

Number of Words Characterizing Digital Balls on the Triangular Tiling

Benedek Nagy^{1,2}(✉)

¹ Department of Computer Science, Faculty of Informatics,
University of Debrecen, Debrecen, Hungary

² Department of Mathematics, Faculty of Arts and Sciences,
Eastern Mediterranean University, Famagusta, North Cyprus, Turkey
nbenedek.inf@gmail.com

Abstract. In this paper, digital balls on two regular tessellations of the plane, on the square and on the triangular grids are analyzed. The digital balls are defined by digital, i.e., path based distance functions. The paths (built by steps to neighbor pixels) from the center to the points (pixels) of the balls are described as traces and generalized traces, respectively, on these grids. On the square grid, there are two usual types of neighborhood, and thus, the number of linearizations of these traces is easily computed by a binomial coefficient. The number of linearizations gives the number of words that describe the same digital ball. In the triangular tiling there are three types of neighborhood, moreover, this grid is not a lattice, therefore, the possible paths that define a ball form a more complicated set, a kind of generalized trace. The linearizations of these traces are described by an associative rewriting system, and, as a main combinatorial result, the number of words that define the same ball is computed.

Keywords: Digital distance · Shortest paths · Neighborhood sequences · Non-traditional grids · Digital disks · Combinatorics · Traces · Trajectories · Generalized traces

1 Introduction

The history of digital geometry has begun in the 1960's by the papers [18, 19] in which two types of widely used neighborhood, namely the cityblock and chessboard neighborhood, on the square grid were described. Based on them digital distances are defined. Since both of them are very rough approximations of the Euclidean distance, it is recommended to use them alternately in a path. This octagonal distance [19] (the name comes from the shape of the digital balls defined by this digital distance) was generalized by allowing any predefined sequence of neighborhood in paths [1, 12]. The theory of neighborhood sequences was also generalized to the triangular grid (i.e., the triangular tiling), see, e.g., [9, 11, 15]. Already in [19] combinatorial problems based on digital distances were investigated, namely 'path counting'. Our aim here is somewhat similar, a related

combinatorial property of some digital distances will be analyzed. We note that digital balls (or disks) based on digital distances are usually used to show some properties of the digital distances [5, 13, 17]. Digital grids and digital distances can be used in various fields of computer science and information technology, e.g., in image processing, in computer graphics, in networks. There are somewhat similar concepts, the digitized circles and disks: they are representing the Euclidean circles and disks on a digital grid. In this paper disks based on digital distances are examined and the term ‘digital ball’ is used to refer them.

Sequential and parallel neighborhood operations play essential roles in image processing as it is already mentioned in [18]. However, a concept for formal description of sequential and parallel executions of various operations comes from another source. In 1977, analyzing basic networks, Mazurkiewicz introduced the concept of partial commutations (see, e.g., [8]). Two parallel events are independent, they commute if their executing order can be arbitrary in a sequential simulation. By using the concept of commutations, the work of the concurrent systems can be described by traces. In these systems some (pairs of) elementary processes (i.e., atomic actions; they are represented by the letters of the alphabet) may depend on each other, and some of them can be (pairwise) independent. The order of two consecutive independent letters can be arbitrary, in this way traces are a kind of generalizations of words. Traces and trace languages play important roles in describing parallel events and processes. For simplifying the descriptions linearizations of trace languages is frequently used; they are sets of words representing traces of the trace languages. A special two-dimensional representation of some traces is given by trajectories (usually on the square grid). These trajectories were also used to describe syntactic constraints for shuffling two parallel events (described by words) in [7]. We believe that trajectories connected to other regular grids are also interesting. A kind of generalization of traces is presented in [6], where apart from the usual permutation rules, some other rewriting rules, namely serialisation equations, are also allowed. As we will see, the sets of words describing the same digital balls on the triangular tiling are generalized traces.

