# Interactions between Digital Geometry and Combinatorics on Words

Srečko Brlek

LaCIM, Université du Québec à Montréal,
C. P. 8888 Succursale "Centre-Ville", Montréal (QC), CANADA H3C 3P8

brlek.srecko@uqam.ca

We review some recent results in digital geometry obtained by using a combinatorics on words approach to discrete geometry. Motivated on the one hand by the well-known theory of Sturmian words which model conveniently discrete lines in the plane, and on the other hand by the development of digital geometry, this study reveals strong links between the two fields. Discrete figures are identified with polyominoes encoded by words. The combinatorial tools lead to elegant descriptions of geometrical features and efficient algorithms. Among these, radix-trees are useful for efficiently detecting path intersection, Lyndon and Christoffel words appear as the main tools for describing digital convexity; equations on words allow to better understand tilings by translations.

## 1   Introduction

The expansion of computers has led in the last few decades to several breakthrough in technological achievements. Among these, digital imaging is increasing its spread and is extensively used in a wide range of applications such as image synthesis, remote sensing, medical image processing to cite a few. Developed mostly by the engineering world, digital geometry has led to the discovery (sometimes a rediscovery) of new results about discrete sets, concurrently to the design of new algorithms tools, and enriched the broader field of discrete geometry.

Combinatorics on words has imposed itself as a powerful tool for the study of discrete, linear, and non-commutative objects that appear in almost any branches of mathematics, and discrete geometry is not an exception. Traditionally, digital geometry works on characterization and recognition of discrete objects using an arithmetic approach or computational geometry. However combinatorics on words provide some useful tools and efficient algorithms for handling discrete objects. Lothaire's books [20, 21, 22] constitute the reference for presenting a unified view on combinatorics on words and many of its applications.

As mentioned by Klette and Rozenfeld in their survey on digital straightness [18]

> "*Related work even earlier on the theory of words, specifically, on mechanical or Sturmian words, remained unnoticed in the pattern recognition community*"

there was a need for new investigations where combinatorics on words would enrich the classical Euclidean approach of digital geometry.

We revisit some classical problems in discrete geometry from this new point of view. For our purpose the discrete plane is identified with the square grid $\mathbb{Z} \times \mathbb{Z}$.

## 2  Preliminaries

We refer to Lothaire [20] for the basic terminology and notation about words on a finite alphabet $\mathscr{A}$. It includes the *empty* word $\varepsilon$, *length, , conjugate, factor, prefix, suffix, proper factor, free monoid* $\mathscr{A}^*$, *morphism, antimorphism, occurrences, palindrome, period, power, primitive, reversal*. The set of all factors of $w$ is denoted by $\mathrm{Fact}(w)$, those of length $n$ is $\mathrm{Fact}_n(w) = \mathrm{Fact}(w) \cap \mathscr{A}^n$, $\mathrm{Pref}(w)$ is the set of all prefixes of $w$, and the set of its palindromic factors is $\mathrm{Pal}(L)$. If $w = pu$, with $|w| = n$ and $|p| = k$, then $p^{-1}w = w[k..n-1] = u$ is the word obtained by erasing from $w$ its prefix $p$. The class of a word $w$ is denoted $[w]$. Every word contains palindromes, the letters and $\varepsilon$ being necessarily part of them. This justifies the introduction of the function $\mathrm{LPS} : \mathscr{A}^* \to \mathscr{A}^*$ which associates to any word $w$ its longest palindromic suffix $\mathrm{LPS}(w)$. Given a total order $<$ on $\mathscr{A}$, the *lexicographic ordering* is defined as usual.

**Lyndon words**   Introduced as *standard lexicographic sequences* by Lyndon in 1954, Lyndon words have several characterizations (see [20]). We shall define them as words being strictly smaller than any of their circular permutations.

**Definition 1.** *A Lyndon word $l \in \mathscr{A}^+$ is a word such that $l = uv$ with $u, v \in \mathscr{A}^+$ implies that $l < vu$.*

Note that Lyndon words are always primitive. The most important result about Lyndon words is the following unique factorization theorem (see Lothaire [20] Theorem 5.1.1).

**Theorem 1.** *Any word $w \in \mathscr{A}^+$ admits a unique factorization as a sequence of decreasing Lyndon words:*

$$w = l_1^{n_1} l_2^{n_2} \cdots l_k^{n_k}, \quad l_1 > l_2 > \cdots > l_k \tag{1}$$

*where $n_i \geq 1$ and $l_i$ is a Lyndon word, for all $i$ such that $1 \leq i \leq k$.*

There exist several algorithms for factorizing a word $w = w_1 w_2 \cdots w_n$ into Lyndon words and the more efficient are linear . An elegant one was invented by Duval [16]. It works by reading from left to right, with at most $2n$ comparisons of letters (see also [31], Section 7.4). Another one, uses the concept of *suffix standardization* of the word $w$, and builds a *suffix array* of $w$, which may be computed in linear time [14]. Then the Lyndon factorization of $w$ is obtained by cutting $w$ just before each left-to-right minimum of its suffix-array.

