How to construct convex polyominoes on DNA Wang tiles ?

F. De Carli * A. Frosini * S. Rinaldi * L. Vuillon †

1 Introduction

The goal of this article is to present a method to construct various classes of convex polyominoes using DNA Wang tiles. The construction is based on some results of [4], where the authors present a coding of convex polyominoes in terms of two dimensional languages, by means of tiling systems [12]. Adopting the same formalism we used in [4], and applying an algorithm by L. De Prophetis and S. Varricchio [11], we are able to transform the tiles of the tiling systems for convex polyominoes to labelled Wang tiles. The last step of the construction is based on a conversion from labelled Wang tiles into DNA Wang tiles [1], which gives an effective way to construct nano structures with convex polyomino shapes.

The possibility of constructing polyominoes by means of DNA Wang tiles seems interesting since *polyominoes* are simple and important discrete structures that appear in several problems related to theoretical computer science and discrete mathematics. In the half century, since Solomon Golomb used the term in his seminal article [13], the study of polyominoes has proved a fertile topic of research. By this period in the mid-1950s, it was clearly a timely notion in discrete models, as the increasingly influential work of Neville Temperley, on problems drawn from statistical mechanics and molecular dynamics [16], and of John Hammersely, dealing with percolation [14], bear witness. More recent years have seen the treatment of numerous related problems, such as the problem of covering a polyomino by rectangles [5] or problems of tiling regions by polyominoes [2, 6].

In our work we describe a general method for constructing various shapes of convex polyominoes using DNA Wang tiles. In particular, we can restrict ourselves to some special classes of polyominoes, by imposing some directedness constraints, and then build, for instance, directed-convex polyominoes or parallelogram polyominoes. Thus we are able to send a planar signal in a certain

*Università di Siena, Dipartimento di Scienze Matematiche e Informatiche, Pian dei Mantellini, 44, 53100, Siena, Italy [frosini, rinaldi]@unisi.it.

†Laboratoire de Mathématiques,UMR 5127 CNRS, Université de Savoie, 73376 Le Bourget du Lac, France, Laurent.Vuillon@univ-savoie.fr

direction.
Using this approach polyominoes, can be viewed as a brick for investigating different machines that send a planar signal and compute on the plane using self assembling of DNA oligo-nuceotides. This domain of DNA computing is very active and the transfer of concepts from theoretical computer science to nano structure is a challenge in order to construct biomolecular machines [9, 17]. These works have shown how information and algorithms can be encoded in biochemical systems, while, as E. Winfree points out some efficiency problems deserve interest:

"Recent theoretical work by Adleman, Goel, Reif, and others, has focused on two issues of efficiency: what kinds of shapes and patterns can be assembled using a small number of tiles, and/or how quickly they can be assembled?"

The structure of the article is the following. First, we recall the basic definitions and notations of two-dimensional languages and tiling systems. In Section 3, we recall some basic definitions on polyominoes, in particular the definitions of convex, directed-convex, and parallelogram polyominoes. In Section 4, we describe the algorithm to transform tiles of a tiling system into labelled Wang tiles. In Section 5, we show explicitly the set of labelled Wang tiles that allows us to construct convex polyominoes. In Section 6 we give an example of a parallelogram polyomino built on labelled Wang tiles. The last section concerns the transformation of labelled Wang tiles into DNA Wang tiles. Moreover, we show that it is possible to control the size of the polyominoes to be constructed, by means of a DNA strand.

2 Local languages and tiling systems

In this section we briefly recall the definitions of *local picture language* and *tiling system*, and the main properties which will be useful to comprehend the rest of the paper. For more details on two-dimensional languages we refer to [12].

Given a finite alphabet Σ, we define *picture* of size (m, n) over Σ, a two dimensional rectangular array of elements of Σ having m rows and n columns.
Following the notation introduced in [12], we surround a (m, n) picture p with a special symbol, indicated by $\#$, not contained in Σ, so that we obtain a new picture $\hat{p}$ of size $(m + 2, n + 2)$ (see Fig. 1). This *boundary symbol* results extremely useful in the general framework of two-dimensional languages, when scanning strategies for pictures are requested, while in our contest it will be used to guarantee the rectangular shape of each picture.
We denote by $B_{2,2}(p)$ the set of all blocks (or sub-pictures) of p of size $(2, 2)$. Each element of $B_{2,2}(p)$ is called a *tile*.

Definition 1 Let Σ be a finite alphabet and Σ^{**} the set of all possible pictures over Σ. 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 $L = \{p \in \Sigma^{**} : B_{2,2}(\hat{p}) \subseteq \theta\}$.

$p =$

1	4	4	4	4
1	2	1	1	4
1	1	2	2	2
1	1	3	2	2

$\hat{p} =$

#	#	#	#	#	#	#
#	1	4	4	4	4	#
#	1	2	1	1	4	#
#	1	1	2	2	2	#
#	1	1	3	2	2	#
#	#	#	#	#	#	#

Figure 1: A picture p in the alphabet $\Sigma = \{1, 2, 3, 4\}$, and the picture $\hat{p}$ obtained by surrounding p with the symbol #.

The set θ is usually called a *representation by tiles* for the local language L, and we write $L = L(\theta)$.