In this paper, two regular tessellations of the two-dimensional plane are used: the square and the triangular grids. Traces and words describing digital balls are analyzed. As we will see, the case of the square grid is very simple, it is shown only for the analogy. Descriptions of a ball on the square grid form a trace, in which the order of the two types of steps can be arbitrary. The triangular grid (tiling) is not a lattice, therefore, the sets of words describing the same balls form more complicated sets.

2 Preliminaries

We assume that the reader is familiar with some of the basic concepts of formal language theory. However, to have a self-contained paper, we recall briefly the concepts that are necessary. Let V be a finite set of symbols, called alphabet, its elements are called letters. The set of all (finite length) words over V is

denoted by V^* . This set also includes the empty word λ . For any $w \in V^*$, $|w|$ denotes the length of w . Let the alphabet be an ordered set $\{a_1, \dots, a_n\}$. For any $a_i \in V$, $|w|_{a_i}$ denotes the number of occurrences of letter a_i in w . Consequently, $\sum_{i=1}^n |w|_{a_i} = |w|$ for every word $w \in V^*$. The Parikh mapping assigns the *Parikh-vector*, $W(|w|_{a_1}, \dots, |w|_{a_n})$, to every word w .

There are three regular tessellations of the plane: the square-, the triangular- and the hexagonal grids. In this paper, we are working on the first two of them. The types and number of neighbors present depend on the grid [2, 11]: usually two types are used in the square grid and three types in the triangular tiling (as we detail in Sects. 3 and 4, respectively). Let V denote the set of possible neighborhood relations. The infinite sequence $B = (b(i))_{i=1}^\infty$ where $b(i) \in V$ represents a possible neighborhood for all $i \in \mathbb{N}$, is called a *neighborhood sequence*. In fact, neighborhood sequences are infinite words over V , thus, the set of them is denoted by V^∞ . Let p and q be two pixels (also called points) of the grid and $B = (b(i))_{i=1}^\infty$ be a neighborhood sequence. A finite point sequence $\Pi(p, q; B)$ of the form $p = p_0, p_1, \dots, p_m = q$, where p_{i-1}, p_i are $b(i)$ -neighbors for $1 \leq i \leq m$, is called a B -path from p to q . We write $m = |\Pi(p, q; B)|$ for the length of the path. Denote by $\Pi^*(p, q; B)$ a shortest path from p to q , and set $d(p, q; B) = |\Pi^*(p, q; B)|$. We call $d(p, q; B)$ the B -distance from p to q . For $k \in \mathbb{N}$, let $D_c^B(k) = \{p \mid d(c, p; B) \leq k\}$ be the digital ball occupied by B in k steps starting from the pixel c : a digital ball $D_c^B(r)$ is the set of exactly those pixels (points) that have digital distance from a point c , called center, at most a given value r , called radius.

If the reader is not familiar with traces, she or he is referred to, e.g., [3, 4]; here we recall only the basic concepts. Let the dependency δ be a reflexive and symmetric binary relation on V . The independency relation, $\iota = (V \times V) \setminus \delta$ is also called commutation. If $(a, b) \in \iota$, then the words $uabv$ and $ubav$ are equivalent to each other for any $u, v \in V^*$. The equivalence relation induced by ι on V^* is called partial commutation. The set of words that are equivalent to w is denoted by $[w]$ and called a *trace*. The linearizations of the traces are exactly those words that belongs to the trace. A trace is represented by the corresponding set of these words. Thus, in traces some of the neighbor letters can be permuted, e.g., the interleaving equation $ab = ba$ can be written for some letters $(a, b) \in \iota$. In generalized traces [6], apart from these equations, serialisation equations of the form $c = de$ ($c, d, e \in V$) are also allowed, representing the serialisation of c into two consecutive steps (letters) d and e . Consequently, a *generalized trace*, denoted by $[[w]]$, for a word $w \in V^*$ is a set of words equivalent to w . Linearizations of generalized traces are understood analogously. Notice that without using any serialisation equations, i.e., with only interleaving equations (i.e., partial commutations, permutations) $[[w]] = [w]$ for every word w .