**A quadtree with a radix tree structure for points in the integer plane** ([9, 10])   Let $\mathbb{B} = \{0, 1\}$ be the base for writing integers. Words in $\mathbb{B}^*$ are conveniently represented in the *radix order* by a complete binary tree (see for instance [19, 22]), where the level $k$ contains all the binary words of length $k$, and the order is given by the breadth-first traversal of the tree. To distinguish a natural number $x \in \mathbb{N}$ from its representation we write $\boldsymbol{x} \in \mathbb{B}^*$. The edges are defined inductively by the rewriting rule $\boldsymbol{x} \to \boldsymbol{x} \cdot 0 + \boldsymbol{x} \cdot 1$, with the convention that 0 and 1 are the labels of, respectively, the left and right edges of the node having value $\boldsymbol{x}$. This representation is extended to $\mathbb{B}^* \times \mathbb{B}^*$ as follows. As usual, the concatenation is extended to the cartesian product of words by setting for $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{B}^* \times \mathbb{B}^*$, and $(\alpha, \beta) \in \mathbb{B} \times \mathbb{B}$

$$(\boldsymbol{x}, \boldsymbol{y}) \cdot (\alpha, \beta) = (\boldsymbol{x} \cdot \alpha, \boldsymbol{y} \cdot \beta).$$

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two binary words having same length. Then the rule

$$(\boldsymbol{x}, \boldsymbol{y}) \to (\boldsymbol{x} \cdot 0, \boldsymbol{y} \cdot 0) + (\boldsymbol{x} \cdot 0, \boldsymbol{y} \cdot 1) + (\boldsymbol{x} \cdot 1, \boldsymbol{y} \cdot 0) + (\boldsymbol{x} \cdot 1, \boldsymbol{y} \cdot 1) \tag{2}$$

defines a $\mathscr{G}' = (N, R)$, sub-graph of $\mathscr{G} = (N, R, T)$, such that :

(i)  the root is labeled $(0,0)$;

(ii)  each node (except the root) has four sons;

(iii)  if a node is labeled $(\boldsymbol{x},\boldsymbol{y})$ then $|\boldsymbol{x}| = |\boldsymbol{y}|$;

(iv)  edges are undirected, e.g. may be followed in both directions.

By convention, edges leading to the sons have labels from the ordered set $\{(0,0),(0,1),(1,0),(1,1)\}$. These labels equip the quadtree with a *radix tree* structure for Equation (2) implies that $(x',y')$ is a son of $(x,y)$, if and only if

$$(x',y') = (2x+\alpha, 2y+\beta),$$

for some $(\alpha,\beta) \in \mathbb{B}\times\mathbb{B}$. Observe that any pair $(x,y)$ of nonnegative integers is represented exactly once in this tree. Indeed, if $|\boldsymbol{x}| = |\boldsymbol{y}|$ (by filling with zeros at the left of the shortest one), the sequence of pairs of digits (the two digits in first place, the two digits in second place, and so on) gives the unique path in the tree leading to this pair. Of course the root may have up to three sons since no edge labeled $(0,0)$ starts from the root.
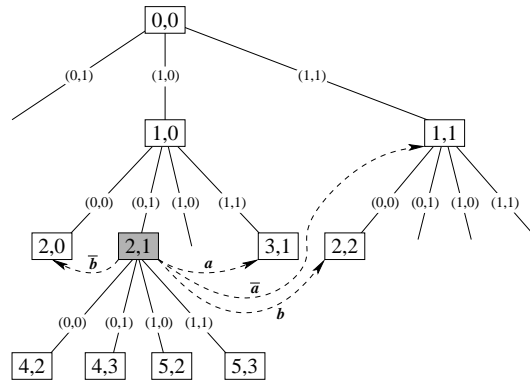


Figure 1: The point $(2,1)$ with its neighbors.

**Neighboring links** [9, 10]   Given $(x,y) \in \mathbb{Z}^2$, a point $(x',y')$ is an $\varepsilon$-*neighbor* of $(x,y)$ if there exists $\varepsilon \in \mathscr{F}$ such that $(x',y') = (x,y) + \varepsilon = (x+\varepsilon_1, y+\varepsilon_2)$.

We superpose on $G'$ the neighboring relation given by the edges of $T$ (dashed lines). More precisely, for each elementary translation $\varepsilon \in \mathscr{F}$, each node $\textcircled{z} = (x,y)$ is linked to its $\varepsilon$-neighbor $\textcircled{z} + \varepsilon$, when it exists. If a level $k$ is fixed, it is easy to construct the graph

$$\mathscr{G}^{(k)} = (N^{(k)}, R^{(k)}, T^{(k)})$$

such that

(i)  if $(\boldsymbol{x},\boldsymbol{y}) \in N^{(k)}$, then $|\boldsymbol{x}| = |\boldsymbol{y}| = k$;

(ii)  the functions $N^{(k)} \hookrightarrow \mathbb{N}\times\mathbb{N} \hookrightarrow \mathbb{B}^*\times\mathbb{B}^*$ are injective;

(iii)  $R^{(k)}$ is the radix-tree representation : $(\mathbb{B}^{<k}\times\mathbb{B}^{<k})\times(\mathbb{B}\times\mathbb{B}) \overset{\bullet}{\to} \mathbb{B}^{\leq k}\times\mathbb{B}^{\leq k}$;

(iv)  the neighboring relation is $T^{(k)} \subseteq N\times(\mathbb{B}\times\mathbb{B})\times N$.

Note that the labeling in Fig. 1 is superfluous: each node represents indeed an integer unambiguously determined by the path from the root using edges in $R$; similarly for the ordered edges. Moreover, if a given subset $M \subset \mathbb{N} \times \mathbb{N}$ has to be represented, then one may trim the unnecessary nodes so that the corresponding graph $\mathscr{G}_M$ is not necessarily complete.

Recall that adding 1 to an integer $x \in \mathbb{B}^k$ is easily performed by a sequential function. Indeed, every positive integer can be written $x = u1^i0^j$, where $i \geq 1$, $j \geq 0$, with $u \in \{\varepsilon\} \cup \{\mathbb{B}^{k-i-j-1} \cdot 0\}$. In other words, $1^j$ is the last run of 1's. The piece of code for adding 1 to an integer written in base 2 is

```
1 : If j ≠ 0 then Return u1^i0^{j-1}1;
2 :          else If u = ε then Return 1·0^i;
3 :                        else Return u·0^{-1}·1·0^i;
4 :          end if
5 : end if
```

where $0^{-1}$ means to erase a 0. Clearly, the computation time of this algorithm is proportional to the length of the last run of 1's. Much better is achieved with the radix tree structure, where, given a node $\textcircled{z}$, its *father* is denoted $f(\textcircled{z})$, and we write $f(x,y)$ or $f(\boldsymbol{x},\boldsymbol{y})$ if its label is $(x,y)$. The following technical lemma is a direct adaptation to $\mathbb{B}^* \times \mathbb{B}^*$ of the addition above.