Example 1 The language of the pictures over $\Sigma = \{0, 1\}$ of square shape with the symbol 0 in one diagonal and the symbol 1 in all the other positions is a local language (see Fig. 2). The representation by tiles is given by:

$\theta_{D0} =$ { $\begin{matrix}\# & \# \\ \# & 0\end{matrix}$, $\begin{matrix}\# & \# \\ 0 & 1\end{matrix}$, $\begin{matrix}\# & \# \\ 1 & 1\end{matrix}$, $\begin{matrix}\# & \# \\ 1 & \#\end{matrix}$, $\begin{matrix}\# & 0 \\ \# & 1\end{matrix}$, $\begin{matrix}0 & 1 \\ 1 & 0\end{matrix}$, $\begin{matrix}1 & 1 \\ 0 & 1\end{matrix}$, $\begin{matrix}1 & \# \\ 1 & \#\end{matrix}$, $\begin{matrix}\# & 1 \\ \# & 1\end{matrix}$, $\begin{matrix}1 & 0 \\ 1 & 1\end{matrix}$, $\begin{matrix}1 & 1 \\ 1 & 1\end{matrix}$, $\begin{matrix}1 & \# \\ 0 & \#\end{matrix}$, $\begin{matrix}\# & 1 \\ \# & \#\end{matrix}$, $\begin{matrix}1 & 1 \\ \# & \#\end{matrix}$, $\begin{matrix}1 & 0 \\ \# & \#\end{matrix}$, $\begin{matrix}0 & \# \\ \# & \#\end{matrix}$ }.

.

We assume that the empty picture belongs to $L(\theta)$ if and only if θ contains the tile $\begin{matrix}\# & \# \\ \# & \#\end{matrix}$.

Definition 2 A tiling system (TS) is a 4-uple $\mathcal{T} = (\Sigma, \Gamma, \theta, \pi)$, 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.

We say that a tiling system $\mathcal{T}$ defines the language $L = \pi(L(\theta))$, where $L(\theta)$ is a local language over Γ and we write by convention $L = L(\mathcal{T})$. Moreover, we say that $L \subseteq \Sigma^{**}$ is *recognizable by tiling systems* (or *tiling recognizable*) if there exists a tiling system $\mathcal{T} = (\Sigma, \Gamma, \theta, \pi)$, such that $L = L(\mathcal{T})$.

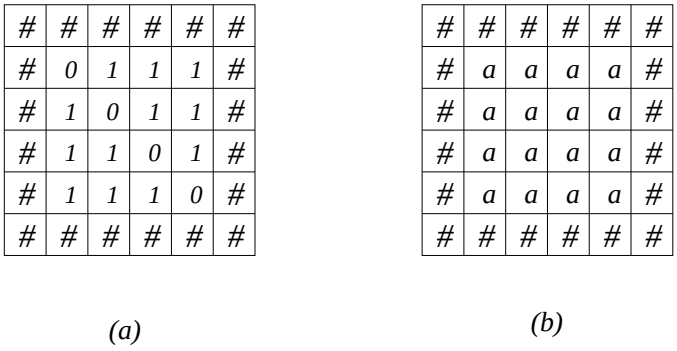


Figure 2: A picture in $L(\theta_{D0})$, (a), and the corresponding picture in $L(\mathcal{T})$, (b).

Example 2 Consider the local language in Example 1. Let $\pi : \Gamma \to \{a\}$ be the projection that maps each element of Γ in a. Let $\Sigma = \{a\}$, then the tiling system $\mathcal{T} = (\Sigma, \Gamma, \theta_{D0}, p)$ recognizes the language $L(\mathcal{T})$ of the pictures over Γ having the form of a square (see Fig. 2). Notice that such a language is not local, whereas it is tiling recognizable.

3 Basics on polyominoes

In the plane $\mathbb{Z} \times \mathbb{Z}$ a *cell* is a unit square, and a *polyomino* is a finite connected union of cells having no cut point. Polyominoes are defined up to translations. A *column* (*row*) of a polyomino is the intersection between the polyomino and an infinite strip of cells whose centers lie on a vertical (horizontal) line.

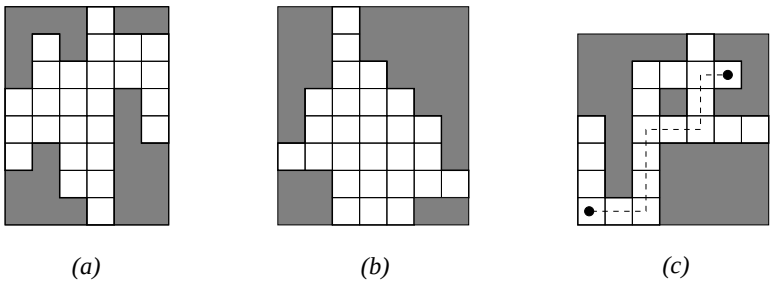


Figure 3: (a) a column-convex polyomino; (b) a convex polyomino; (c) a directed (not convex) polyomino.

