Discrete Mathematics and Theoretical Computer Science DMTCS vol. **12:4**, 2010, 73–94

Deterministic recognizability of picture languages with Wang automata[†]

Violetta Lonati[1 ‡] and Matteo Pradella[2 §]

[1]*Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, Italy. Email:* `lonati@dsi.unimi.it`
[2]*IEIIT, Consiglio Nazionale delle Ricerche, Milano, Italy. Email:* `matteo.pradella@polimi.it`

received 30th November 2009, revised 29th July 2010, accepted 5th October 2010.

We present a model of automaton for picture language recognition, called Wang automaton, which is based on labeled Wang tiles. Wang automata combine features of both online tessellation acceptors and 4-way 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. 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 introduce a natural notion of determinism for Wang automata, and study the resulting class, extending the more traditional approach of diagonal-based determinism, used e.g. by deterministic tiling systems. In particular, we prove that the concept of row (or column) ambiguity defines the class of languages recognized by Wang automata directed by boustrophedonic scanning strategies.

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

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 rather rich of models, see e.g. [17, 15, 22, 11, 7, 12]. Here we regard class REC, the class of *tiling recognizable* languages, introduced in [15] 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 acceptors* [16], *tiling systems* [14], or *Wang systems* [13].

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, i.e

[†]This work includes the results presented in the two distinct papers [20] and [21].
[‡]Supported by the MIUR PRIN project "Mathematical aspects and emerging applications of automata and formal languages".
[§]Supported by the MIUR PRIN project "D-ASAP: Dependable Adaptable Software Architecture for Pervasive Computing".

a particular kind of tiles called *labeled Wang tiles*. Wang automata combine features of both online tessellation acceptors [16] and 4-ways automata [17]: as in online tessellation acceptors, 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 [3, 10]. 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 (a natural scanning strategy used by many algorithms on pictures and 2D arrays, such as shearsort [6, 3, 7]), spirals, and others.

Differently from 4-way automata, Wang automata directed by polite scanning strategies visit each position exactly once, and the next position function does not depend on the input symbol, like traditional finite state automata on strings. Nonetheless, we prove here that this kind of automata are equivalent to tiling systems, thus strictly more powerful than 4-way automata [15].

For string regular languages, two central notions are those of *determinism* and *unambiguity*. In two dimensions, the concept of unambiguity is straightforward and yields to class UREC [14]. UREC defines unambiguously tiling-recognizable languages, whose pictures are the projection of a unique element in the corresponding local language. In an effort to go towards determinism, the authors of [1] introduced an intermediate notion of "line" unambiguity, embodied in classes Row-UREC and Col-UREC, and based on backtracking at most linear in one dimension of the picture.

The concept of determinism for picture languages is still far from being well understood. The most relevant difficulty is that in 2D any notion of determinism seems to require some pre-established "scanning strategy" for reading the picture. Tiling systems are implicitly nondeterministic: REC is not closed under complement, and the membership problem is NP-complete [18]. Clearly, this latter fact severely hinders the potential applicability of the notation. The identification of a reasonably "rich" deterministic subset of REC would spur its application, since it would allow parsing linear with respect to the number of pixels of the input picture.

In past and more recent years, several different deterministic subclasses of REC have been studied, e.g. the classes defined by deterministic 4-way automata [17] or deterministic online tessellation acceptors [16]. This latter model inspired the notion of determinism of [1], that relies on four diagonal-based scanning strategies, each starting from one of the four corners of the picture. To mark this aspect, in this paper we will call the corresponding deterministic class Diag-DREC[(i)].

We cite also an interesting and radically different notion of determinism, proposed in [8], 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 [15], so it is hard to compare it with other approaches. For instance, while it is decidable to check if a tiling system is diagonal-deterministic, at present we do not know anything about decidability for [8].

A new kind of tile-based automaton, closely related to Wang automata, is presented in [9]: *quadrapolic automata* are similar to Wang automata, in that both use variant of labeled Wang tiles for working. The main difference is that quadrapolic automata do not read an input picture by following a scanning strategy - to our knowledge, no definition of determinism for such devices is proposed in the literature. Indeed,

(i) The original name is DREC.

a relevant aspect of Wang automata is the possibility to introduce a natural and decidable notion of determinism, yielding a proper subclass of REC, called Scan-DREC, which is closed under complement, rotation and mirror operations.

Among the different subclasses of Scan-DREC determined by different scanning strategies, we study here the class Snake-DREC of languages recognized by deterministic Wang automata directed by boustrophedonic scanning strategies. Snake-DREC can be defined equivalently in terms of tiling systems or online tessellation acceptors [20], and properly extends Diag-DREC while keeping some important closure properties. For instance, it is still closed under complement, rotation and symmetries. However, like Diag-DREC, it is not closed under intersection. Quite surprisingly, we found that this notion of determinism coincides with line unambiguity of Row-UREC (or Col-UREC): we show that the languages of this class can actually be recognized deterministically by following a boustrophedonic scanning strategy.

The paper is organized as follows. In Section 2 we recall some basic definitions and properties on two-dimensional languages, Wang systems, unambiguity, and determinism. In Section 3 we introduce polite scanning strategies and compare them with other 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. In Section 5 we introduce the concept of determinism natural in this framework, i.e. class Scan-DREC; we present its main properties and also compare some of its subclasses; we consider in particular class Snake-DREC $\subset$ Scan-DREC and prove its characterization in terms of line-unambiguity. Last, in Section 6 we draw some conclusions.

2 Preliminaries

The following definitions are taken and adapted from [15].

Let Σ be a finite alphabet. A two-dimensional array of elements of Σ is a *picture* over Σ. For $n, m \geq 1$, $\Sigma^{n,m}$ denotes the set of pictures of size (n, m), i.e. having n rows and m columns. The *support* of a picture of size (n, m) is the set $n \times m = \{1, 2, \ldots, n\} \times \{1, 2, \ldots, m\}$. $\# \notin \Sigma$ 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

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

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 say that $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$, and $(i, j + 1)$ are *adjacent* to position (i, j).

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.

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

Sometimes we need to consider *fragments*, that can be thought of as connected portions of pictures: fragments are not necessarily rectangular and can include also borders. E.g.: $\begin{matrix} & b \\ c & d \end{matrix}$ is a fragment of the picture $\begin{array}{|c|c|}\hline a & b \\ \hline c & d \\ \hline\end{array}$.

2.1 Tiling recognizable picture languages

An important class of two-dimensional languages is REC, i.e., the class of *tiling-recognizable languages*, originally defined in terms of tiling systems [14]. Here we define this class by using the equivalent notation introduced in [13], which is based on a variant of Wang tiles.

Labeled Wang tiles. Let Σ be a finite alphabet and K be a set of colors, containing the special color # representing borders. A *labeled Wang tile* (or *tile* for short) is a unitary square with colored edges and a *label* in Σ. Formally, a tile is an element $A = (a, t, l, r, b) \in \Sigma \times K^4$, where t, b, r, l represent the colors at top, bottom, right and left edges, respectively.

For better readability, we represent labeled Wang tiles as

$$A = \begin{matrix} & t & \\ l & \boxed{a} & r. \\ & b & \end{matrix} \tag{1}$$

In the rest of the paper, we will suppose that the Wang tilings cover the plane, so all the tiles that are beyond the borders of the considered picture will be the special tile $\sharp = (\#, \#, \#, \#, \#)$.

For any direction $d \in Dirs = \{\uparrow, \rightarrow, \leftarrow, \downarrow\}$, A_d is the color of the edge of A towards direction d. We also use $-d$ for referring to the direction opposite to d. Also, $\lambda(A)$ refers to the label of tile A. For example, in the case of A given by (1), $A_\downarrow = b$ and $\lambda(A) = a$. The set of tiles (including $\sharp$) with labels in Σ and colors in K is Σ_{4K}.

We also consider *partial* tiles, where some colors may be undefined: the set of partial tiles is denoted by Σ_K. The *domain* of a tile A is the set Δ_A of directions where A is defined. Given two partial tiles A, B bearing the same label, we say that B extends A if $B_d = A_d$ for every $d \in \Delta_A$. When we need to emphasize the fact that a tile is not partial, we will call it *complete*.

Wang pictures. Labeled Wang tiles in Σ_{4K} can be used to build pictures over Σ, by using colors to check compatibility: two tiles may be adjacent only if the color of the touching edges is the same.