**Lemma 2.** *Let $G^{(k)}$ be the complete graph representing $\mathbb{B}^{\leq k} \times \mathbb{B}^{\leq k}$ for some $k \geq 1$, $\varepsilon \in \mathscr{F}$, and $\textcircled{z} = (\boldsymbol{x}, \boldsymbol{y})$ be a node of $N^k$. If one of the four conditions holds:*

$$\begin{array}{llll} \text{(i)} & \varepsilon = \boldsymbol{0} \quad \text{and} \quad \boldsymbol{x}[k] = 0, & \text{(ii)} & \varepsilon = \boldsymbol{2} \quad \text{and} \quad \boldsymbol{x}[k] = 1, \\ \text{(iii)} & \varepsilon = \boldsymbol{1} \quad \text{and} \quad \boldsymbol{y}[k] = 0, & \text{(iv)} & \varepsilon = \boldsymbol{3} \quad \text{and} \quad \boldsymbol{y}[k] = 1, \end{array}$$
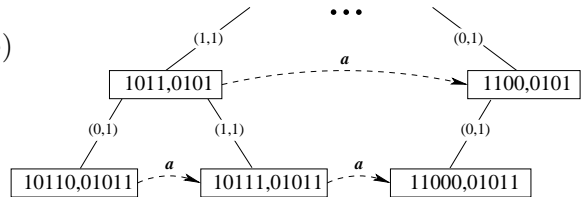
*then $f(\textcircled{z}) = f(\textcircled{z} + \varepsilon)$. Otherwise, $f(\textcircled{z}) + \varepsilon = f(\textcircled{z} + \varepsilon)$.*

The process is illustrated for case (i) in the diagram on the right where the nodes $(10110, \bullet)$ and $(10111, \bullet)$

share the same father while fathers of neighboring nodes

$(\bullet, 01011)$ and $(\bullet, 01011)$

are distinct but share the same neighboring relation.



**A representation for paths in the square grid**  Here, we encode paths with the so-called *Freeman chain code*[23] based on the alphabet $\mathscr{F} = \{\boldsymbol{0}, \boldsymbol{1}, \boldsymbol{2}, \boldsymbol{3}\}$, considered as the additive group of integers mod 4. Basic transformations on $\mathscr{F}$ are rotations $\rho^i : x \mapsto x + i$ and reflections $\sigma_i : x \mapsto i - x$, which extend uniquely to morphisms (w.r.t concatenation) on $\mathscr{F}^*$. Given a nonempty word $w \in \mathscr{F}^*$, the *first differences word* $\Delta(w) \in \mathscr{F}^*$ of $w$ is

$$\Delta(w) = (w_2 - w_1) \cdot (w_3 - w_2) \cdots (w_n - w_{n-1}). \tag{3}$$

One may verify that if $z \in \mathscr{F}^*$, then $\Delta(wz) = \Delta(w)\Delta(w_nz_1)\Delta(z)$. Words in $\mathscr{F}^*$ are interpreted as paths in the square grid, so that we indistinctly talk of any word $w \in \mathscr{F}^*$ as the *path w*. Moreover, the word $\widehat{w} := \rho^2(\widetilde{w})$ is *homologous* to $w$, i.e., in direction opposite to that of $w$ (Figure 2). A word $u \in \mathscr{F}^*$ may contain factors in $\mathscr{C} = \{\boldsymbol{02}, \boldsymbol{20}, \boldsymbol{13}, \boldsymbol{31}\}$, corresponding to cancelling steps on a path. Nevertheless, each word $w$ can be reduced in a unique way to a word $w'$, by sequentially applying the rewriting rules in
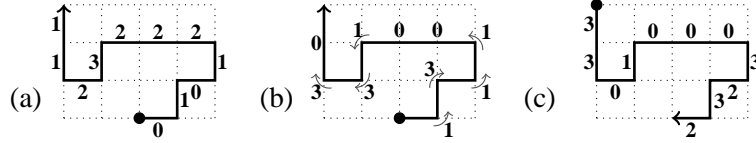
Figure 2: (a) $w = \mathbf{01012223211}$. (b) $\Delta(w) = \mathbf{1311001330}$. (c) $\widehat{w} = \mathbf{33010003232}$.

$\{u \mapsto \varepsilon \mid u \in \mathscr{C}\}$. The *reduced word* $w'$ of $w$ is nothing but a word in $\mathscr{P} = \mathscr{F}^* \setminus \mathscr{F}^* \mathscr{C} \mathscr{F}^*$. The *turning number*[1] of $w$ is defined by $\mathscr{T}(w) = (|\Delta(w')|_\mathbf{1} - |\Delta(w')|_\mathbf{3})/4$.