A polyomino is *convex* if it is both column and row convex (Fig. 6 (b)). A polyomino P is said to be *directed* when every cell of P can be reached from a distinguished cell (usually the leftmost at the lowest ordinate), by a path

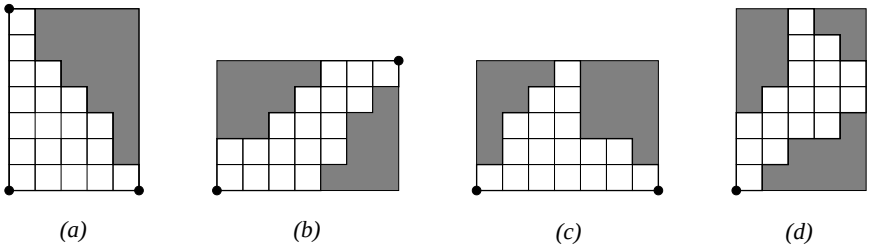


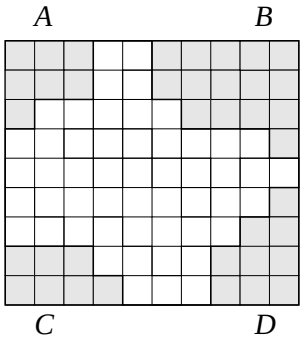
Figure 4: (a) a Ferrers diagram; (b) a parallelogram polyomino; (c) a stack polyomino; (d) a directed-convex polyomino.

which is contained in P and only uses north and east unit steps (Fig. 6 (c)). Figure 4 (d) depicts a polyomino that is both directed and convex. Moreover we can define three types of directed and convex polyominoes, i.e. the *Ferrers diagrams* (Fig. 4 (a)), the *parallelogram polyominoes* (Fig. 4 (b)), and the *stack polyominoes* (Fig. 4 (c)). As Figure 4 shows, each of these three subsets can be characterized, in the set of convex polyominoes, by the fact that two or three vertices of the minimal bounding rectangle of the polyomino must also belong to the polyomino itself.

A generic polyomino P can be can be encoded as a two-dimensional word on the alphabet $\{0, 1\}$ obtained by using a 1 (resp. a 0) to represent the cells of the minimal bounding rectangle belonging to P (resp. not belonging to P). If the language of these two-dimensional words is tiling recognizable, then we say that the class of polyominoes under consideration is *tiling recognizable*.
In [4], we proved that Ferrers diagrams, convex, directed-convex, stack, and parallelogram polyominoes are tiling recognizable classes. The main idea on which relies our construction is to control the convexity of the four disjoint sides A, B, C and D (possibly empty) which form the exterior of a convex polyomino. To each convex polyomino we associate a picture obtained by representing with a 1 every cell belonging to the polyomino, and with the symbol a (resp. b, c, d) every cell in A (resp. B, C, D), as depicted in Fig. 5 (b). We can prove that the language of these pictures is local, hence we can map easily it into the language representing convex polyominoes.

4 Algorithm to transform Tiling Systems into labelled Wang Tiles

In this section we recall some results that link Wang tiles and tiling systems. First of all we recall that a *Wang Tile* is a square in which each edge is assigned a color (or symbol) and a *labelled Wang tile* is a Wang tile in which the interior is assigned a symbol.



#	#	#	#	#	#	#	#	#	#	#	#
#	a	a	a	1	1	b	b	b	b	b	#
#	a	a	a	1	1	b	b	b	b	b	#
#	a	1	1	1	1	1	b	b	b	b	#
#	1	1	1	1	1	1	1	1	1	b	#
#	1	1	1	1	1	1	1	1	1	1	#
#	1	1	1	1	1	1	1	1	1	d	#
#	1	1	1	1	1	1	1	1	d	d	#
#	c	c	c	1	1	1	1	d	d	d	#
#	c	c	c	c	1	1	1	d	d	d	#
#	#	#	#	#	#	#	#	#	#	#	#

(a) *(b)*

Figure 5: (a) a convex polyomino P individuates four disjoint sides; (b) the representation of P as a word on the alphabet $\{a, b, c, d, 1\}$.

We represent a labelled Wang tile by: $\begin{array}{|c|}\hline \alpha \\ \beta \boxed{x} \gamma \\ \delta \\ \hline \end{array}$ where $\alpha, \beta, \gamma, \delta$ are symbols representing colors for the edges, and x is the symbol for the label.
Given a set of Wang tiles, a valid tiling requires all shared edges between tiles to have matching colors. Wang tiles were introduced in [18, 19] and later studied in [7] in relation to problems concerning the tiling of the infinite Euclidean plane. Labelled Wang tiles were used in [11] in matter of recognizability of picture languages. More recently Wang tiles are used for image generation (see [8]) and DNA computing (see [17]).
For the result of this section we refer to the paper [11], which describes the bijective correspondence between tiling systems and labelled Wang tiles. Substantially we will apply the same algorithm, changing just a little the notation.
Let $T = \langle \Sigma, \Gamma, \Theta, \pi \rangle$ be a tiling system, we consider over the set of tiles Θ, five subsets of tiles: Θ_N the tiles of the northern border, Θ_E the tiles of the eastern border, Θ_S the tiles of the southern border, Θ_W the tiles of the western border, Θ_C the corner tiles and Θ_I the set of the remaining tiles (the tiles of the interior).

Before giving the algorithm we need to observe the following facts:

1. If we start from a picture of a tiling system, and we want to represent it using labelled Wand tiles, obviously the dimensions of the picture (i.e. the number of its columns and rows) must be the same in both the representations. While this can be a trivial observation, it will be useful in the following.
2. We must take care of the projection π. This projection maps the alphabet Σ (i.e. the alphabet of $L(\Theta)$) in Γ (i.e. the alphabet of $L(T)$) in such a way that a word $p \in L(\Theta)$ is mapped into $p' \in L(T)$; more precisely the symbol in position (i, j) in p' is the image by π of the symbol in the position (i, j) in p. Thus we insert a label in the labelled Wang tile, at position (i, j), which is the image through π of the symbol in the position (i, j) in p (that is the symbol at position (i, j) in p').