A picture $P \in {\Sigma_{4K}}^{++}$ is called a *Wang picture* if all borders are colored with # and

$$\begin{aligned} P(i, j)_\downarrow &= P(i+1, j)_\uparrow \quad \text{for every} \quad 1 \le i < n, \\ P(i, j)_\rightarrow &= P(i, j+1)_\leftarrow \quad \text{for every} \quad 1 \le j < m, \end{aligned}$$

where (n, m) is the size of P. Next (on the left), the reader may find the example of a Wang picture of size $(2, 2)$. For better readability, we represent Wang pictures by writing each common color only once, as in the figure on the right:

$$
\begin{array}{|ccc|ccc|}
\hline
 & \# & & & \# & \\
\# & \boxed{a} & 4 & 4 & \boxed{b} & \# \\
 & 1 & & & 3 & \\
\hline
 & 1 & & & 3 & \\
\# & \boxed{b} & 2 & 2 & \boxed{a} & \# \\
 & \# & & & \# & \\
\hline
\end{array}
\qquad\qquad
\begin{array}{|ccccc|}
\hline
 & \# & & \# & \\
\# & \boxed{a} & 4 & \boxed{b} & \# \\
 & 1 & & 3 & \\
\# & \boxed{b} & 2 & \boxed{a} & \# \\
 & \# & & \# & \\
\hline
\end{array}
$$

The *label of a Wang picture P* over Σ_{4K} is the picture $p = \lambda(P) \in \Sigma^{++}$ having for pixels the labels of pixels of P, i.e., $p(i, j) = \lambda(P(i, j))$. For instance, the label of the example above is

$$
\begin{array}{|c|c|}
\hline
a & b \\
\hline
b & a \\
\hline
\end{array}\,.
$$

Sometimes we need to consider *Wang fragments*, i.e. picture fragments whose pixels are partial tiles with compatible borders. For instance, if

$$
A = \begin{array}{ccc} & \# & \\ 1 & \boxed{a} & 2 \\ & 3 & \end{array}, \quad
B = \begin{array}{ccc} & \# & \\ 2 & \boxed{b} & \# \\ & 4 & \end{array}, \quad \text{and } C = \begin{array}{ccc} & 4 & \\ & \boxed{c} & \# \\ & & \end{array},
$$

then $\begin{array}{cc} A & B \\ & C & \sharp \end{array}$ is the following Wang fragment:

$$
\begin{array}{ccccccc}
 & \# & & \# & & & \\
1 & \boxed{a} & 2 & \boxed{b} & \# & & \\
 & 3 & & 4 & & \# & \\
 & & & \boxed{c} & \# & \boxed{\#} & \# \\
 & & & \# & & \# &
\end{array}\,.
$$

Given $\Theta \subseteq \Sigma_{4K}$, the set of Wang pictures (resp. Wang fragments) built with tiles in Θ is denoted by $L_{\mathcal{W}}(\Theta)$ (resp. $F_{\mathcal{W}}(\Theta)$). Any Wang fragment is called a *Wang tiling* of its label.

Wang systems. A *Wang system* is a triple $\omega = \langle \Sigma, K, \Theta \rangle$, where Σ is a finite alphabet, K is a set of colors, Θ is a subset of Σ_{4K}. The language generated by ω is the language $L(\omega) \subseteq \Sigma^{++}$ of the labels of all Wang pictures in $L_{\mathcal{W}}(\Theta)$. Notice that a picture $p \in L(\omega)$ may have more than one Wang tiling in ω. REC is the class of picture languages generated by Wang systems.

Example 1 *The language L_{center} of square pictures over $\{0, 1\}$, with odd size greater than 2 and having 1 only in the center, is generated by the Wang system $\langle \Sigma, K, \Theta \rangle$, where $K = \{\#, \times, \diagdown, \diagup, \bullet\}$ and Θ is the set of Wang tiles contained in picture P depicted in Figure 1.*

Notice that it is straightforward to extend the previous Wang system to define the language L'_{center} of square pictures with odd size, and having 1 not only in the center, but possibly elsewhere. E.g., we may set

$$
\Theta' = \left\{ \begin{array}{ccc} & t & \\ l & \boxed{a} & r \\ & b & \end{array} \,\middle|\, a \in \{0, 1\} \text{ and } \begin{array}{ccc} & t & \\ l & \boxed{0} & r \\ & b & \end{array} \in \Theta \right\} \cup \left\{ \begin{array}{ccc} & \bullet & \\ \diagdown & \boxed{1} & \diagup \\ & \times & \end{array} \right\}.
$$

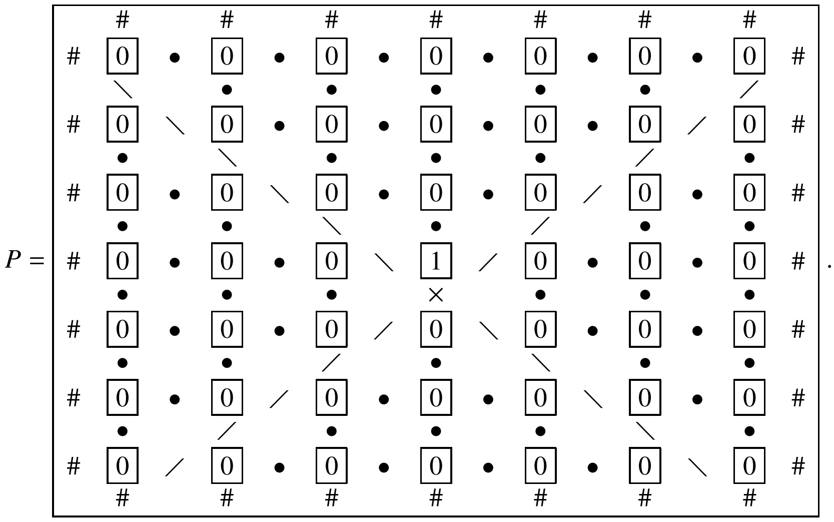

Figure 1: Instance picture for tile-set Θ of Example 1.

Example 2 *Consider the language $L_{half} \subset \Sigma^{++}$ of pictures of size $(n, m), n \geq m \geq 4$, with the first row like $w \cdot \bar{w}$, where $\bar{w}$ is the reverse of w. Then L_{half} is recognized by the Wang system $\langle \Sigma, K, \Theta \rangle$ where $K = \Sigma \cup \{\bullet, \#\}$ and*

$$\Theta = \left\{ \begin{array}{cccccc} \begin{array}{c}\#\\ \#\,\boxed{x}\,\bullet\\ x\end{array}, & \begin{array}{c}x\\ \#\,\boxed{y}\,\bullet\\ x\end{array}, & \begin{array}{c}x\\ \#\,\boxed{y}\,x\\ \#\end{array}, & \begin{array}{c}\bullet\\ x\,\boxed{y}\,x\\ \#\end{array}, & \begin{array}{c}x\\ x\,\boxed{y}\,\#\\ \#\end{array}, & \begin{array}{c}x\\ \bullet\,\boxed{y}\,\#\\ x\end{array}, \\ \begin{array}{c}\#\\ \bullet\,\boxed{x}\,\#\\ x\end{array}, & \begin{array}{c}\#\\ \bullet\,\boxed{x}\,\bullet\\ x\end{array}, & \begin{array}{c}x\\ \bullet\,\boxed{y}\,\bullet\\ x\end{array}, & \begin{array}{c}x\\ \bullet\,\boxed{y}\,x\\ \bullet\end{array}, & \begin{array}{c}\bullet\\ x\,\boxed{y}\,x\\ \bullet\end{array}, & \begin{array}{c}x\\ x\,\boxed{y}\,\bullet\\ \bullet\end{array} \end{array} \;\middle|\; x, y \in \Sigma \right\}.$$

The symbols on borders are used to connect each letter in w to the corresponding letter in $\bar{w}$, along nested paths following a U-like form. In Figure 2 we show an example of picture $p \in L_{half}$, together with its Wang tiling $P \in L_{\mathcal{W}}(\Theta)$ (the colors in P are used in the figure only to emphasize the U-like form of the resulting paths).

2.2 Unambiguous and diagonal-deterministic languages

Here we present the notions of unambiguity and determinism proposed in [14] and [1]. We adapt the basic definitions, results, and examples, proposing them with the notation of Wang systems.

Unambiguity. UREC can be defined as the class of tiling-recognizable languages whose pictures admits exactly one Wang tiling [14]. Formally, we say that a Wang system $\langle \Sigma, K, \Theta \rangle$ is *unambiguous* if, for every picture $p \in L(\omega)$ there exists exactly one $P \in L_{\mathcal{W}}(\Theta)$ such that $p = \lambda(P)$. UREC is the class of all languages generated by unambiguous Wang systems. It is known that UREC $\subset$ REC and, since it is undecidable whether a tiling system is unambiguous [5], it is also undecidable whether a Wang system is unambiguous.

$$
p = \begin{array}{|c|c|c|c|c|c|}
\hline
a & b & c & c & b & a \\ \hline
b & a & b & a & a & c \\ \hline
b & a & b & c & a & a \\ \hline
c & a & b & a & a & a \\ \hline
b & b & c & b & a & c \\ \hline
a & b & b & a & b & c \\ \hline
\end{array} \in \Sigma^{++}, \quad
P = \begin{array}{|ccccccccccccc|}
\hline
 & \# & & \# & & \# & & \# & & \# & & \# & \\
\# & \boxed{a} & \bullet & \boxed{b} & \bullet & \boxed{c} & \bullet & \boxed{c} & \bullet & \boxed{b} & \bullet & \boxed{a} & \# \\
 & a & & b & & c & & c & & b & & a & \\
\# & \boxed{b} & \bullet & \boxed{a} & \bullet & \boxed{b} & \bullet & \boxed{a} & \bullet & \boxed{a} & \bullet & \boxed{c} & \# \\
 & a & & b & & c & & c & & b & & a & \\
\# & \boxed{b} & \bullet & \boxed{a} & \bullet & \boxed{b} & \bullet & \boxed{c} & \bullet & \boxed{a} & \bullet & \boxed{a} & \# \\
 & a & & b & & c & & c & & b & & a & \\
\# & \boxed{c} & \bullet & \boxed{a} & \bullet & \boxed{b} & c & \boxed{a} & \bullet & \boxed{a} & \bullet & \boxed{a} & \# \\
 & a & & b & & \bullet & & \bullet & & b & & a & \\
\# & \boxed{b} & \bullet & \boxed{b} & b & \boxed{c} & b & \boxed{b} & b & \boxed{a} & \bullet & \boxed{c} & \# \\
 & a & & \bullet & & \bullet & & \bullet & & \bullet & & a & \\
\# & \boxed{a} & a & \boxed{b} & a & \boxed{b} & a & \boxed{a} & a & \boxed{b} & a & \boxed{c} & \# \\
 & \# & & \# & & \# & & \# & & \# & & \# & \\
\hline
\end{array}
$$

Figure 2: Example picture in L_{half} (left), and its Wang tiling (right).