We also recall the definition of associative calculus: $\mathcal{C} = (V, P)$ is an *associative calculus*, where V is a finite alphabet and P is a finite set of productions (rewriting rules). Each rewriting rule is an element of $V^* \times V^*$. A rule is usually written in the form $u \Leftarrow v$, where $u, v \in V^*$. Let $w \in V^*$ be given, we say that w' is obtained from w applying the rewriting rule $u \Leftarrow v$, if there exist $w_1, w_2 \in V^*$

such that either $w = w_1uw_2$ and $w' = w_1vw_2$, or $w = w_1vw_2$ and $w' = w_1uw_2$. Actually, w can also be obtained from w' by the same production, thus we may use the notation $w \Leftrightarrow w'$. By the reflexive and transitive closure of \Leftrightarrow , the relation \Leftrightarrow^* is defined. Observe that the calculus \mathcal{C} defines an equivalence relation on V^* . The equivalence class represented by w is denoted by $\mathcal{C}(w) = \{w' \mid w \Leftrightarrow^* w'\}$. An example is $\mathcal{C} = (\{a, c, d, g, f, o, r, t\}, \{at \rightleftharpoons rog, c \rightleftharpoons f, d \rightleftharpoons fr\})$, in which $\mathcal{C}(cat) = \{cat, fat, crog, frog, dog\}$. One may observe that the associative calculus is very general in the sense that a rule may allow to rewrite any word by any other.

3 Traces and Trajectories on the Square Grid

Usually the term digital plane or digital grid refers to \mathbb{Z}^2 , as the square grid is the most known and most used grid both in theory and in practice (e.g., in image processing, in computer graphics and at 2D languages). The two basic neighborhood, the cityblock and the chessboard neighborhood, on the square grid were described by Rosenfeld and Pfaltz in [19]. We use the terms, 1-neighbor and 2-neighbor for them, respectively (since in this way, they can easily be generalized to other grids). Using the Cartesian coordinate frame, one can define these neighborhood relations as follows. Let $p(p(1), p(2))$ and $q(q(1), q(2))$ be two points (pixels), they are m -neighbors ($m \in \{1, 2\}$) if $|p(i) - q(i)| \leq 1$, for $i \in \{1, 2\}$, and $|p(1) - q(1)| + |p(2) - q(2)| \leq m$.

The theory of neighborhood sequences is well developed for \mathbb{Z}^2 [1, 12, 16, 21].

Neighborhood sequences are infinite, but a given ball depends only on the first r elements of the sequence [13], and therefore, here we use finite words over $V = \{1, 2\}$ to describe digital balls, hence redefining the digital balls as follows. A word $w \in V^*$ defines a digital ball:

$$D_c(w) = \{p \mid \text{there is a prefix of } w \text{ such that it describes a path from } c \text{ to } p\}.$$

Figure 1 shows an example. We note here that a greedy algorithm to compute a shortest path and formula to compute the distance for any neighborhood sequence can be found in [12, 14, 16].

3.1 Number of Words Describing the Same Ball on the Square Grid

Let us start this subsection with some relatively simple results about digital balls on the square grid. Let the center of the ball be $c(c(1), c(2))$. Then, (based on the formula for distances, e.g., in [14]),

$$D_c(w) = \{p(p(1), p(2)) \mid \begin{array}{l} |p(1) - c(1)| \leq |w|, |p(2) - c(2)| \leq |w|, \\ |p(1) - c(1)| + |p(2) - c(2)| \leq |w| + |w|_2 \end{array}\}. \quad (1)$$

Consequently, the number of pixels in the ball obtained from w is $(2|w| + 1)^2 - 4 \frac{|w|_1(|w|_1 + 1)}{2}$.