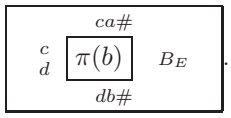
Algorithm to transform tiling Systems into labelled Wang tiles.

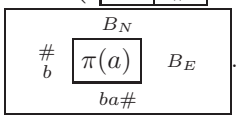
The algorithm performs in the following three steps.

- First we consider Θ_N. Its tiles are of the kind $\begin{array}{|c|c|}\hline \# & \# \\ \hline a & b \\ \hline \end{array}$; for each of them we construct the labelled Wang tile $\begin{array}{|ccc|}\hline & B_N & \\ \begin{smallmatrix}\# \\ a\end{smallmatrix} & \boxed{\pi(b)} & \begin{smallmatrix}\# \\ b\end{smallmatrix} \\ & ab & \\ \hline \end{array}$, where with B_N we mean that we are placed on the northern border, with $\begin{smallmatrix}\# \\ a\end{smallmatrix}$ we mean one symbol, and the same holds for $\begin{smallmatrix}\# \\ b\end{smallmatrix}$ and for ab. Though this notation may be confusing, we use the two symbols (which let us individuate the side of the tile we are considering) to mean one.
- Concerning the set of tiles Θ_W, it contains tiles of the kind $\begin{array}{|c|c|}\hline \# & a \\ \hline \# & b \\ \hline \end{array}$, we replace them with the labelled Wang tile $\begin{array}{|ccc|}\hline & \#a & \\ B_W & \boxed{\pi(b)} & \begin{smallmatrix}a \\ b\end{smallmatrix} \\ & \#b & \\ \hline \end{array}$, where the notation is the same as above.
- For the west-north corner tile $\begin{array}{|c|c|}\hline \# & \# \\ \hline \# & a \\ \hline \end{array} \in \Theta_I$ we have the labelled Wang tile $\begin{array}{|ccc|}\hline & B_N & \\ B_W & \boxed{\pi(a)} & \begin{smallmatrix}\# \\ a\end{smallmatrix} \\ & \#a & \\ \hline \end{array}$.

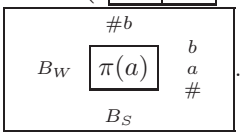
We note that the label of the labelled Wang tiles is given by the projection of the symbol which is placed on the rightmost position of the label of the south edge, and on the lowest position of the label of the east edge of the labelled Wang tile.
For what we say in the previous observations we can not repeat the easy mechanism that we used to translate $\Theta_N, \Theta_W, \Theta_I$ also to translate the remaining subsets of tiles. In fact, following this procedure, we would construct a picture with one row and one column more than the corresponding two-dimensional picture of the tiling system. To get rid of this problem we use a trick to contract the two rows more at south into a unique row and the two columns more at east into one column.

- For the subset Θ_S we translate a pair of tiles of the tiling system in one labelled Wang tile. More precisely, for each tile of Θ_S we translate the pair $\left(\begin{array}{|c|c|}\hline a & b \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|}\hline c & d \\ \hline a & b \\ \hline \end{array} \right)$ with the labelled Wang tile $\begin{array}{|ccc|}\hline & cd & \\ \begin{smallmatrix}c \\ a \\ \#\end{smallmatrix} & \boxed{\pi(b)} & \begin{smallmatrix}d \\ b \\ \#\end{smallmatrix} \\ & B_S & \\ \hline \end{array}$. We observe that the first component of the pair is a tile of Θ_S, and the second one a tile that matches with the row on its north. This allows us to contract the two southern rows.
- For the tiles of the eastern border we repeat an analogous of the previous reasoning in order to contract the two eastern columns: the generic pair of tiles $\left(\begin{array}{|c|c|}\hline a & \# \\ \hline b & \# \\ \hline \end{array}, \begin{array}{|c|c|}\hline c & a \\ \hline d & b \\ \hline \end{array} \right)$ is translated with the labelled Wang tile

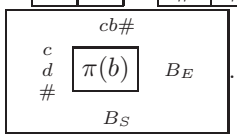


- Concerning the N-E corner tile, the pair $\left(\begin{array}{|c|c|}\hline \# & \# \\ \hline a & \# \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & \# \\ \hline b & a \\ \hline \end{array} \right)$ is translated into the labelled Wang tile 

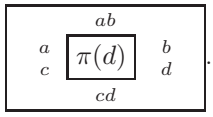
.

- For the S-W corner tile, the pair $\left(\begin{array}{|c|c|}\hline \# & a \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|}\hline \# & b \\ \hline \# & a \\ \hline \end{array} \right)$ is translated into the labelled Wang tile 

.

- For the E-S corner tile, the tiles $\begin{array}{|c|c|}\hline a & \# \\ \hline \# & \# \\ \hline \end{array}$, $\begin{array}{|c|c|}\hline c & b \\ \hline d & a \\ \hline \end{array}$, $\begin{array}{|c|c|}\hline d & a \\ \hline \# & \# \\ \hline \end{array}$, $\begin{array}{|c|c|}\hline b & \# \\ \hline a & \# \\ \hline \end{array}$ are translated into the labelled Wang tile 

.

- The set Θ_I contains tiles of the kind $\begin{array}{|c|c|}\hline a & b \\ \hline c & d \\ \hline \end{array}$, that we translate with the labelled Wang tile 

.