Diagonal determinism. The notion of determinism proposed in [1] is inspired by the deterministic version of online tessellation acceptors [16], which are directed according to a corner-to-corner direction (namely, from top-left to bottom-right, or tl2br).

Consider a scanning strategy that respects the tl2br direction: any position (i, j) is read only if all the positions that are above and to the left of (i, j) have already been read. Roughly speaking, tl2br determinism means that any accepted picture $p \in \Sigma^{++}$ admits exactly one Wang tiling $P' \in L_{\mathcal{W}}(\Theta)$, which can be build deterministically when scanning p with any such strategy. Formally, a Wang system $\omega = \langle \Sigma, K, \Theta \rangle$ is called tl2br-*deterministic* if, for any $x, y \in K \cup \{\#\}$ and $a \in \Sigma$, there exists at most one tile

$$
\begin{array}{ccc} & x & \\ y & \boxed{a} & u \\ & v & \end{array} \in \Theta.
$$

By rotation, one can define *d-deterministic* Wang systems (d-DWS) for any corner-to-corner direction d in {tl2br, tr2bl, bl2tr, br2tl}, where t, b, l, and r stand as usual for top, bottom, left, and right, respectively.

We use Diag-DREC to denote the family of languages recognized by some d-DWS (for every corner-to-corner direction d).

Example 3 *The WS of Example 1 is not* tl2br-*deterministic as testified by tiles* $\begin{array}{ccc} & \bullet & \\ \bullet & \boxed{0} & \bullet \\ & \bullet & \end{array}$ *and* $\begin{array}{ccc} & \bullet & \\ \bullet & \boxed{0} & \diagup \\ & \diagup & \end{array}$. *Actually, language L_{center} admits a* tl2br-*deterministic WS and hence is in Diag-DREC [4].*

Example 4 *The WS of Example 2 is not* tl2br-*deterministic as testified by tiles* $\begin{array}{ccc} & x & \\ \bullet & \boxed{y} & \bullet \\ & x & \end{array}$ *and* $\begin{array}{ccc} & x & \\ \bullet & \boxed{y} & x \\ & \bullet & \end{array}$.

Line unambiguity. Consider the side-to-side direction t2b (from top to bottom). A Wang system $\langle K, \Sigma, \Theta \rangle$ is called t2b-*unambiguous* if, for any Wang fragment $\mathbf{X} = X_1 X_2 \cdots X_m \in \Theta^{1,m} \cup \{\sharp\}^{1,m}$ and $\mathbf{a} = (a_1, a_2, \cdots, a_m) \in \Sigma^{1,m}$, there exists at most one Wang fragment $\mathbf{A} = A_1 A_2 \cdots A_m \in \Theta^{1,m}$ such that

$$\lambda(\mathbf{A}) = \mathbf{a} \quad \text{and} \quad \begin{array}{cccccc} \sharp & X_1 & X_2 & \dots & X_m & \sharp \\ \sharp & A_1 & A_2 & \dots & A_m & \sharp \end{array} \in F_{\mathcal{W}}(\Theta) \tag{2}$$

Similar properties define d-unambiguous Wang systems (d-UWS) for any side-to-side direction $d \in \{\text{t2b}, \text{b2t}, \text{l2r}, \text{r2l}\}$.

Row-UREC (resp. Col-UREC) denotes the class of *row-unambiguous* (resp. *column-unambiguous*) languages, i.e., the languages generated by t2b-UWS or b2t-UWS (resp. l2r-UWS or r2l-UWS). The following relation holds, with all strict inclusions [1]:

$$\text{Diag-DREC} \subset (\text{Col-UREC} \cap \text{Row-UREC}) \subset (\text{Col-UREC} \cup \text{Row-UREC}) \subset \text{UREC}. \tag{3}$$

In the rest of the paper we will use the informal term *line unambiguity* for referring both to row and to column unambiguity.

Example 5 *The WS of Example 1 is d-unambiguous for any side-to-side direction d.*

Example 6 *The WS of Example 2 is not d-unambiguous for any side-to-side direction d: starting from the bottom of the picture, one does not know how to color the last row; starting from the top, one does not know where the inner U turns; starting from the side, one does not know, when moving upwards from an U to the next inner one, how to color it.*

3 Two-dimensional scanning strategies

The notions of determinism in [1, 2, 10, 20] are all based on some fixed pre-established scanning strategies. This approach is rooted in a natural analogy with string languages: for instance finite state automata read their input string using a fixed strategy, usually left-to-right. The ability of moving the input head depending on the actual content of the string makes the device more powerful (e.g. Turing Machines exhibit this behavior).

Our approach basically follows the same vein, so we are going to consider strategies that do not depend on the content of the input picture. This means that the order in which we are visiting two different pictures, that have the same dimensions, is the same.

Scanning strategies are defined in terms of partial functions; for a partial function f, if the value of f at t is not defined, then we write $f(t) = \perp$.

Definition 3.1 *A* scanning strategy *is a family*

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

where each $\mu_{n\times m}$ is a partial function such that $\mu_{n\times m}(t) \neq \perp$ for some t implies $\mu_{n\times m}(s) \neq \perp$ for every $1 \leq s < t$; $\mu_{n\times m}$ is called the scanning function over support *$n \times m$. A scanning strategy is said to be* continuous *if, for every t, n, and m, $\mu_{n\times m}(t+1)$ is adjacent to $\mu_{n\times m}(t)$, provided they are both defined; it is said to be* one-pass *if each scanning function $\mu_{n\times m}$ restricted to $\{1, 2, \ldots, nm\}$ is a bijection and $\mu_{n\times m}(t) = \perp$ for every $t > nm$.*

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

Some one-pass scanning strategies are illustrated in Figure 3. Actually they are not fully defined: only the function $\mu_{3\times4}$ on domain 3×4 is depicted, whereas the other functions should be defined analogously; each position c in 3×4 contains the number t such that $c = \mu_{3\times4}(t)$.

The strategies from (a) to (e) are all continuous: (a) has a boustrophedonic behavior, (b) draws nested L-like path, (c) draws nested U-like paths, (d) has a spiral behavior, (e) combines the behavior of (a) in the first half of the picture and (d) in the second one. The strategy (f) is not continuous and visits one row after the other, from left to right and from top to bottom.

(a) boustrophedonic snake ($\mathcal{S}$)

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

(b) L-like

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

(c) U-like ($\mathcal{U}$)

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

(d) spiral ($\mathcal{C}$)

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

(e) mixed

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

(f) row by row

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

Figure 3: Some one-pass scanning strategies: the number in each pixel denotes its scanning order.

As it is, this definition is quite general, and its generality may be source of issues. For instance, it is not uniform with respect to picture size: if we consider strings (i.e. $n = 1$) we could define strategies going from left to right for odd values of m, and from right to left otherwise. We could also exploit the knowledge of the input picture size to "jump", e.g. reading first the left-most (or first) character, then the last, then the second, then the last but one, and so on. Clearly this kind of strategy can be used to recognize strings like a^nb^n with a finite state device. We believe that this excessive flexibility is given by a kind of knowledge of the input picture which has a "global" flavor: we can rely on the exact size of the input picture, before even reading it.

In our opinion 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 3, (f) 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. 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 $\eta : 2^{Dirs} \times Dirs \to Dirs$ such that $\eta(D, d) = \perp$ if $-d \notin D$. Informally, η is used to chose where to go next: for a given position, we have a set of already considered edges, given by the set D of directions, and d, the direction from the "last-considered" edge; then $\eta(D, d)$ is the direction towards the position to visit next. Clearly, if $|D| = 1$, then d is the unique element of D; if $|D| = 3$, then d' is uniquely determined.

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 support $n \times m$, 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 support $n \times m$, and we set $d_1 = d_s$.

- The inductive definition(ii) of $\mu_{n\times m}(t+1)$ for $t \geq 1$ is given by:

$$D_t = \{d \in Dirs : (\mu_{n\times m}(t))_d \in E_t\} \qquad E_{t+1} = E_t \cup \mathrm{edges}(\mu_{n\times m}(t))$$

$$d_{t+1} = \eta(D_t, d_t) \qquad \mu_{n\times m}(t+1) = \mu_{n\times m}(t) \boxplus d_{t+1}$$

  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 3.2 *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 support; 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 3.3 *A scanning strategy is called* polite *if it is blind and one-pass.*

Example 7 *The one-pass scanning strategies represented in Figure 3(a-d) are all blind, hence polite. In particular we denote the strategies (a), (c), (d) as $\mathcal{S}$, $\mathcal{U}$, $\mathcal{C}$, respectively.*

For instance, $\mathcal{C}$ is induced by the triple $\langle \eta_{\mathcal{C}}, \mathrm{tl}, \rightarrow\rangle$, where its next-position function is shown in the following table. For better readability, set of directions and incoming direction (i.e. $D \in 2^{Dirs}$, $d \in Dirs$)

(ii) In the definition, also d_t, D_t, and E_t depend on n and m; for better readability, this dependence is not explicit. We also agree that the value of any expression containing $\perp$ is still $\perp$.

are graphically depicted together in a partially outlined rectangle, where marked sides correspond to elements of D. E.g. $D = \{\uparrow, \leftarrow\}, d = \rightarrow$ *is shown as* $\rightarrow$.