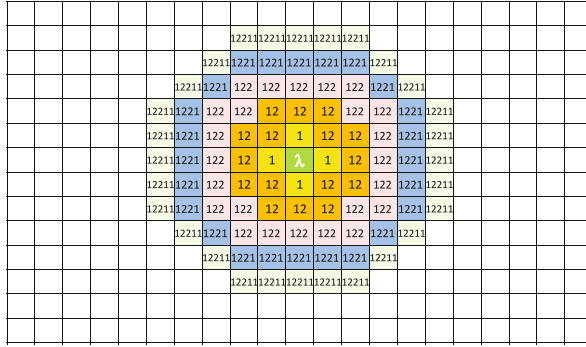


Fig. 1. The digital ball defined by the word $w = 12211$. Each pixel contains a prefix of w that gives a path to there from the center marked by λ . Same color is used to highlight pixels that have the same distance from the center (Color figure online).

Now, let us turn to the main topic. By Eq. (1), it is clear that the order of the letters of w does not matter, only the numbers of their occurrences are important. Consequently, instead of a word w , its Parikh-vector, i.e., the vector $W(|w|_1, |w|_2)$ uniquely determines the digital ball $D_c(w)$. In this way, actually, the balls are represented by traces. The two letters of V are independent of each other. Therefore, the linearizations of these traces contain the commutative closure of the words w .

Now we show how the number of words describing the same ball can be computed. The radius of a ball is $r = |w|$, and, thus, the number of words describing ball $D_c(w)$ is $\binom{|w|}{|w|_1}$. That is the number of words in the equivalence class $[w]$ defined by w .

One can easily prove using Eq. (1) that the balls with the same radius form a well-ordered set:

Proposition 1. $D_c(w) \subsetneq D_c(w')$ if and only if $|w|_2 < |w'|_2$, where $|w| = |w'| = r$.

We note one important fact about the square grid: it is a lattice, and thus, the steps in a path can freely be permuted. In the next section, the triangular tiling is considered; and that is not a lattice. Therefore, as we will show, to describe the balls on the triangular tiling is a more challenging and more complex task.

4 Describing Digital Balls on the Triangular Tiling

4.1 The Triangular Tiling

The triangular grid (i.e., the triangular tiling), preserving the symmetry of the grid, can be described by three coordinates [9,11,13,20]. There are two types of pixels (by orientation): the even pixels have zero-sum triplets, while the odd pixels have one-sum triplets. Formally, the neighborhood relations are defined

as follows. Let $p(p(1), p(2), p(3))$ and $q(q(1), q(2), q(3))$ be two pixels, they are m -neighbors ($m \in \{1, 2, 3\}$) if

- $|p(i) - q(i)| \leq 1$, for $i \in \{1, 2, 3\}$, and
- $|p(1) - q(1)| + |p(2) - q(2)| + |p(3) - q(3)| \leq m$.

Various neighborhoods and the used coordinate system are shown in Fig. 2. Observe that the 3 ‘closest’ neighbors, the 1-neighbors of a pixel have the opposite parity (shape). There are 6 more 2-neighbors and their parity is the same as the parity of the original pixel, while the 3 additional 3-neighbors have the opposite parity.

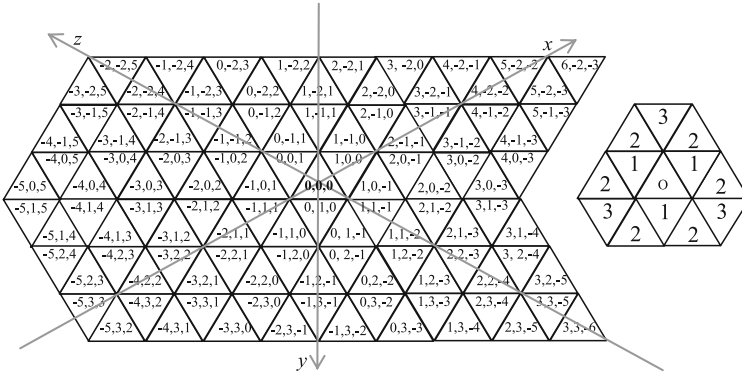


Fig. 2. The triangular tiling: pixels with coordinates (left) and various neighbors (right) that are usually considered [2].