A path $w$ is *closed* if it satisfies $|w|_\mathbf{0} = |w|_\mathbf{2}$ and $|w|_\mathbf{1} = |w|_\mathbf{3}$, and it is *simple* if no proper factor of $w$ is closed. A *boundary word* is a simple and closed path, and a *polyomino* is a subset of $\mathbb{Z}^2$ contained in some boundary word. It is convenient to represent each closed path $w$ by its conjugacy class $[w]$, also called *circular word*. An adjustment is necessary to the function $\mathscr{T}$, for we take into account the closing turn. The first differences also noted $\Delta$ is defined on any closed path $w$ by setting

$$\Delta([w]) \equiv \Delta(w) \cdot (w_1 - w_n),$$

which is also a closed word. By applying the same rewriting rules, a circular word $[w]$ is *circularly-reduced* to a unique word $[w']$. If $w$ is a closed path, then the *turning number*[1] of $w$ is

$$\mathscr{T}(w) = \mathscr{T}([w]) = \left( |\Delta([w'])|_\mathbf{1} - |\Delta([w'])|_\mathbf{3} \right)/4.$$

It corresponds to its total curvature divided by $2\pi$. Clearly, the turning number $\mathscr{T}([w])$ of a closed path $w$ belongs to $\mathbb{Z}$ (see [3, 6]).

**The convex hull of a finite set of points**    The lexicographic order $<$ on points of $\mathbb{R}^2$ or $\mathbb{Z}^2$ is such that $(x,y) < (x',y')$ when either $x < x'$ or $x = x'$ and $y < y'$. The *convex hull* of a finite set $S$ of points in $\mathbb{R}^2$ is the intersection of all convex sets containing these points and is denoted by $\mathrm{Conv}(S)$. $S$ being finite, it is clearly a polygon in the plane whose vertices are elements of $S$. The *upper convex hull* of $S$, denoted by $\mathrm{Conv}^+(S)$, is the clockwise oriented sequence of consecutive edges of $\mathrm{Conv}(S)$ starting from the lowest vertex and ending on the highest vertex. The *lower convex hull* of $S$, denoted by $\mathrm{Conv}^-(S)$, is the clockwise oriented sequence of consecutive edges of $\mathrm{Conv}(S)$ starting from the highest vertex and ending on the lowest vertex.

## 3    The Daurat-Nivat relation [4, 5]

We recall from Daurat and Nivat [15] that a *discrete* set $E$ is a subset of $Z^2$ and an element $(i,j) \in E$ corresponds to a unit square with vertices $(i \pm \frac{1}{2}, j \pm \frac{1}{2})$. One sets $P_{1/2} = (\frac{1}{2}, \frac{1}{2}) + Z^2$ and the salient and reentrant points are defined as follows:

**Definition 2.** (Daurat and Nivat[15]) *Let $E$ be a discrete set. Then*

(i) *A corner is a couple $(\mathbf{M},\mathbf{N})$ when $\mathbf{M} \in P_{1/2}$ and $\mathbf{N} \in Z^2$ and $\mathbf{M} - \mathbf{N}$ is in $(\{\pm\frac{1}{2}, \pm\frac{1}{2}\})$.*

(ii) *A corner $(\mathbf{M},\mathbf{N})$ is salient if $\mathbf{N} \in E$ and $\mathbf{M}$ is the extremity of two edges of the border of $E$ which are also sides of the square $\mathbf{N} + [-\frac{1}{2}, \frac{1}{2}]^2$.*

---

[1] In [3, 6], the authors introduced the notion of *winding number* of $w$ which is $4\mathscr{T}(w)$.

(iii) *A corner* (**M**,**N**) *is* reentrant *if* $\mathbf{N} \in E$ *and* **M** *is the extremities of two consecutive edges of the border E which are not sides of the square* $\mathbf{N} + [-\frac{1}{2}, \frac{1}{2}]^2$.

**Definition 3.** (Daurat and Nivat[15]) *The multiset of border-salient (resp. border-reentrant) points of a discrete set E, denoted $S_B(E)$ (resp. $R_B(E)$) is the multiset whose support is included in $P_{1/2}$ and such that for any* $\mathbf{M} \in P_{1/2}$, *the number* $\mathrm{mult}_{S_B(E)}(\mathbf{M})$ *(resp.* $\mathrm{mult}_{R_B(E)}(\mathbf{M})$*) is the number of* **N** *such that* (**M**,**N**) *is a salient corner (resp. reentrant corner).*

In other words, in their terminology modulo a translation by $(\frac{1}{2}, \frac{1}{2})$, a point **M** on the boundary of a polyomino *P* is salient (see Figure 3 (a)) if it belongs to the intersection of two consecutive sides of a square belonging to *P*. The point **M** is reentrant (see Figure 3 (b)) if it is the intersection of two consecutive edges of the contour of *P* which are sides of a square not belonging to *P*.



(a)                (b)

Figure 3: Salient and reentrant points in the terminology of Daurat and Nivat

In our framework, a salient point of the boundary *w* of a polyomino corresponds to a left turn (a **1** in $\Delta(w)$) and a reentrant one to a right one (a **3** in $\Delta(w)$), provided the traversal is done in a counterclockwise manner. The Daurat-Nivat [15] states that the *S* salient and *R* reentrant points in every polyomino are related by the formula
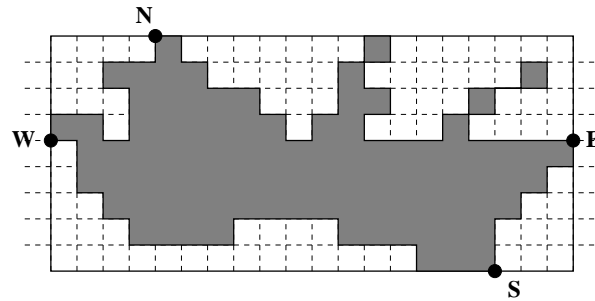
$$S - R = 4. \tag{4}$$



Figure 4: A region and its four extremal points.

