves are suitable only for square pictures). For example, if we consider unary languages, having different strategies which depends on the shape/size of the figure itself may be exploited to exceed REC also in this case.

In our opinion all these issues could be addressed by introducing a qualitative concept, that we will call of *blindness* of the strategy. We consider *blind* a strategy which proceeds locally, by scanning adjacent positions, and cannot "see" neither the picture content, nor its size: it can only "feel" a border and an already considered position, when it reaches it. Considering the strategies presented in Figure 1, (a) is not blind, since it uses the knowledge of picture's width, after reaching the end of a row, to "jump" back to the beginning of the next row. Analogously, (e) is not blind, since it exploits the knowledge of the width of picture, to change direction when reaching its half. We accept all the other presented strategies, as they only depend on local information: already considered positions, and borders.

In the following we try to capture this idea of blindness, by adding some constraints to the scanning strategies we consider. To this aim, we shall need some notations.

Given a position y, we use edges(y) to denote the set of 4 edges adjacent to y. $Dirs$ is the set of directions $\{\mathrm{r}, \mathrm{l}, \mathrm{t}, \mathrm{b}\}$. For every position y, and $d \in Dirs$, the edge of y in direction d is denoted by $y|d$, and the position adjacent to y in direction d is denoted by $y \boxplus d$.

A *next-position function* is a partial function $\eta : 2^{Dirs} \times Dirs \rightarrow Dirs$ such that $\eta(D, d) = \perp$ if $d \notin D$. Informally, the meaning of η is that, for a given position, we have a set of already considered edges, given by the set D of directions, and d, the "last-considered" one. η is used to choose where to go next, i.e. the direction towards the position to visit next.

Now fix any next-position function η, any starting corner $c_s \in Corners = \{\mathrm{tl}, \mathrm{tr}, \mathrm{br}, \mathrm{bl}\}$ and any starting direction $d_s \in Dirs$. Then, for every picture domain $n \times m$, consider the following scanning function $\mu_{n\times m}$ over $n \times m$.

- The starting position is

$$\mu_{n\times m}(1) = \begin{cases} (1,1) & \text{if } c_s = \mathrm{tl} \\ (1,m) & \text{if } c_s = \mathrm{tr} \\ (n,1) & \text{if } c_s = \mathrm{bl} \\ (n,m) & \text{if } c_s = \mathrm{br} \end{cases}$$

  moreover we define E_1 as the set of outer edges (i.e. those adjacent to borders) of the picture domain $n \times m$, and we set $d_1 = d_s$.

– The inductive definition[3] of $\mu_{n\times m}(i+1)$ for $i \geq 1$ is given by:

$$D_i = \{d \in \mathit{Dirs} : \mu_{n\times m}(i)|d \in E_i\} \qquad E_{i+1} = E_i \cup \mathrm{edges}(\mu_{n\times m}(i))$$

$$d_{i+1} = \eta(D_i, d_i) \qquad \mu_{n\times m}(i+1) = \mu_{n\times m}(i) \boxplus d_{i+1}$$

Notice that $\mu_{n\times m}(1)|d_1$ must be in E_1 for $\eta(D_1, d_1)$ to be defined.

We say that $\mu = \{\mu_{n\times m}\}_{n,m}$ is the scanning strategy *induced by* the triple $\langle \eta, c_s, d_s\rangle$.

Definition 2. *A scanning strategy is* blind *if it is induced by a triple $\langle \eta, c_s, d_s\rangle$, where η is a next-position function, c_s a starting corner, and d_s a starting direction.*

Notice that, in general, a blind scanning strategy is not one-pass. However, it is continuous and satisfies the other requirements we need. First, all scanning functions are defined by the same triple $\langle \eta, c_s, d_s\rangle$ for every picture domain; second, the next position to visit always depends only on this information: which neighboring positions have already been visited, and which direction we are moving from. This yields the following definition.

Definition 3. *A scanning strategy is called* polite *if it is blind and one-pass.*

4 Wang automata

We are now able to formally introduce Wang automata and to show that they are equivalent to tiling systems.

Let *Colrs* be a set of *colors*. If the edges adjacent to a position are (partially or fully) colored, a coloring will be used to summarize their colors. Formally, we call *coloring* any partial function $\gamma : \mathit{Dirs} \to \mathit{Colrs}$. The set of directions where γ is defined is denoted by Δ_γ. If $\Delta_\gamma = \mathit{Dirs}$, then γ is called a *full* coloring. Given $\gamma_1, \gamma_2 \in \mathit{Colrs}$, we say that γ_2 extends γ_1 if $\gamma_2(d) = \gamma_1(d)$ for every $d \in \Delta_{\gamma_1}$.