The neighborhood sequences on the triangular tiling are infinite sequences over the alphabet $V = \{1, 2, 3\}$. Paths, distances and digital balls are defined analogously to the case of the square grid. However, there are neighborhood sequences that generate exactly the same distance functions. The next concept and result are recalled from [10].

Let $B, B' \in V^\infty$ be two neighborhood sequences. B' is called the *minimal equivalent neighborhood sequence* of B , if the following two conditions hold:

- $d(p, q; B) = d(p, q; B')$ for all pairs of points p, q , and
- for each neighborhood sequence B'' , if $d(p, q; B) = d(p, q; B'')$ for all pairs of points p, q , then $b'(i) \leq b''(i)$ for all i .

Lemma 1. *The minimal equivalent neighborhood sequence B' of B is uniquely determined, and is given by*

$$b'(i) = \begin{cases} b(i), & \text{if } b(i) < 3; \\ 3, & \text{if } b(i) = 3 \text{ and } i \text{ is the smallest index with this property;} \\ 3, & \text{if } b(i) = 3 \text{ and there is an } l : l < i, b(l) = 3 \text{ and} \\ & \sum_{k=j+1}^{i-1} b'(k) \text{ is odd, with } j = \max \{l | l < i, b'(l) = 3\}; \\ 2, & \text{otherwise.} \end{cases}$$

One may restrict the usage of neighborhood sequences to the minimal equivalent neighborhood sequences by representing each digital distance by only one, uniquely determined neighborhood sequence. Thus, in this paper we use only words to describe digital balls that are prefixes of some neighborhood sequences that are minimal equivalent neighborhood sequences. In this way, by counting the words describing the same ball, roughly speaking, we count the number of distance functions which could define the same ball.

Definition 1. *The subwords $u, v \in V^*$ are region equivalent if $D_c(w_1uw_2) = D_c(w_1vw_2)$ for every $w_1, w_2 \in V^*$.*

One can easily establish the fact that region equivalence is an equivalence relation among words of V^* .

Proposition 2. *The subword 2 is region equivalent to subword 11.*

Proof. By a step to a 1-neighbor at most 1 of the coordinates is changing and it is changing by ± 1 depending on the parity of the pixel. By a step to a 2-neighbor at most 2 of the coordinates are changing and if exactly 2 of them are changing then they change by $+1$ and -1 . One can see that exactly those points are reached by a 2-step as the points that are reached by two consecutive 1-steps. □

The previous proposition shows an effect that is not present on the square grid.

Proposition 3. *The subword 12 is region equivalent to subword 21.*

Proof. By Proposition 2 one can see, that 12 is region equivalent to 111, and thus, to 21. □

Proposition 4. *The subword 32 is region equivalent to subwords 23. Moreover, they are region equivalent to 33.*

Proof. At 3-steps all the three coordinates can be changed (by ± 1). However, there are only two types of pixels, namely 0-sum (even) and 1-sum (odd) pixels. Therefore, one can see that by two consecutive 3-steps those points can be reached for which one of the coordinates is changed with at most $+2$, the other with at most -2 , and the third one by at most ± 1 , depending on the parity. Actually, as one may list those points, they are the same as the points reached by a 2-step and by a 3-step independently on their order. □

One can see that by Proposition 4 not only the positions, but the letters can also be changed, and, actually, the sum of their values can also be changed. However, as we have restricted ourselves only for neighborhood sequences that are minimal equivalent neighborhood sequences, thus, the subword 33 is excluded from our analysis (together with all its related subwords having even sum between two neighbor occurrences of letter 3, e.g., 3223 or 31213).