The four *extremal* points are defined by the coordinates: **W** is the lowest intersection with the left side of the bounding rectangle *Q*, **N** the leftmost intersection with the top side, **E** the highest intersection with the right side, and **S** the rightmost intersection with the bottom side. Note that the four extra salient points can be canonically identified as **W**, **S**, **E** and **N**. Then, the Daurat-Nivat relation [15] may be restated as

**Proposition 3.** *The turning number of a boundary word w is* $\mathscr{T}(w) = \pm 1$.

Note that a boundary word *w* is *positively oriented* (counterclockwise) iff its turning number is $\mathscr{T}(w) = 1$.

**Remarks** 1. This rather elementary geometrical property is fundamental in proving a result about deciding whether a polyomino tiles the plane by translation or not.

2. This result can be extended easily to take into account hexagonal grids in which case the alphabet $\mathscr{F}$ must be extended to 6 letters, in which case the Daurat-Nivat relation becomes (see [4, 5] for more details)

$$S - R = 6. \tag{5}$$

3. The statement above includes closed paths not necessarily simple. It cannot therefore be used to determine whether a closed path is simple or not.

## 4 Path Intersection [9, 10]

Many problems in discrete geometry involve the analysis of the contour of discrete sets and many problems are solved by using linear algorithms in the length of the contour word. However, most of the time it is assumed that the path encoded by this word does not intersect itself. Checking non intersection amounts to check if a grid point is visited twice. Of course, one might easily provide an $\mathscr{O}(n\log n)$ algorithm where sorting is involved, or use hash tables providing a linear time algorithm on average but not in worst case. The underlying principle of the algorithm is to build a graph $\mathscr{G} = (N, R, T)$ where $N$ is a set of nodes associated to points of the plane, $R$ and $T$ are two distincts sets of oriented edges. The edges in $R$ give a quadtree structure on the nodes while the edges in $T$ are links from each node to its neighbors.

**The Algorithm** First, we assume that the path is coded by a word $w$ starting at the origin $(0,0)$, and stays in the first quadrant $\mathbb{N} \times \mathbb{N}$. This means that the coordinates of all points are nonnegative. Note that in $\mathbb{N} \times \mathbb{N}$, each point has exactly four neighbors with the exception of the origin $(0,0)$ which admits only two neighbors, namely $(0,1)$ and $(1,0)$, and the points on the half lines $(x,0)$ and $(0,y)$ with $x,y \geq 1$ which admit only three neighbors.

Now, assume that the node $(\boldsymbol{x},\boldsymbol{y})$ exists and that its neighbor $(x+1,y+0)$ does not. If $|\boldsymbol{x}| = |\boldsymbol{y}| = k$, then the translation $(x,y) + (1,0)$ is obtained in three steps by the following rules:

1. take the edge in $R$ to $f(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}[1..k-1], \boldsymbol{y}[1..k-1])$;

2. take (or create) the edge in $T$ from $f(\boldsymbol{x},\boldsymbol{y})$ to $\textcircled{z} = f(\boldsymbol{x},\boldsymbol{y}) + (1,0)$;

3. take (or create) the edge in $R$ from $\textcircled{z}$ to $\textcircled{z} \cdot (0, \boldsymbol{y}[k])$.

By Lemma 2, we have $\textcircled{z} \cdot (0, \boldsymbol{y}[k]) = (x+1, y+0)$, so that it remains to add the neighboring link $(\boldsymbol{x},\boldsymbol{y}) \overset{\mathbf{0}}{\dashrightarrow} (x+1, y+0)$. Then, a nonempty word $w \in \mathscr{F}^n$ is sequentially processed to build the graph $\mathscr{G}_w$, and we illustrate the algorithm on the input word $w = \mathbf{0011}$.

● *Initialization*: one starts with the graph consisting of the node $(0,0)$ marked as visited. For convenience, the *non-visited* nodes $(0,1), (1,0)$, and the links from $(0,0)$ to its neighbors are also added. This is justified by the fact that the algorithm applies to nonempty words. Since $(0,0)$ is an ancestor of all nodes, this ensures that every node has an ancestor linked with its neighbors. The *current node* is set to $(0,0)$ and this graph is called the *initial graph* $\mathscr{G}_\varepsilon$.
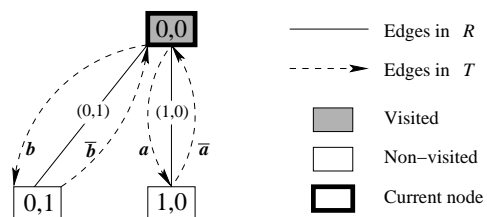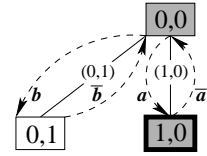


Figure 5: Initial graph $\mathscr{G}_\varepsilon$.

• *Read $w_1 = $ **0***: this corresponds to the translation $(0,0) + (1,0)$. A neighboring link labeled **0** starting from $(0,0)$ and leading to the node $(1,0)$ does exist, so the only thing to do is to follow this link and mark the node $(1,0)$ as visited. The current node is now set to $(1,0)$, and this new graph is called $\mathscr{G}_0$.

Figure 6: Graph $\mathscr{G}_\mathbf{0}$.

• *Read $w_2 = $ **0***: this time, there is no edge in $\mathscr{G}_0$ labeled **0** starting from $(1,0)$. Using the translation rules above, we perform:

(1) go back to the father $f(1,0) = (0,0)$;

(2) follow the link **0** to $(1,0)$;

(3) add node $(2,0) \sim (1,0) \cdot (0,0) = (10,00)$.

Then an edge from $(1,0)$ to $(2,0)$ with label **0** is added to $T$. Finally the node $(2,0)$ is marked as *visited*, and becomes the current node.
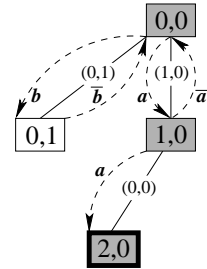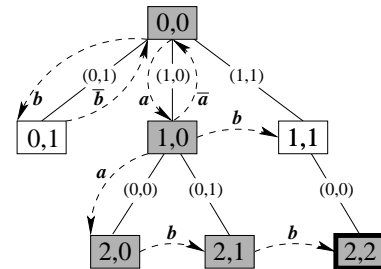
Figure 7: Graph $\mathscr{G}_\mathbf{00}$.