Performing the translation we have obtained all the labelled Wang tiles necessary to represent the local language $L(\Theta)$ recognized by the tiling system T. Finally, we must take care of the projection π. We recall that π maps the alphabet Σ, that is the alphabet of $L(\Theta)$ and of the label of the labelled Wang tiles, in Γ. Then we must replace the labels of the labelled Wang tiles with the respective images through π, and we have completed the translation.

The reader can find the proof of the correctedness and validity of the algorithm in [11]. In the next section we present a slight improvement to the previously defined procedure, by adding a class W_λ, in order to control the transformation of particular objects, the empty polyomino and of polyominoes of sizes $1 \times n$ and $m \times 1$ in labelled Wang tiles.

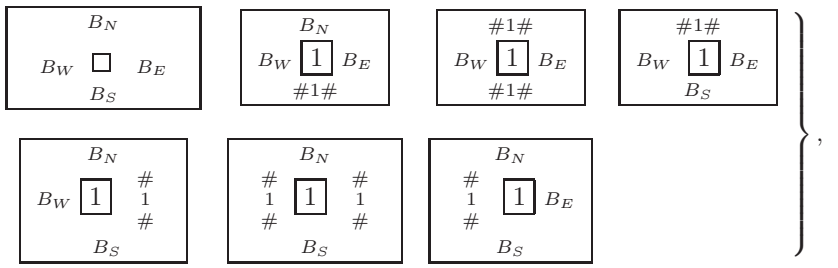
5 Convex polyominoes constructed on labelled Wang tiles

In the paper [4] it was proved that many classes of convex polyominoes can be encoded as words of tiling recognizable two-dimensional languages. In particular it was showed the coding for the class of convex polyominoes. We refer to [3, 4] for formal definition

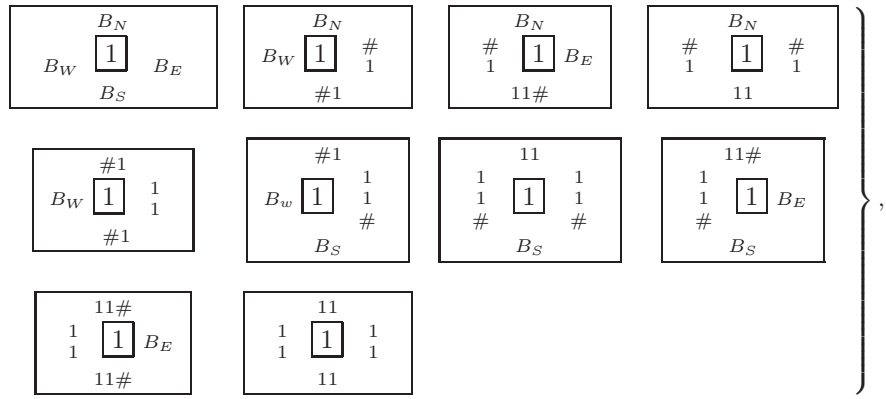
of the tiling system $T_c = \langle \Sigma_c, \{0,1\}, \Theta_c, \pi \rangle$ which recognizes the two dimensional language $\mathcal{L}_c$ of the convex polyominoes. We recall that $\Sigma_c = \{a, b, c, d\}$ and the projection π for this language is $\pi(a) = \pi(b) = \pi(c) = \pi(d) = 0$, $\pi(1) = 1$ (see Fig. 5).

We are now able to encode T_c using the algorithm presented in the previous section, and then give the set of labelled Wang tiles that represents the language $\mathcal{L}_c$.