The independence of letters 1 and 2 (Proposition 2) and of letters 2 and 3 (stated in Proposition 4) are analogous to the independence of letters 1 and 2

on the square grid. But, on the triangular tiling we have letters that are not independent of each other. To show this, one can consider the following example: $D_c(13) \neq D_c(31)$. For instance, with $c(0, 0, 0)$ $p(-1, 2, -1) \in D_c(13)$, but $p(-1, 2, -1) \notin D_c(31)$; and $q(1, -2, 1) \in D_c(31)$, but $q(1, -2, 1) \notin D_c(13)$. More formally, we state the following.

Proposition 5. *The subword 13 is not region equivalent to subword 31. More precisely, for any pair of words $w, u \in V^*$, $D_c(w13u) \neq D_c(w31u)$.*

Proof. The proof consists of two parts:

In the first part we prove the statement for words having 13 and 31 as suffixes. Thus, first we show that for any word w the words $w13$ and $w31$ describe different balls. Then, let us start from the ball $D_c(w)$ for any $w \in V^*$. Consider those point(s) of $D_c(w)$ that has smallest y coordinate (i.e. the second coordinate), let denote this set of points by $T_c(w)$ (top points of $D_c(w)$). Then, there are two cases, according to the parity of these point(s).

- If, there is an odd point $p(p(1), p(2), p(3))$ in $T_c(w)$, then, obviously $p(1) \geq 0, p(2) \leq 0, p(3) \geq 0, p(1) + p(2) + p(3) = 1$ because $p(2)$ can be minimal only under these conditions. Furthermore, the odd point $q(p(1)+1, p(2)-2, p(3)+1)$ is in $D_c(w13)$ (and also in $T_c(w13)$), but not in $D_c(w31)$ (since no odd point having y coordinate $p(2) - 2$ is in this ball).
- In the case, when $T_c(w)$ contains only even point(s), chose one of them, let us say, $p(p(1), p(2), p(3))$ (in this case, $p(1) \geq 0, p(2) \leq 0, p(3) \geq 0, p(1) + p(2) + p(3) = 0$). Then, the even point $q(p(1)+1, p(2)-2, p(3)+1)$ is in $D_c(w31)$ (and also in $T_c(w31)$), but not in $D_c(w13)$ (no points with y coordinate $p(2) - 2$ can be in this ball).

Thus, in both cases, the balls defined by $w13$ and $w31$ differ, for every $w \in V^*$. Moreover, the sets $T_c(w31)$ and $T_c(w13)$ are not the same, either the values of y coordinate differ or the parities of the points they contain are not the same (e.g., one of them does not contain odd points, but the other does).

The second part of the proof uses induction on the length of the word u , to show that words $w13u$ and $w31u$ are not defining the same ball, moreover the sets $T_c(w13u)$ and $T_c(w31u)$ differ either by the y coordinate of the points they contain, or by the parities of the points they contain does not match.

- The base of the induction, the case when $u = \lambda$, i.e., its length is 0 was proven in the first part.
- As induction hypothesis, let us assume that for every $w \in V^*$ and $u \in V^*$ having $|u| = n$ for some nonnegative integer n , the words $w13u$ and $w31u$ are not defining the same ball, moreover, the sets $T_c(w13u)$ and $T_c(w31u)$ have the mentioned difference.
- We state that $w31u'$ is defining another ball than $w13u'$, when $|u'| = n + 1$ and the difference of sets $T_c(w13u')$ and $T_c(w31u')$ is also preserved.

To prove this statement, observe that u' can be written exactly one of the following forms: $u' = u1$ or $u' = u2$ or $u' = u3$ for some $u \in V^*$ with $|u| = n$. Thus,

for $w13u$ and $w31u$ we can apply the induction hypothesis: they are describing different balls. Now, the proof is going by cases:

Let us assume that u does not contain the letter 3.