• *Read $w_3 = $ **1***: this amounts to perform the translation $(2,0) + (0,1)$. Since the edge to $f(2,0)$ is labeled by $(0,0)$, we know that the second coordinate of the current node $(2,0)$ is even. Therefore, $(2,1)$ and $(2,0)$ must be siblings, that is $f((2,0) + (0,1)) = f((2,0))$. What we need to do then is :

(1) go back to the father $f(2,0) = (1,0)$;

(2) follow the edge **1** if it exists;

Since it is does not exist, it must be created to reach the node $(2,1) \sim (10,01) = (1,0) \cdot (0,1)$. Again an edge from $(2,0)$ to $(2,1)$ with label **1** is added, $(2,1)$ is marked as *visited* and is now the current node.
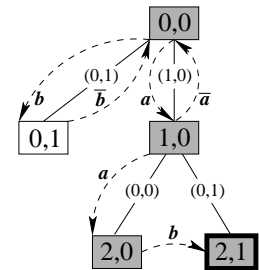
Figure 8: Graph $\mathscr{G}_{001}$.

• *Read $w_4 = $ **1***: since $f((2,1))$ has no neighboring link labeled by **1**, recursion is used to find (or build if necessary) the node corresponding to its translation by **1**. This leads to the creation of the node $(1,1) \sim (0,0) \cdot (1,1)$ marked as *non-visited*. Then, the node $(2,2) \sim (1,1) \cdot (0,0)$ is added, marked as *visited*, and becomes the current node. Note that a neighboring links between $(1,0)$ and $(1,1)$, $(2,1)$ and $(2,2)$ are added in order to possibly avoid searches.

Figure 9: Graph $\mathscr{G}_{0011}$.

This algorithm is linear and we refer to [10] for details about the complexity analysis which is rather involved.

## 5   Digital convexity[12, 7]

The notion of convexity does not translate trivially, and detecting if a discrete region of the plane is convex requires a deeper analysis. There are several (more or less) equivalent definitions of digital

convexity, depending on whether or not one asks the digital set to be connected. We say that a word $w$ is *digitally convex* if it is the boundary word of a finite 4-connected subset $S$ of $\mathbb{Z}^2$ such that $S = \mathrm{Conv}(X) \cap \mathbb{Z}^2$. Given such a region:
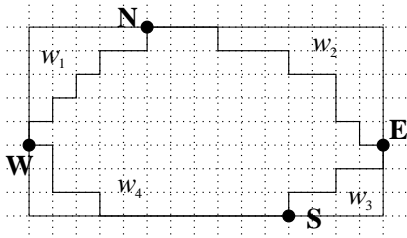
W is the lowest on the Left side;
N is the leftmost on the Top side;
E is the highest on the Right side;
S is the rightmost on the Bottom side;
So that $w \equiv w_1 w_2 w_3 w_4$.

We say that a word $w_1$ in $\mathscr{F}^*$ is *NW-convex* iff there are no integer points between the upper convex hull of the points $\{\phi_{w_i}(j)\}_{j=1...|w_i|}$ and the path $w_i$. Then we have

$$w_i \quad \text{is convex} \iff \rho^{i-1}(w_i) \quad \text{is NW-convex.}$$

Clearly, the convexity of $w$ requires the convexity of each $w_i$ for $i = 1,2,3,4$, and we have the following obvious property. Observe that if for some $i$, $w_i$ contains more than 2 letters, that is if $\mathrm{Alph}(\rho^{i-1}(w_i)) \not\subseteq \{\mathbf{0},\mathbf{1}\}$, then $w$ is not digitally convex.

**Theorem 4** ([7]). *A word $v$ is NW-convex iff its unique Lyndon factorization $l_1^{n_1} l_2^{n_2} \cdots l_k^{n_k}$ is such that all $l_i$ are primitive Christoffel words.*

For example, the Lyndon factorization of the word $v = \mathbf{1011010100010}$ is

$$v = (\mathbf{1})^1 \cdot (\mathbf{011})^1 \cdot (\mathbf{01})^2 \cdot (\mathbf{0001})^1 \cdot (\mathbf{0})^1,$$

where $\mathbf{0}$, $\mathbf{011}$, $\mathbf{01}$, $\mathbf{0001}$ and $\mathbf{0}$ are all Christoffel words.
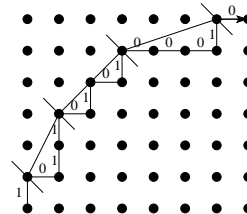
This result leads to a fast optimal algorithm for checking digital convexity of a boundary word. It is based on the linear time algorithms for computing the Lyndon factorization of the contour word and for the recognition of Christoffel factors which digital line segments. By avoiding arithmetical computations the algorithm is much simpler to implement and much faster in practice (see [7] for more details).