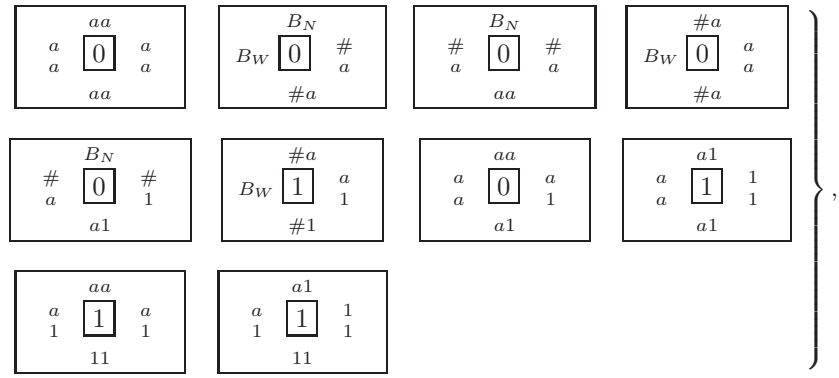
$W_\lambda =$

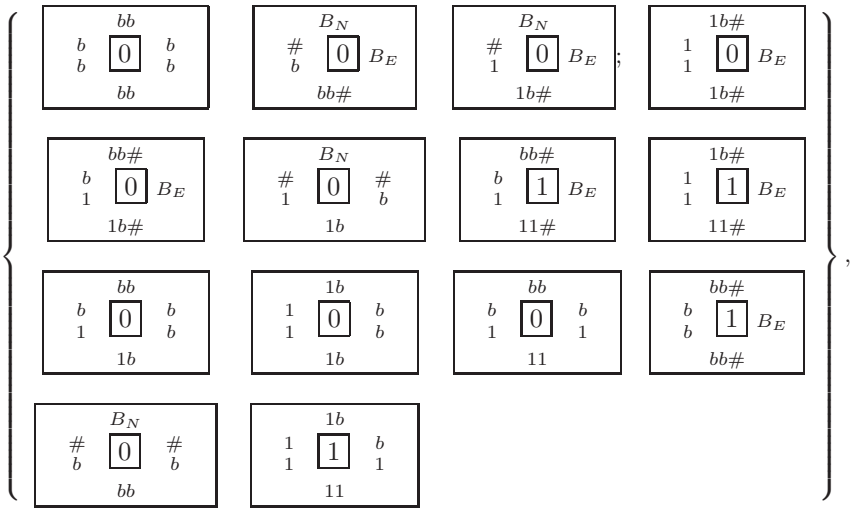


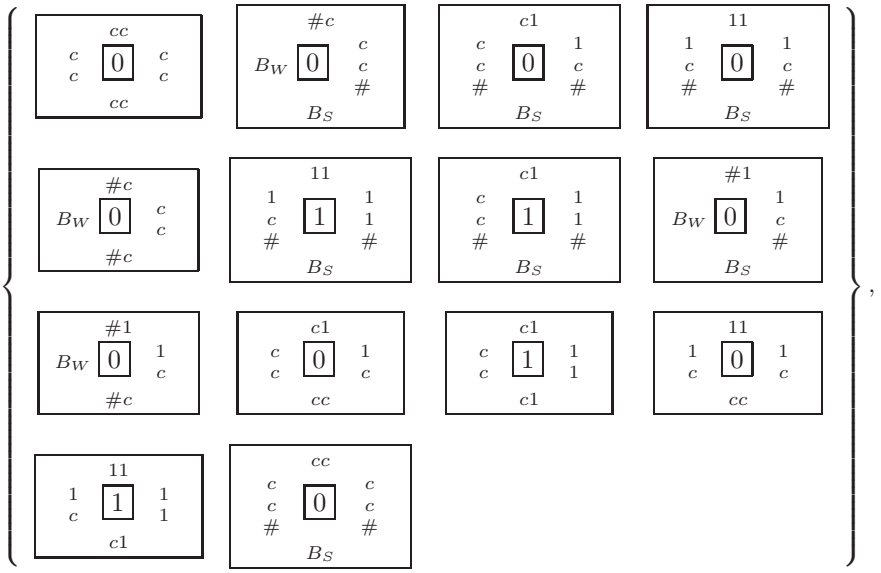
$W_R =$

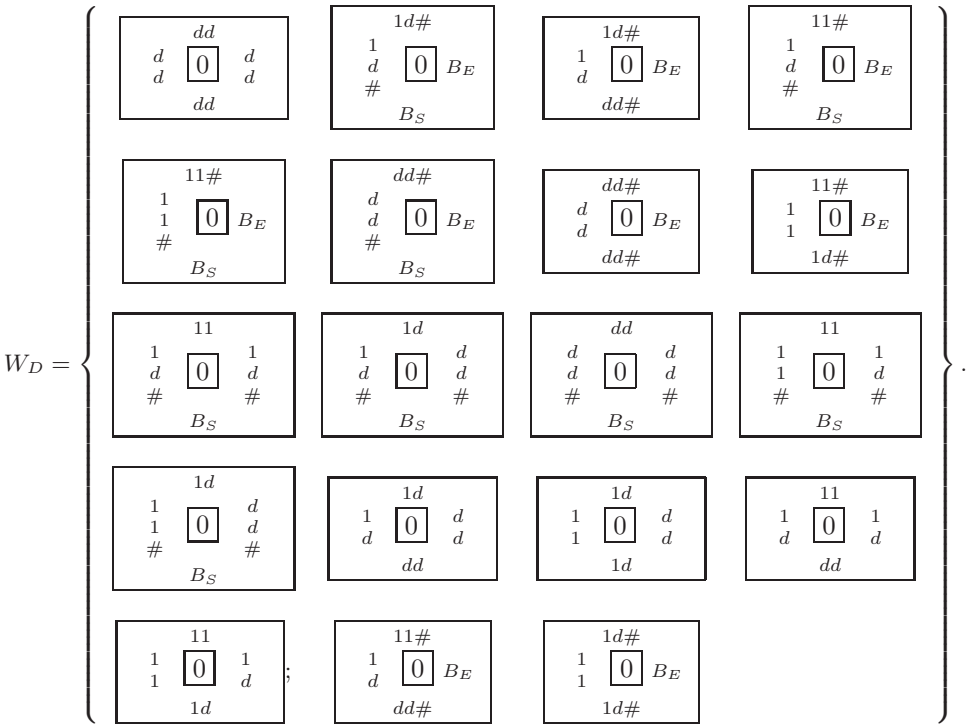


$W_A =$



$W_B =$ 

$W_C =$ 



To summarize, convex polyominoes are generated on the plane using the set of labelled Wang tiles $W_{Conv} = W_\lambda \cup W_R \cup W_A \cup W_B \cup W_C \cup W_D$ where:

i. W_λ generates rectangles of dimensions 0, $1 \times n$ and $m \times 1$;

ii. W_R generates the other rectangles;

iii. W_A controls the upper right side of the exterior of the polyomino (side A in the picture Fig. 5), W_B the upper left side, W_C the lower right side and W_D the lower left side.