D, d	$\eta_{\mathcal{C}}(D, d)$
$\rightarrow$	$\downarrow$
$\downarrow$	$\downarrow$
$\downarrow$	$\rightarrow$
$\rightarrow$	$\rightarrow$

D, d	$\eta_{\mathcal{C}}(D, d)$
$\rightarrow$	$\uparrow$
$\uparrow$	$\uparrow$
$\uparrow$	$\leftarrow$
$\leftarrow$	$\leftarrow$

D, d	$\eta_{\mathcal{C}}(D, d)$
$\leftarrow$	$\downarrow$
$\downarrow$	$\downarrow$
$\uparrow$	$\uparrow$
$\leftarrow$	$\leftarrow$

D, d	$\eta_{\mathcal{C}}(D, d)$
$\rightarrow$	$\rightarrow$
$\rightarrow$	$\downarrow$

It is interesting to note that any polite scanning strategy can be described in terms of Strategies (a-d) of Figure 3 as proved in [19].

Related works. In the literature on 2D languages, two recent works considered the problem of defining pre-established scanning strategies for pictures, namely [2] and [10].

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, in a sense like the one we introduced here in Definition 3.1, thus allowing complex behaviors. In practice, and like in our approach, scanning strategies presented and actually used in [2] are simpler, and do not exceed REC. Among these strategies there are some that we do not consider here, e.g. row-by-row, left-to-right scan, that we discard because is not continuous and rely on the knowledge of picture width. On the other hand, all these strategies are not more expressive than the ones considered here, as we will see next.

The very recent work [10] 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. This approach is similar to ours, in that it requires continuity, but different, because some knowledge of the picture size is used to define complex strategies such as Peano-Hilbert curves.

4 Wang automata

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

Definition 4.1 *A* μ-directed Wang automaton *(μ-WA) is a tuple* $\langle \Sigma, K, \delta, \mu, F \rangle$ *where:*

- Σ *is a finite input alphabet,*
- K *is a finite set of colors*
- $F \subset \Sigma_{4K}$,
- $\delta : \Sigma_K \times Dirs \to 2^{\Sigma_{4K}}$ *is a partial function such that each tile in* $\delta(A, d)$ *extends* A,
- μ *is a polite scanning strategy induced by some* $\langle \eta, c_s, d_s \rangle$ *such that* $\delta(A, d) \neq \emptyset$ *implies* $\eta(\Delta_A, d) \neq \perp$.

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 elements of $\Sigma_K \times Dirs$ are the *states* of the automaton). For each accepting computation, the automaton produces a Wang 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 polite and hence 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 towards position y, the pixel and the colors of borders of p at position y are summarized by A, then let $d' = \eta(\Delta_A, d)$ and $A' \in \delta(A, d)$. Hence the automaton may execute this move: color the borders of y according to A', set the current direction to d', and move to 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 borders of the final position are colored according to some Wang tile in F.

Example 8 *Consider the language L_{half} presented in Example 2. Starting from the Wang system sketched in the same example, one can define an equivalent $\mathcal{C}$-WA as described in Table 1. Note that $\delta(A,d)$ has at most one element, for any A, d, so in the table these are not represented as sets. For better readability, the table also presents the next-position function η. The set of accepting tiles is*

$$\left\{ \begin{array}{ccc} & x & \\ \bullet & \boxed{y} & \bullet \\ & x & \end{array} \;\middle|\; x \in K \right\}.$$

It is interesting to note that the set of complete tiles coincides with tile-set Θ of Example 2.

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

Proof: We show that, for every polite μ, μ-directed Wang automata are equivalent to Wang systems.

First let $\tau = \langle \Sigma, K, \delta, \mu, F \rangle$ be a μ-WA recognizing a language L. Then, given $A' \in \delta(A,d)$, let $d' = \eta(\Delta_A, d)$ and, for every direction x, set

$$f(x) = \begin{cases} d' & \text{if } x = d' \\ d & \text{if } x = -d \\ \perp & \text{otherwise.} \end{cases}$$

Now consider the labeled Wang tile $B \in \Sigma_{4(K \times Dirs)}$ defined by setting $\lambda(B) = \lambda(A')$ and $B_d = (A'_d, f(d))$ for every direction d. Together with its label, any such tile B carries simultaneously two kinds of information: B_d's give the colors assigned to borders by the automaton, and $f(d)$ gives the path followed by the head of the automaton, corresponding to the scanning strategy μ. Moreover, for every $A'' \in F$, consider all tiles

A	d	δ	η
$\begin{smallmatrix} & \# & \\ \# & \boxed{x} & \\ & & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & \# & \\ \# & \boxed{x} & \bullet \\ & x & \end{smallmatrix}$	$\downarrow$
$\begin{smallmatrix} & x & \\ \# & \boxed{y} & \\ & & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & x & \\ \# & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\downarrow$
$\begin{smallmatrix} & x & \\ \# & \boxed{y} & \\ & \# & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & x & \\ \# & \boxed{y} & x \\ & \# & \end{smallmatrix}$	$\rightarrow$
$\begin{smallmatrix} & & \\ x & \boxed{y} & \\ & \# & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & \bullet & \\ x & \boxed{y} & x \\ & \# & \end{smallmatrix}$	$\rightarrow$
$\begin{smallmatrix} & & \\ x & \boxed{y} & \# \\ & \# & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & x & \\ x & \boxed{y} & \# \\ & \# & \end{smallmatrix}$	$\uparrow$
$\begin{smallmatrix} & & \\ & \boxed{y} & \# \\ & x & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \# \\ & x & \end{smallmatrix}$	$\uparrow$

A	d	δ	η
$\begin{smallmatrix} & \# & \\ & \boxed{x} & \# \\ & x & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & \# & \\ \bullet & \boxed{x} & \# \\ & x & \end{smallmatrix}$	$\leftarrow$
$\begin{smallmatrix} & \# & \\ & \boxed{x} & \bullet \\ & & \end{smallmatrix}$	$\leftarrow$	$\begin{smallmatrix} & \# & \\ \bullet & \boxed{x} & \bullet \\ & x & \end{smallmatrix}$	$\leftarrow$
$\begin{smallmatrix} & \# & \\ \bullet & \boxed{x} & \bullet \\ & & \end{smallmatrix}$	$\leftarrow$	$\begin{smallmatrix} & \# & \\ \bullet & \boxed{x} & \bullet \\ & x & \end{smallmatrix}$	$\downarrow$
$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \\ & & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\downarrow$
$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \\ & \bullet & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & x \\ & \bullet & \end{smallmatrix}$	$\rightarrow$
$\begin{smallmatrix} & & \\ x & \boxed{y} & \\ & \bullet & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & \bullet & \\ x & \boxed{y} & x \\ & \bullet & \end{smallmatrix}$	$\rightarrow$

A	d	δ	η
$\begin{smallmatrix} & & \\ x & \boxed{y} & \bullet \\ & \bullet & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & x & \\ x & \boxed{y} & \bullet \\ & \bullet & \end{smallmatrix}$	$\uparrow$
$\begin{smallmatrix} & & \\ & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\uparrow$
$\begin{smallmatrix} & x & \\ & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\leftarrow$
$\begin{smallmatrix} & x & \\ & \boxed{y} & \bullet \\ & & \end{smallmatrix}$	$\leftarrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\leftarrow$
$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \bullet \\ & & \end{smallmatrix}$	$\leftarrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\downarrow$

Table 1: $\mathcal{C}$-WA for L_{half}: δ and η stand respectively for $\delta(A, d)$ and $\eta(\Delta_A, d)$; $x, y \in K$.

$C \in \Sigma_{4(K \times Dirs)}$, which are like A'', but have exactly one edge colored with an inbound arrow (while all the other second components are $\perp$).

Let Θ be the set of any such B and C. One can verify that each Wang picture over Θ corresponds to an accepting computation of the automaton. Hence, L is the language generated by the Wang system $\omega = (K \times Dirs, \Sigma, \Theta)$.

Vice versa, let $\omega = (K, \Sigma, \Theta)$ be a Wang system recognizing a language L. Then, take any polite scanning strategy μ, and define the μ-WA $\tau = \langle \Sigma, K, \delta, \mu, F \rangle$ where F is the set of all Wang tiles over K, and δ is defined by setting, for each direction d and partial tile A:

$$\delta(A, d) = \begin{cases} \{A' \in \Theta \mid A' \text{ extends } A\} & \text{if } \eta(\Delta_A, d) \neq \perp, \\ \emptyset & \text{otherwise.} \end{cases}$$

One can prove that the language generated by τ is L and this concludes the proof. $\square$

5 Determinism in Wang automata

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

Definition 5.1 *A μ-WA $\langle \Sigma, K, \delta, \mu, F \rangle$ is* deterministic *if $\delta(A, d)$ has at most one element for every tile $A \in \Sigma \times K^4$ and direction d. Deterministic μ-WA are denoted by μ-DWA. The union of classes $\mathcal{L}(\mu\text{-DWA})$ over all polite scanning strategies μ is denoted by Scan-DREC.*

If δ is the transition function of a μ-DWA, we shall write $\delta(A, d) = A'$ instead of $\delta(A, d) = \{A'\}$.