It is worth noting that many results about Sturmian words have been obtained by using geometrical properties. This close relation between the two domains raises a number of combinatorial problems such as enumeration of convex words of given length [30]. Another interesting property is the factorial closure of digitally convex words: while a geometrical proof is rather easy to obtain, it begs for a purely combinatorial proof. A formal language characterization is also a challenge.
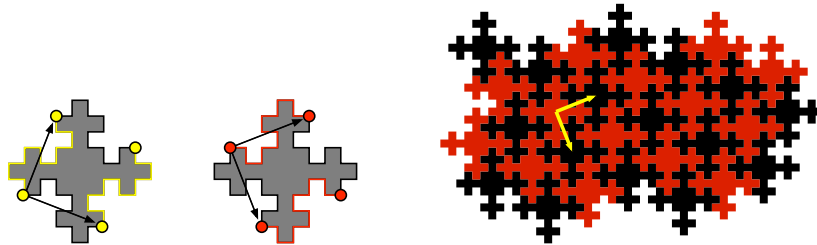
## 6  Tilings

Tilings appeared as one of the archetypes of the close relationship between art and mathematics, and are present in human history under various representations. The beautiful book of Grünbaum and Shephard [17] contains a systematic study of tilings, presenting a number of challenging problems (see also [2] for related work). For instance, the problem of designing an efficient algorithm for deciding whether a

given polygon tiles the plane becomes more tractable when restricted to polyominoes, that is, subsets of the square lattice $\mathbb{Z}^2$ whose boundary is a non-crossing closed path. Indeed, while a sufficient condition is provided by the *Conway criterion* in [32], one of its consequences is that the only objects that tile the plane by translation in two directions are generalizations of parallelograms and parallel hexagons, hexagons whose opposite sides are equal and parallel (see [32] p. 225 for more details). One may consider these tiles are continuous deformations of either the *unit square* or the *regular hexagon*.

**Tiling the plane by translating a single polyomino** ([13, 8])   Such a polyomino is called *exact* in [33] and Beauquier and Nivat [1] characterized them by showing that their boundary word satisfies the equation $b(P) = X \cdot Y \cdot Z \cdot \widehat{X} \cdot \widehat{Y} \cdot \widehat{Z}$, where at most one of the variables is empty and where $\widehat{W} = \rho^2(\widetilde{W})$. This condition is referred to as the BN-factorization. An exact polyomino is said to be a *hexagon* if none of the variables $X$, $Y$, $Z$ is empty and a *square* if one of them is so. While decidability was already established in [33], recently, a $\mathcal{O}(n\log^3 n)$ was designed for deciding if a word $w \in \mathscr{F}$ tiles the plane by translation [8, 11]. It uses several data structures that include radix-trees, for checking that $w$ is a closed non crossing path [10], and suffix-trees for building a tricky algorithm for checking the BN-factorization [13].

**Square tiles**   In this case there is a linear algorithm for deciding if a word is a square [8]. Moreover, it turns out that a square has at most two distinct square tilings [26, 27], which means that the BN-factorizations are distinct and therefore not conjugate. Motivated by the attempt of characterizing the square tiles, the study of equations of the form $A B \widetilde{A} \widetilde{B} \equiv X Y \widetilde{X} \widetilde{Y}$ lead to a subset of solutions in bijection with the BN-factorizations. It turns out that the solutions are strongly related to local periodicity involving palindromes and conjugate words [25, 29]. They also give some insight on their shape since palindromes represent symmetric sides. An interesting fact is that some infinite families of such tiles, namely the Fibonacci tiles have a fractal characteristic and are connected with some problems in number theory [28], while Christoffel words yield another infinite family of squares [24].



**Remark**   Again a number of problems are raised by these connections about the enumeration and generation of tiling polyominoes. It turns out that the prototype of a square is the basic one cell polyomino, and it suffices to replace each pair of sides by homologous paths. It remains only to check that theses paths do not intersect, which is achieved in linear time thanks to the optimal algorithm we have. For the generation of double square tiles, the prototype is the cross polyomino, and their generation is a bit more involved. The case of hexagon grids deserves also some attention [8].

# References

[1] D. Beauquier & M. Nivat (1991): *On translating one polyomino to tile the plane*. Discrete Comput. Geom. 6(6), pp. 575–592, doi:10.1007/BF02574705.

[2] Peter Brass, William Moser & János Pach (2005): *Research problems in discrete geometry*. Springer, New York.

[3] S. Brlek, G. Labelle & A. Lacasse (2005): *Algorithms for polyominoes based on the discrete Green theorem*. Discrete Appl. Math. 147(2-3), pp. 187–205, doi:10.1016/j.dam.2004.09.011.

[4] S. Brlek, G. Labelle & A. Lacasse (2005): *A note on a result of Daurat and Nivat*. In C. de Felice & A. Restivo, editors: *Proc. DLT 2005, 9-th Int. Conf. on Developments in Language Theory, Palermo, Italia, 4–8*, LNCS 3572, Springer-Verlag, pp. 189–198.

[5] S. Brlek, G. Labelle & A. Lacasse (2006): *Properties of the contour path of discrete sets*. Internat. J. Found. Comput. Sci. 17(3), pp. 543–556, doi:10.1142/S012905410600398X.

[6] S. Brlek, G. Labelle & A. Lacasse (2006): *Shuffle operations on lattice paths*. In M.Rigo, editor: *Proc. CANT2006, Int. School and Conf. on Combinatorics, Automata and Number theory, Liège, Belgium, 8–19*, University of Liège.

[7] S. Brlek, J.-O. Lachaud, X. Provençal & C. Reutenauer (2009): *Lyndon + Christoffel = digitally convex*. Pattern Recognition 42(10), pp. 2239 – 2246, doi:10.1016/j.patcog.2008.11.010.

[8] S. Brlek, X. Provençal & Jean-Marc Fédou (2009): *On the tiling by translation problem*. Discrete Appl. Math. 157(3), pp. 464–475, doi:10.1016/j.dam.2008.05.026.

[9] Srecko Brlek, Michel Koskas & Xavier Provençal (2009): *A Linear Time and Space Algorithm for Detecting Path Intersection*. In Srečko Brlek, Christophe Reutenauer & Xavier Provençal, editors: *Proc. DGCI 2009, 15th IAPR Int. Conf. on Discrete Geometry for Computer Imagery, Montréal, Canada*, LNCS 5810, Springer, pp. 397–408.