6 Construction of a Parallelogram polyomino using labelled Wang tiles

An interesting refinement of the algorithm we have proposed is the encoding of parallelogram polyominoes using labelled Wang tiles. The utility of such encoding is for instance to simulate a planar signal. In particular, such constructions appear in the theory of cellular automata (see [15]) and also for the construction of planar signal machines in order to investigate planar computation using geometry (see [10]). The goal is to send many planar signals and to synchronize

the signals in order to make one or many actions. The best known example in discrete mathematics is of course the "firing squad synchronization problem" and the discrete and continuous variations of this problem ([10, 15]).

We recall that the class of parallelogram polyominoes can be defined using the notions of convexity and and of directed polyomino. In fact, parallelogram polyominoes can be characterized –within the class of convex polyominoes– by the property that two vertices (precisely the south-west and the north-east vertices) of the minimal bounding rectangle of the polyomino must also belong to the polyomino itself (see Fig.4 (b)).
As for the class of convex polyominoes, it is possible to encode the class of the parallelogram polyominoes by means of a two-dimensional languages [4].
So, let us indicate with $\mathcal{L}_P$ the recognizable two-dimensional language that represents parallelogram polyominoes. Then –applying again our algorithm– we can translate $\mathcal{L}_P$ into a set W_P of labelled Wang tiles. More explicitly this set of Wang tiles is given by $W_P = W_\lambda \cup W_R \cup W_A \cup W_C$, where W_λ, W_R, W_A, and W_C have been defined in the previous section.

Just to give an example, we show a parallelogram polyomino, the two-dimensional word coming from $\mathcal{L}_P$, its projection by π (where $\pi(a) = \pi(c) = 0, \pi(1) = 1$) and its encoding by means of labelled Wang tiles.

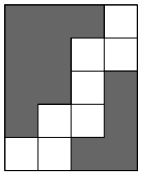


Figure 6: A directed (parallelogram) polyomino

#	#	#	#	#	#
#	a	a	a	1	#
#	a	a	1	1	#
#	a	a	1	c	#
#	a	1	1	c	#
#	1	1	c	c	#
#	#	#	#	#	#

$\overrightarrow{\pi}$

#	#	#	#	#	#
#	0	0	0	1	#
#	0	0	1	1	#
#	0	0	1	0	#
#	0	1	1	0	#
#	1	1	0	0	#
#	#	#	#	#	#

Figure 7: The two dimensional word that represents the polyomino in Figure 6 (on the left) and its projection by π (on the right).

Similarly, using labelled Wang tiles, we can construct the following families of convex polyominoes:

- directed convex polyominoes, using the set $W_{DC} = W_\lambda \cup W_R \cup W_A \cup W_B \cup W_C$;

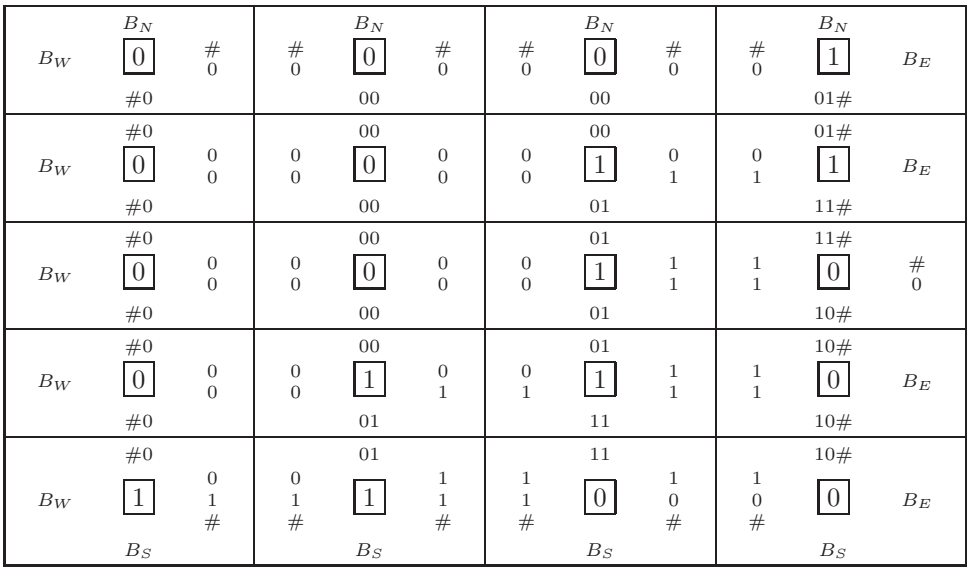


Figure 8: The encoding of the polyomino in Figure 6 by means of labelled Wang tiles.

- Ferrers diagrams, using the set $W_F = W_\lambda \cup W_R \cup W_B$;
- stack polyominoes the set $W_S = W_\lambda \cup W_R \cup W_A \cup W_B$.

7 From Labelled Wang tiles to DNA Wang tiles

The construction of Barish, Rothemund and Winfree [1] suggests a method to transform the set of labelled Wang tiles into the set of DNA Wang tiles. As a matter of fact, we are able to construct a nano structure carrying a bit (0 or 1) according to the exterior of the polyomino (labelled by 0) or the interior of the polyomino (labelled by 1). Following such a construction, we could construct the set of DNA Wang tiles associated with convex polyominoes, parallelogram polyominoes, directed-convex polyominoes, stack polyominoes or Ferres diagrams.