Example 9 *The $\mathcal{C}$-WA in Example 8 is deterministic.*

Example 10 *Language L_{half} presented in Example 2 can be recognized also by a $\mathcal{U}$-DWA (see Table 2). The set of accepting tiles is*

$$\left\{ \begin{smallmatrix} & \# & \\ \bullet & \boxed{x} & \bullet \\ & x & \end{smallmatrix} \;\middle|\; x \in K \right\}.$$

The first column in Table 2 is used for the outermost U, therefore going down first, then right, then up. The second column covers internal U's, following the direction down, then right, then up. The third column is for the other internal U's, i.e. following the direction down, then left, then up. As in Example 8, the set of complete tiles coincides with tile-set Θ of Example 2.

A	d	δ	η	A	d	δ	η	A	d	δ	η
$\begin{smallmatrix} & \# & \\ \# & \boxed{x} & \\ & & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & \# & \\ \# & \boxed{x} & \bullet \\ & x & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & \# & \\ \bullet & \boxed{x} & \\ & & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & \# & \\ \bullet & \boxed{x} & \bullet \\ & x & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & \# & \\ & \boxed{x} & \bullet \\ & & \end{smallmatrix}$	$\leftarrow$	$\begin{smallmatrix} & \# & \\ \bullet & \boxed{x} & \bullet \\ & x & \end{smallmatrix}$	$\downarrow$
$\begin{smallmatrix} & x & \\ \# & \boxed{y} & \\ & & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & x & \\ \# & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \\ & & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & x & \\ & \boxed{y} & \bullet \\ & & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\downarrow$
$\begin{smallmatrix} & x & \\ \# & \boxed{y} & \\ & \# & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & x & \\ \# & \boxed{y} & x \\ & \# & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \\ & \bullet & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & x \\ & \bullet & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & x & \\ & \boxed{y} & \bullet \\ & \bullet & \end{smallmatrix}$	$\downarrow$	$\begin{smallmatrix} & x & \\ x & \boxed{y} & \bullet \\ & \bullet & \end{smallmatrix}$	$\leftarrow$
$\begin{smallmatrix} & & \\ x & \boxed{y} & \\ & \# & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & \bullet & \\ x & \boxed{y} & x \\ & \# & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & & \\ x & \boxed{y} & \\ & \bullet & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & \bullet & \\ x & \boxed{y} & x \\ & \bullet & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & & \\ & \boxed{y} & x \\ & \bullet & \end{smallmatrix}$	$\leftarrow$	$\begin{smallmatrix} & \bullet & \\ x & \boxed{y} & x \\ & \bullet & \end{smallmatrix}$	$\leftarrow$
$\begin{smallmatrix} & & \\ x & \boxed{y} & \# \\ & \# & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & x & \\ x & \boxed{y} & \# \\ & \# & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & & \\ x & \boxed{y} & \bullet \\ & \bullet & \end{smallmatrix}$	$\rightarrow$	$\begin{smallmatrix} & x & \\ x & \boxed{y} & \bullet \\ & \bullet & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & & \\ \bullet & \boxed{y} & x \\ & \bullet & \end{smallmatrix}$	$\leftarrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & x \\ & \bullet & \end{smallmatrix}$	$\uparrow$
$\begin{smallmatrix} & & \\ & \boxed{y} & \# \\ & x & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \# \\ & x & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & & \\ & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & & \\ \bullet & \boxed{y} & \\ & x & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & x & \\ \bullet & \boxed{y} & \bullet \\ & x & \end{smallmatrix}$	$\uparrow$
$\begin{smallmatrix} & \# & \\ & \boxed{x} & \# \\ & x & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & \# & \\ \bullet & \boxed{x} & \# \\ & x & \end{smallmatrix}$	$\leftarrow$	$\begin{smallmatrix} & \# & \\ & \boxed{x} & \bullet \\ & x & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & \# & \\ \bullet & \boxed{x} & \bullet \\ & x & \end{smallmatrix}$	$\leftarrow$	$\begin{smallmatrix} & \# & \\ \bullet & \boxed{x} & \\ & x & \end{smallmatrix}$	$\uparrow$	$\begin{smallmatrix} & \# & \\ \bullet & \boxed{x} & \bullet \\ & x & \end{smallmatrix}$	$\rightarrow$

Table 2: $\mathcal{U}$-DWA for L_{half}: δ and η stand respectively for $\delta(A, d)$ and $\eta(\Delta_A, d)$; $x, y \in K$.

Proposition 5.1 *For any polite μ, $\mathcal{L}(\mu$-DWA$)$ is a boolean sub-class of REC.*

Proof: Given two μ-DWAs τ_1 and τ_2 recognizing two languages L_1 and L_2 respectively, one can reason as in [15, Theorem 7.4] to build a μ-DWA recognizing the intersection $L_1 \cap L_2$ (the set of colors will be $K_1 \times K_2$, where K_i is the set of colors used by τ_i).

The closure under complement is quite easy, too. Let $\tau = \langle \Sigma, K, \delta, \mu, F \rangle$ 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 τ scans the whole input picture. For example, one can use a special color $\diamond$ to complete the computations that halt prematurely: for any partial tile A, let $A^\diamond$ be the complete tile that extends A with color $\diamond$; then, whenever $\delta(A, d)$ is empty but $\eta(\Delta_A, d) \neq \perp$, then set $\delta'(A, d) = \{A^\diamond\}$; also set $\delta'(A, d) = \{A^\diamond\}$ if A already has color $\diamond$ on some edge. Finally, let A be in F' if and only of it is not in F. One can verify that $\langle \Sigma, K \cup \{\diamond\}, \delta', \mu, F' \rangle$ is a μ-DWA accepting the complement of L.

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

Corollary 5.1 *Scan-DREC is a proper subclass of REC, closed under complement, rotation and mirror.*

Proof: The closure under complement is a straightforward consequence of the previous proposition. REC being not closed under complement, by Theorem 4.1 we have Scan-DREC $\subset$ REC. The closure under rotation (resp. mirror) is quite obvious, since one could easily define the *rotation (resp. mirror) of a scanning strategy* (and consequently the *rotation (resp. mirror) of a μ-WA*) and this operation preserves determinism. □

Proposition 5.2 *There exists a language in $\mathcal{L}(\mathcal{C}$-DWA$)$ and not in $\mathcal{L}(\mathcal{S}$-DWA$)$.*

Proof: Consider the language L_{half} defined in Example 2 and let L be its intersection with its 180° rotation, i.e., the language of pictures of size $(n, m), n \geq m \geq 4$, with both the first and the last rows like $w \cdot \bar{w}$, where $\bar{w}$ is the reverse of w. We prove that $L \in \mathcal{L}(\mathcal{C}\text{-DWA}) \setminus \mathcal{L}(\mathcal{S}\text{-DWA})$.

First of all, notice that $\mathcal{L}(\mathcal{C}\text{-DWA})$ is closed under rotation. For instance, let $\mathcal{C}'$ be the scanning strategy obtained as the 90° counter-clockwise rotation of $\mathcal{C}$. Then any $\mathcal{C}'$-DWA can be simulated by a $\mathcal{C}$-DWA as follows: propagate the symbols in the first column rightwards and check them in the second spiral round; the rest of the computation is as before. Hence, L_{half} being in $\mathcal{L}(\mathcal{C}\text{-DWA})$ as shown in Example 8, we get that $L \in \mathcal{L}(\mathcal{C}\text{-DWA})$.

On the other hand, $L \notin \mathcal{L}(\mathcal{S}\text{-DWA})$ by counting reasons. Let us assume by contradiction that there exists a $\mathcal{S}$-DWA τ recognizing L, and consider the language $T_n \subseteq L$ formed by all square pictures of side size $2n$ having a fixed symbol in Σ everywhere except in the last row. For every $p \in T_n$, let $k(p)$ be the color assigned by τ to the left border of position $(2n, n+1)$. For n great enough, the number of words over Σ of length n is greater than the number of colors used by τ, hence there exist an integer $\bar{n} > 1$, two words $w_1 \neq w_2$ of length $\bar{n}$ and two pictures $p_1, p_2 \in T_{\bar{n}}$ such that the last row of p_i ends with w_i for $i = 1, 2$ and $k(p_1) = k(p_2)$. Consider the pictures p'_1 and p'_2 built from p_1 and p_2 by replacing their last rows with $\bar{w_1} \cdot w_1$ and $\bar{w_1} \cdot w_2$, respectively. Then, the computation of τ over both p'_1 and p'_2 ends at position $(2\bar{n}, 1)$ with the same tile, but clearly p'_1 is in L whereas p'_2 is not, a contradiction. □

Proposition 5.3 *There exists a language in $\mathcal{L}(\mathcal{S}$-DWA$)$ and not in $\mathcal{L}(\mathcal{C}$-DWA$)$.*

Proof: Consider the language $L_{\exists r=fr}$ of square pictures where there is at least one row that is equal to the first one.

On the one hand, it is very easy to define a $\mathcal{S}$-DWA for $L_{\exists r=fr}$, as a $\mathcal{S}$-DWA can propagate (by coloring) the symbols found in the first row to the rows below, one by one, and then check if at least one copy of the first row is present in the input picture.