Definition 4. *A μ-*directed Wang automaton *(μ-WA) is a tuple $\langle \Sigma, \mathit{Colrs}, \delta, \mu, F\rangle$ where:*

- *Σ is a finite input alphabet,*
- *Colrs is a finite set of colors, and C is the set of colorings over Colrs,*
- *F is a set of full colorings over Colrs,*
- *$\delta : \Sigma \times C \times \mathit{Dirs} \to 2^C$ is a partial function such that each coloring in $\delta(\sigma, \gamma, d)$ is full and extends γ,*
- *μ is a blind scanning strategy induced by some $\langle \eta, c_s, d_s\rangle$ such that $\delta(\sigma, \gamma, d) \neq \emptyset$ implies $\eta(\Delta_\gamma, d) \neq \perp$.*

[3] In the definition, also d_i, D_i, and E_i depend on n and m. For better readability, this dependence is not explicit.

A Wang automaton can be seen as having a head that visits a picture, by moving from a position to an adjacent one, and coloring at each step the edges of the position it is visiting (in a sense, the element of $C \times Dirs$ are the *states* of the automaton). For each accepting computation, the automaton produces a Wang-tiled picture whose label is equal to the input picture. The movements of the head are lead by the scanning strategy μ, whereas the coloring operations the automaton performs are determined by a finite control formalized by function δ. Since the scanning strategy μ is blind, the automaton visits the picture positions independently of the input symbols, and only the choice of colors to assign to edges is nondeterministic.

More precisely, the behavior of a μ-directed Wang automaton over an input picture p can be described as follows. At the beginning, the head of the automaton points at the position in the starting corner c_s and the current direction is set to d_s. When the current direction is d, the head is pointing at position y, the pixel of p at position y is σ, and the colors of borders of y are summarized by γ, then let $d' = \eta(\Delta_\gamma, d)$ and $\gamma' \in \delta(\sigma, \gamma, d)$. Hence the automaton may execute this move: apply γ' to the borders of y, set the current direction to d', and move to the position $y \boxplus d'$. If no move is possible, the automaton halts. The input picture p is accepted if there is a computation such that the coloring of the final position is in F.

As illustrated in the following theorem, for nondeterministic Wang automata the choice of the scanning strategy (as long as it is polite) is not relevant from the point of view of the recognizing power of the device. In the next section we will show that this is no longer true when determinism is concerned.

Theorem 1. *For every polite scanning strategy μ, we have $\mathcal{L}(\mu\text{-}WA) = REC$.*

Proof. REC being generated by Wang systems [10], the result is proved if we show that, for every polite μ, μ-directed Wang automata are equivalent to Wang systems.

First let $A = \langle \Sigma, Colrs, \delta, \mu, F \rangle$ be a μ-WA recognizing a language L. Then, define the Wang system $\omega = (Colrs \times Dirs, \Sigma, \Phi)$ by setting, for every $\gamma' \in \delta(\sigma, \gamma, d)$ and $d' = \eta(\Delta_\gamma, d)$,

$$(\gamma'(\mathrm{l}), a(\mathrm{l}))\ \underset{(\gamma'(\mathrm{b}), a(\mathrm{b}))}{\overset{(\gamma'(\mathrm{t}), a(\mathrm{t}))}{\boxed{\quad \sigma \quad}}}\ (\gamma'(\mathrm{r}), a(\mathrm{r})) \in \Phi,$$

where $a(x)$ may represent current direction d or next direction d', i.e.

$$a(x) = \begin{cases} d' & \text{if } x = d' \\ d & \text{if } x = -d \\ \perp & \text{otherwise,} \end{cases} \qquad \text{where} \qquad -\mathrm{b} = \mathrm{t}, -\mathrm{t} = \mathrm{b}, -\mathrm{l} = \mathrm{r}, -\mathrm{r} = \mathrm{l}.$$

Together with their labels, these labeled Wang tiles carry two pieces of information: the colors assigned by the automaton and the path followed by the head of the automaton, corresponding to the scanning strategy μ. One can verify that each Wang-tiled picture over Φ corresponds to an accepting computation of the automaton. Hence, the language generated by ω is L.