[10] Srecko Brlek, Michel Koskas & Xavier Provençal (2011): *A linear time and space algorithm for detecting path intersection in $\mathbb{Z}^d$*. Theoretical Computer Science 412(36), pp. 4841 – 4850, doi:10.1016/j.tcs.2011.04.019.

[11] Srečko Brlek & Xavier Provençal (2006): *On the problem of deciding if a polyomino tiles the plane by translation*. In Jan Holub & Jan Žďárek, editors: *Proceedings of the Prague Stringology Conference '06*, ISBN80-01-03533-6, Czech Technical University in Prague, Prague, Czech Republic, pp. 65–76.

[12] Srečko Brlek, Jacques-Olivier Lachaud & Xavier Provençal (2008): *Combinatorial view of digital convexity*. In David Coeurjolly, Isabelle Sivignon, Laure Tougne & Florent Dupont, editors: *Proc. DGCI 2008, 14th IAPR Int. Conf. on Discrete Geometry for Computer Imagery, Lyon, France*, LNCS 4992, Springer, pp. 57–68.

[13] Srečko Brlek & Xavier Provençal (2006): *An Optimal Algorithm for Detecting Pseudo-squares*. In Attila Kuba, László G. Nyúl & Kálmán Palágyi, editors: *Proc. DGCI 2006, 13th IAPR Int. Conf. on Discrete Geometry for Computer Imagery, Szeged, Hungary, October 25-27, 2006, Proceedings*, LNCS 4245, Springer, pp. 403–412.

[14] Maxime Crochemore, Christophe Hancart & Thierry Lecroq (2007): *Algorithms on strings*. Cambridge University Press, Cambridge, doi:10.1017/CBO9780511546853.

[15] A. Daurat & M. Nivat (2003): *Salient and Reentrant Points of Discrete Sets*. In A. del Lungo, V. di Gesu & A. Kuba, editors: *Proc. IWCIA'03, Int. Workshop on Combinatorial Image Analysis*, Electronic Notes in Discrete Mathematics 12, Elsevier Science, Palermo, Italia, pp. 208–219.

[16] Jean-Pierre Duval (1983): *Factorizing words over an ordered alphabet*. J. Algorithms 4(4), pp. 363–381.

[17] Branko Grünbaum & G. C. Shephard (1987): *Tilings and patterns*. W. H. Freeman and Company, New York.

[18] Reinhard Klette & Azriel Rosenfeld (2004): *Digital straightness—a review*. Discrete Appl. Math. 139(1-3), pp. 197–230, doi:10.1016/j.dam.2002.12.001.

[19] Donald E. Knuth (1973): *The art of computer programming. Volume 3*. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont.

[20] M. Lothaire (1997): *Combinatorics on words*. Cambridge Mathematical Library, Cambridge University Press, Cambridge, doi:`10.1017/CBO9780511566097`.

[21] M. Lothaire (2002): *Algebraic combinatorics on words*. Encyclopedia of Mathematics and its Applications 90, Cambridge University Press, Cambridge.

[22] M. Lothaire (2005): *Applied combinatorics on words*. Encyclopedia of Mathematics and its Applications 105, Cambridge University Press, Cambridge.

[23] Stanislaw Maloň & Herbert Freeman (1961): *On the encoding of arbitrary geometric configurations*. IRE Trans. EC-10, pp. 260–268.

[24] A. Blondin Massé, S. Brlek, A. Garon & S. Labbé (2009): *Christoffel and Fibonacci Tiles*. In S. Brlek, X. Provençal & C. Reutenauer, editors: *Proc. DGCI 2009, 15th IAPR Int. Conf. on Discrete Geometry for Computer Imagery, LNCS* 5810, Springer-Verlag, MontrÃ©al, Canada, pp. 67–78.

[25] A. Blondin Massé, S. Brlek, A. Garon & S. Labbé (2009): *Palindromes and local periodicity*. In: *Words 2009, 7th Int. Conf. on Words.* Electronic proceedings.

[26] A. Blondin Massé, S. Brlek, A. Garon & S. Labbé (2011): *Every polyomino yields at most two square tilings*. In: *Lattice Paths 2010, 7th Int. Conf. on Lattice paths combinatorics and applications, July 4–7, 2011, Siena, Italy*, pp. 57–61.

[27] A. Blondin Massé, S. Brlek & S. Labbé: *A square tile fills the plane by translation in at most two distinct ways*. Submitted to Discrete Appl. Math.

[28] A. Blondin Massé, S. Brlek, S. Labbé & M. Mendès France: *Fibonacci snowflakes*. To appear in Annales des Sciences Mathématiques du Québec.

[29] A. Blondin Massé, S. Brlek, A. Garon & S. Labbé (2011): *Equations on palindromes and circular words*. Theoretical Computer Science 412(27), pp. 2922 – 2930, doi:`10.1016/j.tcs.2010.07.005`.

[30] Xavier Provençal (2011): *Minimal non-convex words*. Theoretical Computer Science 412(27), pp. 3002 – 3009, doi:`10.1016/j.tcs.2010.06.025`.

[31] Christophe Reutenauer (1993): *Free Lie algebras. London Mathematical Society Monographs. New Series* 7, The Clarendon Press Oxford University Press, New York.

[32] Doris Schattschneider (1980): *Will it tile? Try the Conway criterion!* Math. Mag. 53(4), pp. 224–233, doi:`10.2307/2689617`.

[33] H. A. G. Wijshoff & J. van Leeuwen (1984): *Arbitrary versus periodic storage schemes and tessellations of the plane using one type of polyomino*. Inform. and Control 62(1), pp. 1–25, doi:`10.1016/S0019-9958(84)80007-8`.