On the other hand, we provide some intuition for the fact that $L_{\exists r=fr} \notin \mathcal{L}(\mathcal{C}\text{-DWA})$. Let us suppose that there is a $\mathcal{C}$-DWA, i.e. $\tau = \langle \Sigma, K, \delta, \mathcal{C}, F \rangle$, such that $\mathcal{L}(\tau) = L_{\exists r=fr}$. We now consider w.l.o.g a generic picture $p \in L_{\exists r=fr}$ having even side size, i.e. $2n$. We call w the first row of p, p_u its upper half, p_l its lower half (i.e. both having size $(n, 2n)$). It is easy to check if another copy of the row w is present in p_u (using a technique analogous to the one used before with $\mathcal{S}$-DWA). But, to check if w is present in p_l, following a spiral scanning strategy, one needs to represent its content in the coloring of the lower part. Every "turn" of the spiral is able to propagate a number of symbols of w that depends on a constant k which is proportional to $|K|/|\Sigma|$. Hence, to propagate w to the coloring of the lower part of p, the scanning of the input picture has to perform $2n/k$ "rounds", thus visiting completely the lower $2n/k$ rows of p_l. Let us call this last subpicture p'_l, that has size $(2n/k, 2n)$.

The following facts hold: (1) it is impossible to visit p'_l after the first $2n/k$ rounds of the scanning; (2) the coloring at positions corresponding to p'_l in p cannot encode the fact that w is (or is not) a row of p'_l, because before the first $2n/k$ rounds, the content of w was not completely available in the lower part of the picture; (3) it is impossible to represent the whole content of p'_l, made of $4n^2/k$ symbols, in the coloring of the tiles which are partial after the first $2n/k$ rounds of the input picture scanning, because the number of these tiles is proportional to n.

Therefore, let us suppose that w is present in p_u only as the first row, while it is not present in the upper $n - 2n/k$ rows of p_l. Because of facts (1-3), it is impossible for τ to state if p is in $L_{\exists r=fr}$, or not. □

Corollary 5.2 *$\mathcal{L}(\mathcal{S}$-DWA$)$ and $\mathcal{L}(\mathcal{C}$-DWA$)$ are incomparable.*

In particular we characterize the boustrophedonic scanning strategy $\mathcal{S}$ in terms of line unambiguity.

Definition 5.2 *Snake-DREC is the closure under rotation of $\mathcal{L}(\mathcal{S}$-DWA$)$.*

Notice that Snake-DREC was originally described by means of tiling systems or online tessellation acceptors [20]. The term snake comes from the boustrophedonic order of visiting of the pixels: if we draw the trajectory of the scanning strategy, it looks like a snake covering line by line (or column by column) the whole picture.

Theorem 5.1 *Snake-DREC coincides with class Row-UREC $\cup$ Col-UREC.*

Before proving this theorem, we summarize its main consequences as follows:

Corollary 5.3 *Snake-DREC is closed under complement, rotation and mirrors, but not under intersection. Moreover*

$$\textit{Diag-DREC} \subset \textit{Snake-DREC} \subset \textit{Scan-DREC} \subseteq \textit{UREC}.$$

Proof: Proposition 5.1 implies the closure under complement; the closure under rotation is obvious by definition; the closure under mirrors follows by the closure under mirrors of both Row-UREC and Col-UREC; $L_{\exists r=fr}$ is in Snake-DREC as shown in proof of Proposition 5.3, but its intersection with all its rotations is not [1].

The first proper inclusion is a straightforward consequence of the previous theorem and relation (3). Clearly Snake-DREC $\subseteq$ Scan-DREC $\subseteq$ UREC by definition. Snake-DREC is properly included in Scan-DREC: let L_{half} be the language defined in Example 2, and let L be its intersection with all its $90°$ rotations. Then L is in $\mathcal{L}(\mathcal{C}$-DWA$)$ since $\mathcal{L}(\mathcal{C}$-DWA$)$ is closed under rotation and $L_{\text{half}} \in \mathcal{L}(\mathcal{C}$-DWA$)$ by Proposition 5.2; on the other hand L is not in Snake-DREC for counting reasons similar to those used in Proposition 5.2. □

$$\begin{tikzcd}[column sep=small, row sep=small]
\sharp \arrow[r] \arrow[dr] \arrow[ddr] & A_1 \arrow[r] \arrow[dr] & A_2 \arrow[r] & A_3 \arrow[r] \arrow[dr] & A_4 \arrow[r] & \sharp \\
 & A'_1 \arrow[dr] & A'_2 \arrow[r] & A'_3 & A'_4 & \\
 & A''_1 \arrow[r] & A''_2 \arrow[r] & A''_3 & &
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small, row sep=small]
\sharp \arrow[r] \arrow[dr] \arrow[ddr] & A_1 \arrow[r] \arrow[dr] & A_2 \arrow[r] & A_3 \arrow[r] \arrow[dr] & A_4 \arrow[r] & \sharp \\
 & A'_1 & A'_2 \arrow[r] & A'_3 & A'_4 & \\
 & A''_1 & & & &
\end{tikzcd}$$

Figure 4: Graph (left) and tree (right) of the Wang tilings of row (a_1, a_2, a_3, a_4).

Proof of Theorem 5.1

We actually show that

$$\mathcal{L}(\mathcal{S}\text{-DWA}) = \mathcal{L}(\text{t2b-UWS}),$$

the proof of theorem follows by applying rotations. In one direction, the result is easy: starting from any $\mathcal{S}$-DWA and using the construction of Theorem 4.1 one obtains a Wang system which is t2b-unambiguous.

The converse is less intuitive; in order to prove it, from now on let $\omega = \langle \Sigma, K, \Theta \rangle$ be a t2b-UWS. We build a $\mathcal{S}$-DWA

$$\tau = \langle \Sigma, K', \delta, \mathcal{S}, F \rangle$$

that simulates ω. Before formally defining τ, let us first point out some important remarks.

Given any Wang fragment $\mathbf{X} = X_1 X_2 \ldots X_m \in \Theta^{1,m} \cup \{\sharp\}^{1,m}$ and $\mathbf{a} = (a_1, a_2, \ldots, a_m) \in \Sigma^{1,m}$, there exists at most one Wang fragment $\mathbf{A} = A_1 A_2 \ldots A_m \in \Theta^{1,m}$ satisfying relation (2). However, we have no guarantees that $\mathbf{A}$ can be built from left to right deterministically. For instance, for $m = 4$ and for some fixed $\mathbf{X}$, ω may allow the choices of Wang tiles represented in Figure 4 (left).
Clearly, ω being t2b-unambiguous, only one branch of the graph ends with $\sharp$: the one corresponding to $\mathbf{A}$. In the other cases, a backtracking linear in the length of the row is always sufficient to (eventually) determine $\mathbf{A}$.

Remark 1 *Since we are building a Wang tiling of a fixed row* $\boldsymbol{a}$*, at each position we can choose among tiles that all have the same label. E.g.,* $\lambda(A_1) = \lambda(A'_1) = \lambda(A''_1) = a_1$.

Remark 2 *Since* ω *is* t2b*-unambiguous, the branch corresponding to* $\boldsymbol{A}$ *cannot contain Wang tiles with in-degree greater that one (otherwise there would exist two different Wang tilings of row* $\boldsymbol{a}$ *satisfying relation (2), a contradiction). In other words, if two or more branches "collapse", the successive tiles may be ignored. Then we can assume that the graph of the Wang tilings of* $\boldsymbol{a}$ *is actually a tree, where each tile has exactly one predecessor. We call it the* tree of the Wang tiling of **a** in Θ. *For the previous example, the tree is depicted in Figure 4 (right).*

Similar remarks hold if we try to build the Wang tiling of row **a** from right to left.

To simulate ω deterministically on a picture p, we proceed as follows. Let P be the unique Wang tiling of p in $L_{\mathcal{W}}(\Theta)$. To locally represent the tree of Wang tilings of the current row, we store at each position the set α of tiles in Θ that may appear at the corresponding position in P. Notice that any such set α contains tiles having the same label, by Remark 1; moreover, each tile in any such set α admits exactly one predecessors in the tree, by Remark 2. The set of tile-sets α containing tiles with the same label is denoted by Φ; $\lambda(\alpha)$ is the label of any tiles in α. We also use the following convention: $a, b, \ldots$ are used to denote symbols in Σ; $A, B, \ldots$ are used to denote (possibly partial) tiles in Σ_K; and $\alpha, \beta, \ldots$ are used to denote elements of Φ.

When scanning rightwards the first row of p, we compute and keep trace of the tree of its Wang tilings; at the end of the row, we determine which branch is successful (i.e, the one that corresponds to the first row of P). When scanning the second row backwards, we use such information (together with the traces we left in the previous scan) to reconstruct backwards the successful branch and, at the same time, we compute and keep trace of the tree of the Wang tilings of the current row. This procedure continues till the last row has been scanned. So, the set of colors K' of τ must contain both pieces of information. This leads to the following definition:

$$K' = \{\#\} \cup \Phi \cup (\Theta \times \Phi).$$

Colors in Φ are used for the tiles in the first row and, in other rows, only at top and bottom of tiles, whereas colors in $\Theta \times \Phi$ are used at left and right: the role of color $(A, \beta) \in \Theta \times \Phi$ is the following: A is the correct Wang tile that one should have chosen in the position immediately above the current one, whereas β keeps trace of all possible Wang tiles that may appear in the current position.