Vice versa, let $\omega = (Colrs \times Dirs, \Sigma, \Phi)$ be a Wang system recognizing a language L. Then, take any polite scanning strategy μ, and define the μ-WA $A = \langle \Sigma, Colrs, \delta, \mu, F \rangle$

where F is the set of all full colorings over $Colrs$, and δ is defined only for those triples (σ, γ, d) such that $\eta(\Delta_\gamma, d) \neq \perp$, and there exists some labeled Wang tile

$$c(\mathrm{l})\overset{c(\mathrm{t})}{\underset{c(\mathrm{b})}{\boxed{\;\sigma\;}}}c(\mathrm{r}) \in \Phi \qquad \text{with} \qquad \gamma(x) = c(x) \text{ if } \gamma(x) \neq \perp .$$

In this case, also set $\delta(\sigma, \gamma, d) = \gamma'$ where $\gamma'(x) = c(x)$ for every direction x. One can prove that the language generated by A is L and this concludes the proof. □

5 Determinism in Wang automata

In the framework of Wang automata, it is quite natural to introduce the concept of determinism:

Definition 5. *A μ-WA $\langle \Sigma, Colrs, \delta, \mu, F\rangle$ is deterministic if $\delta(\sigma, \gamma, d)$ has at most one element for every symbol $\sigma \in \Sigma$, coloring γ over Colrs, and direction d. Deterministic μ-WA are denoted by μ-DWA. The union of classes $\mathcal{L}(\mu\text{-DWA})$ over all polite μ is denoted by Scan-DREC.*

Example 3. Consider the language L_{half} presented in Example 1 and let $\sqcup$ be the scanning strategy that draws $\sqcup$-like paths, represented in Figure 1(d). Starting from the Wang system sketched in Example 2, one can define an equivalent $\sqcup$-DWA. Indeed, the Wang-tiled picture P in Equation (1) can be build deterministically from p by scanning it according to $\sqcup$.

Proposition 1. *For any polite μ, $\mathcal{L}(\mu\text{-DWA})$ is a boolean sub-class of REC.*

Proof (sketch). Given two μ-DWAs A_1 and A_2 recognizing two languages L_1 and L_2 respectively, one can reason as in [12, Theorem 7.4] to build a μ-DWA recognizing the intersection $L_1 \cap L_2$ (the set of colors will be the set of pairs (k_1, k_2) where each k_i is a color used by A_i).

The closure under complement is quite easy, too. Let $A = \langle \Sigma, Colrs, \delta, \mu, F\rangle$ be a μ-DWA recognizing L. We show how to build a deterministic Wang automaton recognizing the complement of L. First of all, modify δ so that any computation of A scans the whole input picture. For example, one can use a special color k to complete the computations that halt prematurely: for any coloring γ, let γ_k be the full coloring that extends γ with color k; then, whenever $\delta(\sigma, \gamma, d)$ is empty but $\eta(\Delta_\gamma, d) \neq \perp$, then set $\delta'(\sigma, \gamma, d) = \{\gamma_k\}$; also set $\delta'(\sigma, \gamma, d) = \{\gamma_k\}$ if γ already assigns k to some edge. Finally, let γ be in F' if and only of it is not in F. One can verify that $\langle \Sigma, Colrs \cup \{k\}, \delta', \mu, F'\rangle$ is a μ-DWA accepting the complement of L.

The closure under union is a consequence of the previous properties. □

Corollary 1. *Scan-DREC is closed under complement and rotation.*

Proof (sketch). The closure under complement is a straightforward consequence of the previous proposition. The closure under rotation is quite obvious, since one could easily define the *rotation of a scanning strategy* (and consequently the *rotation of a μ-WA*) and this operation preserves determinism. □

In particular, if ᘐ is the spiral scanning strategy represented in Figure 1(c), one can prove the following lemma.

Lemma 1. *$\mathcal{L}$(ᘐ-DWA) is closed under rotation.*

Proof (sketch). Let ᘐ′ be the scanning strategy obtained as the 90° clockwise rotation of ᘐ. Then any ᘐ′-DWA can be simulated by a ᘐ-DWA as follows: propagate the symbols in the first row downwards and check them in the second spiral round; the rest of the computation is as before. □

Proposition 2. *Snake-DREC ⊂ Scan-DREC ⊂ REC.*

Proof (sketch). Let τ be a snake-deterministic tiling system. First, one can slightly modify the construction in [10, Proposition 12] in order to build a Wang system equivalent to τ preserving its snake-determinism. Then, one can apply the construction of Theorem 1 (second part) to build an equivalent μ-WA, where μ is the boustrophedonic scanning strategy represented in Figure 1(b). Such automaton can be proved to be μ-DWA, hence by applying rotations one gets Snake-DREC ⊆ Scan-DREC.

To prove that the inclusion is proper, consider the language L of square pictures of even size with the first row like $x \cdot \bar{x}$, where $\bar{x}$ is the reverse of x, and let L^R be its intersection with all its rotations. Then, one can prove L is in Snake-DREC; however, by counting reasons it is possible to prove that L^R is not in Snake-DREC. On the contrary, improving the reasoning of Example 3, one can prove that $\mathcal{L}$(ᘐ-DWA) contains L, hence it contains also L^R by Lemma 1.