The next step consists in the real construction of the nano structures in solution. In particular, it will be interesting to study the shape of convex polyominoes constructed using DNA tiles, to investigate which shapes are constructed more frequently, and how these properties may change by varying important factors such as the temperature or the concentration in solution of different tiles.
We plan to carry on such a line of research in some future works realized in collaboration with the physicists in Grenoble, with the main purpose to construct polyominoes in nano-structure and to study the typical shapes of DNA polyominoes.

To end this study, an interesting further problem is to construct a strand in order

to control the perimeter of the polyomino generated by DNA Wang tiles. This seems interesting, since the perimeter is one of the most important parameters on which polyominoes have been studied in literature.
We observe that, in each step of the construction, the picture is surrounded by the symbol #. In the last set of tiles, this symbol appears in the Wang tiles of the border. Nevertheless, we also use the symbols B_N, B_E, B_S and B_W for coding the border when we pass to DNA tiles.
Actually, it is sufficient to use 7 symbols (the 4 symbols B_N, B_E, B_S, B_W, plus 3 others) to construct a DNA strand to impose the size of the polyomino. The goal of this last construction is to force the size of the polyomino. This DNA strand begins with B_S, then in the corner we have B_{WS}, then m times B_W, then B_{WN}, then n times B_N, then B_{NE} and at the end B_E. This strand imposes that the constructed polyomino has perimeter equal to $2m + 2n$.

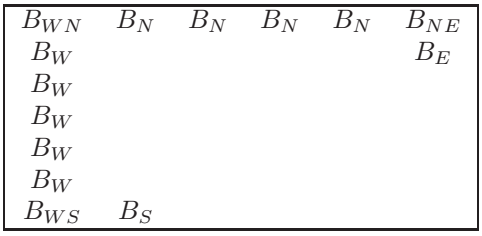


Figure 9: The strand that controls the perimeter of the convex polyomino.

Of course, such a construction gives only by a theoretical point of view a convex polyomino with given perimeter, since in reality there can be errors in the self-assembling. In some possible future work it would be also interesting to study in solution the average number of errors concerning the perimeter of the polyomino generated using DNA tiles.

References

[1] R. Barish, P. Rothemund and E. Winfree, Two computational primitives for algorithmic self-assembly: copying and counting, *Nano letters*, Vol. 5, 2586 - 2592 (2005).

[2] D. Beauquier, M. Nivat, On translating one polyomino to tile the plane, *Disc. Comput. Geom.* 6 (1991) 575–592.

[3] F. De Carli, A. Frosini, S. Rinaldi, A. Sorbi Some remarks on tiling recognizable languages, *Pure Mathematics and Applications*, Vol. 16, 69-80 (2005).

[4] F. De Carli, A. Frosini, S. Rinaldi, L. Vuillon On the Tiling System Recognizability of Various Classes of Convex Polyominoes, *Annals of Combinatorics*, to appear.

[5] S. Chaiken, D. J. Kleitman, M. Saks and J. Shearer, Covering regions by rectangles, *SIAM J. Discr. and Alg. Meth.* 2 (1981) 394–410.

[6] J. H. Conway, J. C. Lagarias, Tiling with polyominoes and combinatorial group theory, *J. Comb. Th. A*, 53 (1990) 183–208.

[7] K. Culik II, An aperiodic set of 13 wang tiles, *Discrete Mathematics* 160 (1996) 245–251.

[8] M.F. Choen, J. Shade, S. Hiller, O. Deussen, Wang Tiles for image and texture generation, ACM Transaction on Graphics (2003).

[9] M.Daley, L.Kari. DNA computing: Models and implementations, *Comments on Theoretical Biology*, vol.7, No.3, 2002, 177-198.

[10] J. Durand-Lose, Calculer géométriquement sur le plan, machines à signaux, HDR, University of Nice, (2003).

[11] L. De Prophetis, S. Varricchio, Recognizability of rectangular pictures by Wang systems, *Journal of Automata, Languages and Combinatorics*, Vol. 2, 269 - 288 (1998).

[12] D. Giammarresi, A. Restivo, Two-dimensional languages, *Handbook of Formal Languages, vol.3*, Springer-Verlag Berlin, A. Salomaa, G. Rozenberg Eds., (1997) 215–267.

[13] S. W. Golomb, Checker boards and polyominoes, *Amer. Math. Monthly*, vol. 61, n. 10 (1954) 675–682.

[14] J.M. Hammersley, Percolation processes II: the connective constant, *Proc. Cambridge. Philos. Soc.*, 53 (1957) 642–645.

[15] J. Mazoyer , On optimal solutions to the Firing squad synchronization problem , *Theoret. Comput. Sci.*, 168 (1996), no. 2, p. 367–404.

[16] H. N. V. Temperley, Combinatorial problems suggested by the statistical mechanics of domains and of rubber-like molecules, *Phys. review* 2 103 (1956) 1–16.

[17] E. Winfree, Algorithmic Self-Assembly of DNA: Theoretical Motivations and 2D Assembly Experiments, *Journal of Biomolecular Structure & Dynamics*, ISSN 0739-1102 Conversation 11, Issue #2 (2000) 0-940030-81-0 Proceedings of the Eleventh Conversation, University at Albany, SUNY, June 15-19, 1999.

[18] H. Wang, Proving theorems by pattern recognition II, *Bell SystemsTechnical Journal*, 40 (1961) 1–42.

[19] H. Wang, Games, logic, and computers, *Scientific American* November (1965), 98–106.