We are now able to define the (deterministic) transition function δ of τ. It has to be a partial function

$$\delta : \Sigma_{K'} \times Dirs \to \Sigma_{4K'}$$

that satisfies two conditions: i) each Wang tiles in $\delta(W, d)$ extends W; ii) $\delta(W, d) \neq \perp$ implies $\eta(\Delta_W, d) \neq \perp$, where η is the next-position function associated with $\mathcal{S}$. Thus, it is sufficient to define $\delta(W, d)$ in the cases represented in Table 3: the left part of the table is used for the first and last row of the input picture, whereas the right part covers the other rows to be visited rightwards or leftwards.

In Table 3, for each $W \in \Sigma_{K'}$ such that $\delta(W, d) \neq \perp$, a picture fragment $f(W)$ is introduced; moreover tile-sets α_0, α_1, α and tile C appearing in the table are defined, for each W, by setting:

$$\begin{aligned} \alpha_0 &= \{A \in \lambda^{-1}(a) \mid f(W) \in F_{\mathcal{W}}(\Theta)\}, && (4)\\ \alpha_1 &= \{A \in \lambda^{-1}(a) \mid \exists! B \in \beta \text{ such that } f(W) \in F_{\mathcal{W}}(\Theta)\}, && (5)\\ \alpha &= \{A \in \lambda^{-1}(a) \mid \exists! B \in \beta, \exists! C \in \gamma \text{ such that } f(W) \in F_{\mathcal{W}}(\Theta)\}. && (6) \end{aligned}$$

The uniqueness of B and C is required by Remark 2. Notice that α_0 is used only at the beginning of rows, α_1 only in the first row, α in all the other positions of a picture.

Finally, F is defined by

$$\begin{aligned} F \;=\; & \left\{ \begin{array}{ccc} & \gamma & \\ \# & \boxed{a} & {}^{D}_{\beta} \\ & \# & \end{array} \;\middle|\; \exists! A \in \lambda^{-1}(a), \exists! B \in \beta, \exists! C \in \gamma \text{ such that } \begin{array}{ccc} & C & D \\ \sharp & A & B \\ & \sharp & \end{array} \in F_{\mathcal{W}}(\Theta) \right\} \\ \cup\; & \left\{ \begin{array}{ccc} & \gamma & \\ {}^{D}_{\beta} & \boxed{a} & \# \\ & \# & \end{array} \;\middle|\; \exists! A \in \lambda^{-1}(a), \exists! B \in \beta, \exists! C \in \gamma \text{ such that } \begin{array}{ccc} D & C & \\ B & A & \sharp \\ & \sharp & \end{array} \in F_{\mathcal{W}}(\Theta) \right\}. \end{aligned}$$

We now prove that the $\mathcal{S}$-DWA $\tau = \langle \Sigma, K', \delta, \mathcal{S}, F \rangle$ is equivalent to the t2b-unambiguous Wang system $\omega = \langle \Sigma, K, \Theta \rangle$.

First we show that $L(\omega) \subseteq L(\tau)$. Let $p \in L(\omega)$ and let $P \in L_{\mathcal{W}}(\Theta)$ be the Wang tiling of p according to ω. W.l.o.g assume that p has a even number of row (the other case is similar), and let $a_{i,j}$ and $A_{i,j}$ be the pixels of p and P respectively. Then one can easily verify that the automaton τ accepts p producing a Wang tiling P' for p as in Figure 5, where $A_{i,j} \in \alpha_{i,j}$ for every $i = 1, 2, \ldots, n-1$ and $j = 2, 3, \ldots, m-1$.

W	d	$f(W)$	δ	η
$\begin{array}{cc} & \# \\ \# & \boxed{a} \end{array}$	$\downarrow$	$\begin{array}{cc} & \sharp \\ \sharp & A \end{array}$	$\begin{array}{ccc} & \# & \\ \# & \boxed{a} & \alpha_0 \\ & \alpha_0 & \end{array}$	$\rightarrow$
$\begin{array}{cc} & \# \\ \beta & \boxed{a} \end{array}$	$\rightarrow$	$\begin{array}{cc} & \sharp \\ B & A \end{array}$	$\begin{array}{ccc} & \# & \\ \beta & \boxed{a} & \alpha_1 \\ & \alpha_1 & \end{array}$	$\rightarrow$
$\begin{array}{ccc} & \# & \\ \beta & \boxed{a} & \# \end{array}$	$\rightarrow$	$\begin{array}{ccc} & \sharp & \\ B & A & \sharp \end{array}$	$\begin{array}{ccc} & \# & \\ \beta & \boxed{a} & \# \\ & \alpha_1 & \end{array}$	$\downarrow$
$\begin{array}{cc} & \{D\} \\ \# & \boxed{a} \\ & \# \end{array}$	$\downarrow$	$\begin{array}{cc} & D \\ \sharp & A \\ & \sharp \end{array}$	$\begin{array}{ccc} & \{D\} & \\ \# & \boxed{a} & \begin{smallmatrix} D \\ \alpha_0 \end{smallmatrix} \\ & \# & \end{array}$	$\rightarrow$
$\begin{array}{cc} & \gamma \\ \begin{smallmatrix} D \\ \beta \end{smallmatrix} & \boxed{a} \\ & \# \end{array}$	$\rightarrow$	$\begin{array}{cc} D & C \\ B & A \\ & \sharp \end{array}$	$\begin{array}{ccc} & \gamma & \\ \begin{smallmatrix} D \\ \beta \end{smallmatrix} & \boxed{a} & \begin{smallmatrix} C \\ \alpha \end{smallmatrix} \\ & \# & \end{array}$	$\rightarrow$
$\begin{array}{cc} \{D\} & \\ \boxed{a} & \# \\ \# & \end{array}$	$\downarrow$	$\begin{array}{cc} D & \\ A & \sharp \\ \sharp & \end{array}$	$\begin{array}{ccc} & \{D\} & \\ \begin{smallmatrix} D \\ \alpha_0 \end{smallmatrix} & \boxed{a} & \# \\ & \# & \end{array}$	$\leftarrow$
$\begin{array}{cc} \gamma & \\ \boxed{a} & \begin{smallmatrix} D \\ \beta \end{smallmatrix} \\ \# & \end{array}$	$\leftarrow$	$\begin{array}{cc} C & D \\ A & B \\ \sharp & \end{array}$	$\begin{array}{ccc} & \gamma & \\ \begin{smallmatrix} C \\ \alpha \end{smallmatrix} & \boxed{a} & \begin{smallmatrix} D \\ \beta \end{smallmatrix} \\ & \# & \end{array}$	$\leftarrow$

W	d	$f(W)$	δ	β
$\begin{array}{cc} & \{D\} \\ \# & \boxed{a} \end{array}$	$\downarrow$	$\begin{array}{cc} & D \\ \sharp & A \end{array}$	$\begin{array}{ccc} & \{D\} & \\ \# & \boxed{a} & \begin{smallmatrix} D \\ \alpha_0 \end{smallmatrix} \\ & \alpha_0 & \end{array}$	$\rightarrow$
$\begin{array}{cc} & \gamma \\ \begin{smallmatrix} D \\ \beta \end{smallmatrix} & \boxed{a} \end{array}$	$\rightarrow$	$\begin{array}{cc} D & C \\ B & A \end{array}$	$\begin{array}{ccc} & \gamma & \\ \begin{smallmatrix} D \\ \beta \end{smallmatrix} & \boxed{a} & \begin{smallmatrix} C \\ \alpha \end{smallmatrix} \\ & \alpha & \end{array}$	$\rightarrow$
$\begin{array}{ccc} & \gamma & \\ \begin{smallmatrix} D \\ \beta \end{smallmatrix} & \boxed{a} & \# \end{array}$	$\rightarrow$	$\begin{array}{ccc} D & C & \\ B & A & \sharp \end{array}$	$\begin{array}{ccc} & \gamma & \\ \begin{smallmatrix} D \\ \beta \end{smallmatrix} & \boxed{a} & \# \\ & \alpha & \end{array}$	$\downarrow$
$\begin{array}{cc} \{D\} & \\ \boxed{a} & \# \end{array}$	$\downarrow$	$\begin{array}{cc} D & \\ A & \sharp \end{array}$	$\begin{array}{ccc} & \{D\} & \\ \begin{smallmatrix} D \\ \alpha_0 \end{smallmatrix} & \boxed{a} & \# \\ & \alpha_0 & \end{array}$	$\leftarrow$
$\begin{array}{cc} \gamma & \\ \boxed{a} & \begin{smallmatrix} D \\ \beta \end{smallmatrix} \end{array}$	$\leftarrow$	$\begin{array}{cc} C & D \\ A & B \end{array}$	$\begin{array}{ccc} & \gamma & \\ \begin{smallmatrix} C \\ \alpha \end{smallmatrix} & \boxed{a} & \begin{smallmatrix} D \\ \beta \end{smallmatrix} \\ & \alpha & \end{array}$	$\leftarrow$
$\begin{array}{ccc} & \gamma & \\ \# & \boxed{a} & \begin{smallmatrix} D \\ \beta \end{smallmatrix} \end{array}$	$\leftarrow$	$\begin{array}{ccc} & C & D \\ \sharp & A & B \end{array}$	$\begin{array}{ccc} & \gamma & \\ \# & \boxed{a} & \begin{smallmatrix} D \\ \beta \end{smallmatrix} \\ & \alpha & \end{array}$	$\downarrow$

Table 3: The (deterministic) transition function δ of the Wang automaton τ that simulates ω: δ and η stand respectively for $\delta(W, d)$ and $\eta(\Delta_W, d)$. The two components of elements of $\Theta \times \Phi$ are shown one above the other for better readability. Tile-sets α_0, α_1, α and tile C are defined in (4- 5- 6).