The last inclusion is a consequence of the previous proposition, since REC is not closed under complement. □

6 Conclusion and open problems

In this paper we have introduced a new model of 2D automata that recognize class REC and hence are strictly more powerful than traditional 4-way automata. The deterministic version of such a model is very natural and satisfies some interesting properties: it defines a class of picture languages which is closed under complement and extends some relevant subclasses of REC already studied in the literature. We conclude by stating some open problems concerning determinism in 2D.

Is Scan-DREC closed under union or intersection? Notice that the argument in proof of Theorem 1 cannot be applied when we have to intersect languages that are recognized by DWAs directed according to different scanning strategies.

Which is the relation among Snake-DREC, $\mathcal{L}$(ᘐ-DWA) and $\mathcal{L}$(⊔-DWA)? We have some examples that distinguish these classes: for instance the language L^R used to prove Proposition 2 is in $\mathcal{L}$(ᘐ-DWA) but not in Snake-DREC. However we do not know whether these classes are included one in another.

Which is the relation between Scan-DREC and the class of languages recognized by deterministic 4-way automata? We know that the latter class is incomparable to both Diag-DREC and Snake-DREC. But the language that separates them and is not in Snake-DREC is again the one used to prove Proposition 2, which is in Scan-DREC.

References

1. M. Anselmo, D. Giammarresi, and M. Madonia. From determinism to non-determinism in recognizable two-dimensional languages. In *Proc. DLT 2007*, volume 4588 of *Lecture Notes in Computer Science*, pages 36–47. Springer, 2007.
2. M. Anselmo, D. Giammarresi, and M. Madonia. Tiling automaton: A computational model for recognizable two-dimensional languages. In *Proc. CIAA 2007*, volume 4783 of *Lecture Notes in Computer Science*, pages 290–302. Springer, 2007.
3. A. Bertoni, M. Goldwurm, and V. Lonati. On the complexity of unary tiling-recognizable picture languages. *Fundamenta Informaticae*, 91(2):231–249, 2009.
4. B. Borchert and K. Reinhardt. Deterministically and sudoku-deterministically recognizable picture languages. In *Proc. LATA 2007*, 2007.
5. S. Bozapalidis and A. Grammatikopoulou. Recognizable picture series. *Journal of Automata, Languages and Combinatorics*, 10(2/3):159–183, 2005.
6. R. Brijder and H. J. Hoogeboom. Perfectly quilted rectangular snake tilings. *Theoretical Computer Science*, 410(16):1486–1494, 2009.
7. A. Cherubini, S. Crespi Reghizzi, and M. Pradella. Regional languages and tiling: A unifying approach to picture grammars. In *Proc. MFCS 2008*, volume 5162 of *Lecture Notes in Computer Science*, pages 253–264. Springer, 2008.
8. A. Cherubini and M. Pradella. Picture languages: From Wang tiles to 2D grammars. In S. Bozapalidis and G. Rahonis, editors, *Proc. CAI 2009*, volume 5725 of *Lecture Notes in Computer Science*, pages 13–46. Springer, 2009.
9. S. Crespi Reghizzi and M. Pradella. Tile Rewriting Grammars and Picture Languages. *Theoretical Computer Science*, 340(2):257–272, 2005.
10. L. de Prophetis and S. Varricchio. Recognizability of rectangular pictures by Wang systems. *Journal of Automata, Languages and Combinatorics*, 2(4):269–288, 1997.
11. D. Giammarresi and A. Restivo. Recognizable picture languages. *International Journal Pattern Recognition and Artificial Intelligence*, 6(2-3):241–256, 1992. Special Issue on *Parallel Image Processing*.
12. D. Giammarresi and A. Restivo. Two-dimensional languages. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, volume 3, Beyond Words, pages 215–267. Springer-Verlag, Berlin, 1997.
13. K. Inoue and A. Nakamura. Some properties of two-dimensional on-line tessellation acceptors. *Information Sciences*, 13:95–121, 1977.
14. K. Inoue and I. Takanami. A survey of two-dimensional automata theory. *Information Sciences*, 55(1-3):99–121, 1991.
15. K. Lindgren, C. Moore, and M. Nordahl. Complexity of two-dimensional patterns. *Journal of Statistical Physics*, 91(5-6):909–951, June 1998.
16. V. Lonati and M. Pradella. Snake-deterministic tiling systems. In *Proc. MFCS 2009*, volume 5734 of *Lecture Notes in Computer Science*, pages 549–560. Springer, 2009.
17. O. Matz. On piecewise testable, starfree, and recognizable picture languages. In M. Nivat, editor, *Proc. FoSSaCS 1998*, volume 1378 of *Lecture Notes in Computer Science*, pages 203–210. Springer, 1998.