Vice versa, let $p \in L(\tau)$ with pixels $a_{i,j}$ (again we assume w.l.o.g. that p has a even number of row), and let P' be the Wang tiling of p produced by τ. Then P' is as in Figure 5, for every $i = 1, 2, \ldots, n-1$ and $j = 2, 3, \ldots, m-1$. Now, for every odd i let $A_{i,1}$ be the unique element of $\alpha_{i,1}$ such that $A_{i,1}A_{i,2} \in F_{\mathcal{W}}(\Theta)$; similarly, for every even i let $A_{i,m}$ be the unique element of $\alpha_{i,m}$ such that $A_{i,m-1}A_{i,m} \in F_{\mathcal{W}}(\Theta)$. Moreover, let $A_{n,1}$ and $A_{n,2}$ be the unique tiles such that

$$\begin{array}{ccc} & A_{n-1,1} & A_{n-1,2} \\ \sharp & A_{n,1} & A_{n,2} \\ & \sharp & \end{array} \in F_{\mathcal{W}}(\Theta),$$

and, for every $j = 3, \ldots m$, let $A_{n,j}$ be the unique element of $\alpha_{n,j}$ such that

$$\begin{array}{cc} A_{n-1,j-1} & A_{n-1,j} \\ A_{n,j-1} & A_{n,j} \\ \sharp & \end{array} \in F_{\mathcal{W}}(\Theta).$$

Then it is immediate to verify that the Wang picture P with pixels $A_{i,j}$ is a Wang tiling of p and belongs to $L_{\mathcal{W}}(\Theta)$. Hence, $p \in L(\omega)$ and $L(\tau) \subseteq L(\omega)$.

$$P' = \begin{array}{|ccccccccccccc|}
\hline
 & \# & & \# & & \# & & \# & & \# & & \# & \\
\# & \boxed{a_{1,1}} & \alpha_{1,1} & \boxed{a_{1,2}} & \alpha_{1,2} & \boxed{\cdots} & \cdots & \boxed{\cdots} & \cdots & \boxed{a_{1,m-1}} & \alpha_{1,m-1} & \boxed{a_{1,m}} & \# \\
 & \alpha_{1,1} & & \alpha_{1,2} & & \cdots & & \cdots & & \alpha_{1,m-1} & & \{A_{1,m}\} & \\
\# & \boxed{a_{2,1}} & \begin{smallmatrix} A_{1,2} \\ \alpha_{2,2} \end{smallmatrix} & \boxed{a_{2,2}} & \cdots & \boxed{\cdots} & \cdots & \boxed{\cdots} & \begin{smallmatrix} A_{1,m-1} \\ \alpha_{2,m-1} \end{smallmatrix} & \boxed{a_{2,m-1}} & \begin{smallmatrix} A_{1,m} \\ \alpha_{2,m} \end{smallmatrix} & \boxed{a_{2,m}} & \# \\
 & \{A_{2,1}\} & & \alpha_{2,2} & & \cdots & & \cdots & & \alpha_{2,m-1} & & \alpha_{2,m} & \\
\# & \boxed{a_{3,1}} & \begin{smallmatrix} A_{2,1} \\ \alpha_{3,1} \end{smallmatrix} & \boxed{a_{3,3}} & \begin{smallmatrix} A_{2,2} \\ \alpha_{3,2} \end{smallmatrix} & \boxed{\cdots} & \cdots & \boxed{\cdots} & \cdots & \boxed{a_{3,m-1}} & \begin{smallmatrix} A_{2,m-1} \\ \alpha_{3,m-1} \end{smallmatrix} & \boxed{a_{3,m}} & \# \\
 & \alpha_{3,1} & & \alpha_{3,2} & & \cdots & & \cdots & & \alpha_{3,m-1} & & \{A_{3,m}\} & \\
\# & \boxed{\cdots} & \cdots & \boxed{\cdots} & \cdots & \boxed{\cdots} & \cdots & \boxed{\cdots} & \cdots & \boxed{\cdots} & \cdots & \boxed{\cdots} & \# \\
 & \cdots & & \cdots & & \cdots & & \cdots & & \cdots & & \cdots & \\
\# & \boxed{\cdots} & \cdots & \boxed{\cdots} & \cdots & \boxed{\cdots} & \cdots & \boxed{\cdots} & \cdots & \boxed{\cdots} & \cdots & \boxed{\cdots} & \# \\
 & \alpha_{n-1,1} & & \alpha_{n-1,2} & & \cdots & & \cdots & & \alpha_{n-1,m-1} & & \{A_{n-1,m}\} & \\
\# & \boxed{a_{n,1}} & \begin{smallmatrix} A_{n-1,2} \\ \alpha_{n,2} \end{smallmatrix} & \boxed{a_{n,2}} & \cdots & \boxed{\cdots} & \cdots & \boxed{\cdots} & \begin{smallmatrix} A_{n-1,m-1} \\ \alpha_{n,m-1} \end{smallmatrix} & \boxed{a_{n,m-1}} & \begin{smallmatrix} A_{n-1,m} \\ \alpha_{n,m} \end{smallmatrix} & \boxed{a_{n,m}} & \# \\
 & \# & & \# & & \# & & \# & & \# & & \# & \\
\hline
\end{array}$$

Figure 5: Wang tiling obtained by the computation of τ on picture p.

Therefore, $L(\tau) = L(\omega)$ and this proves that any t2b-UWS can be simulated by a $\mathcal{S}$-DWA, thus concluding the proof of Theorem 5.1.

6 Conclusions

Wang automata are a simple and natural model for defining REC. Their main strength is coupling the power of Wang systems with simple scanning strategies, that permits a straightforward introduction of the notion of determinism. We proved here that such notion defines a relevant deterministic subclass of REC, properly containing the traditional one based on diagonal determinism. These results make WAs a worthy and relevant addition to the literature on 2D operational devices.

Among the open problems we mention the following:

- Is Scan-DREC a proper subset of UREC?
- Relation among $\mathcal{L}(\mathcal{C}$-DWA), $\mathcal{L}(\mathcal{S}$-DWA) and deterministic subclasses determined by the other significant scanning strategies, such as the ones of Figure 3 (b, c).
- Relation between Scan-DREC and deterministic 4-way automata.

As far as future work is concerned, we plan to investigate extensions of the Wang automata model, based on more complex (so "less polite") scanning strategies. For instance, we could drop the blindness requirement, thus allowing a behavior that depends on the actual content of the input picture. Or we could admit automata that are able to pass zero or more than one time over the same position. Such extensions may make the model more complex and powerful and have to be carefully considered and studied, since they could exceed REC.

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] M. Anselmo, D. Giammarresi, and M. Madonia. A computational model for recognizable two-dimensional languages. *Theoretical Computer Science*, 410(37):3520–3529, 2009.

[4] M. Anselmo, D. Giammarresi, and M. Madonia. Deterministic and unambiguous families within recognizable two-dimensional languages. *Fundamenta Informaticae*, 98(2-3):143–166, 2010.

[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] P. Behrooz. *Introduction to Parallel Processing: Algorithms and Architectures*. Kluwer Academic Publishers, Norwell, MA, USA, 1999.

[7] A. Bertoni, M. Goldwurm, and V. Lonati. On the complexity of unary tiling-recognizable picture languages. *Fundamenta Informaticae*, 91(2):231–249, 2009.

[8] B. Borchert and K. Reinhardt. Deterministically and sudoku-deterministically recognizable picture languages. In *Proc. LATA 2007*, 2007.

[9] S. Bozapalidis and A. Grammatikopoulou. Recognizable picture series. *Journal of Automata, Languages and Combinatorics*, 10(2/3):159–183, 2005.

[10] R. Brijder and H. J. Hoogeboom. Perfectly quilted rectangular snake tilings. *Theoretical Computer Science*, 410(16):1486–1494, 2009.

[11] 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.

[12] 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.

[13] L. de Prophetis and S. Varricchio. Recognizability of rectangular pictures by Wang systems. *Journal of Automata, Languages and Combinatorics*, 2(4):269–288, 1997.

[14] 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*.

[15] 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.

[16] K. Inoue and A. Nakamura. Some properties of two-dimensional on-line tessellation acceptors. *Information Sciences*, 13:95–121, 1977.

[17] K. Inoue and I. Takanami. A survey of two-dimensional automata theory. *Information Sciences*, 55(1-3):99–121, 1991.

[18] K. Lindgren, C. Moore, and M. Nordahl. Complexity of two-dimensional patterns. *Journal of Statistical Physics*, 91(5-6):909–951, June 1998.

[19] V. Lonati and M. Pradella. Strategies to scan pictures with automata based on Wang tiles. Submitted. Available at `http://lonati.dsi.unimi.it/lavori/strategies_rapporto.pdf`.

[20] 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.

[21] V. Lonati and M. Pradella. Picture recognizability with automata based on Wang tiles. In *Proc. SOFSEM 2010*, volume 5901 of *Lecture Notes in Computer Science*, pages 576–587. Springer, 2010.

[22] 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.