- If $u' = u1$, then there are two cases, according to the possible differences of the sets $T_c(w13u)$ and $T_c(w31u)$:
 - If the y coordinate of the points contained in these sets differ, then it is clear that by a 1-step, if the y value is changed (decreased by 1), only even point(s) are reached. If it is not changed, then odd point(s) with the same y coordinate are also reached and become elements of the appropriate new set T_c . In this way, the sets $T_c(w13u1)$ and $T_c(w31u1)$ differ, and thus the words $w13u1$ and $w31u1$ are not describing the same ball.
 - If $T_c(w13u)$ and $T_c(w31u)$ contains points with the same value of y coordinate, then, by our assumption, they must differ by the parity of the points they contain: one of them does not contain odd points. By the additional 1-step the corresponding set T_c will be extended with odd point(s), while the other set that had already odd points will change to contain only even points with one less second coordinate value. In this way, the difference of $T_c(w13u1)$ and $T_c(w31u1)$ is proven.
- If $u' = u2$, then we can face to the same cases again. However, with a 2-step the second coordinate value is decreased by 1 in both cases when the sets $T_c(w13u)$ and $T_c(w31u)$ are changed to $T_c(w13u2)$ and $T_c(w31u2)$, but the parity of the points contained in these sets are copied from the previous sets, respectively. If $T_c(w13u)$ and $T_c(w31u)$ had points with different value of y coordinate, then so $T_c(w13u2)$ and $T_c(w31u2)$ do. If the parities of points of $T_c(w13u)$ and $T_c(w31u)$ do not match, then the parities of points of $T_c(w13u2)$ and $T_c(w31u2)$ also mismatch. Thus, this case is proven.
- Finally, the case $u' = u3$ is analyzed. Observe, that in this case both $w13u3$ and $w31u3$ contains at least twice the letter 3. Moreover, the sum of the values 1's and 2's between these two occurrences of 3 is even in one case, and odd in the other case (the sum of the values of u and one more, respectively). As such, by Lemma 1, in exactly one of the cases, this last 3 can be changed to a value 2 without changing the ball that is described by that word. Consequently, the balls $D_c(w13u3)$ and $D_c(w31u3)$ differs: some points that are in the ball having last element 3 cannot be in the ball in which this value is changed to 2, the sum of the values in these two words differ, and since both of them are prefixes of minimal equivalent neighborhood sequences, there are some points p for which the sum of the coordinate differences from point c is exactly the sum of the elements and p is contained in the respective ball (see, e.g., [15]).

This last case show also, why we can assume that u does not contain the letter 3.

Thus, the proof is finished, the subwords 13 and 31 cannot be replaced in any context without changing the described balls. □

As we can see, on the triangular tiling we do not have a statement analogous to Proposition 1. To work with digital balls on the triangular tiling is more

interesting. The order of steps becomes important, because the triangular tiling is not a lattice, i.e., some of the grid-vectors do not translate the grid to itself.

4.2 Generalized Traces Describing Digital Balls

We refer to [13,17] for description of the possible shapes of digital balls and approximation of the Euclidean disks by them. Here, we are doing related things: an associative calculus is applied to describe words representing digital balls, and it is shown how these sets of words can be seen as generalized traces.

Now we present an associative calculus that provides all the words that are region equivalent to the one the process starts with (that is, they describe the same ball). The independence relation contains pairs of letters such that one of the letters is 2. In terms of generalized traces, this fact can be concluded as:

- Letter 2 commutes with any other letters (Propositions 3 and 4).
- Moreover, on the triangular tiling the steps indicated by 2 can be broken to two steps indicated by 1's (Proposition 2). Actually, it is a serialisation.

Theorem 1. *With the associative calculus $\mathcal{C} = (\{1, 2, 3\}, \{12 \rightleftharpoons 21, 23 \rightleftharpoons 32, 11 \rightleftharpoons 2\})$ $\mathcal{C}(w) = [[w]]$, i.e., \mathcal{C} provides the equivalence classes, the generalized traces, such that $w' \in \mathcal{C}(w)$ if and only if $D_c(w) = D_c(w')$.*

Proof. By Propositions 2–4, one can see that the rules of the system \mathcal{C} causes region-equivalent words. Moreover, if w is a prefix of a minimal equivalent neighborhood sequence, then all the words $w' \in \mathcal{C}(w)$ have this property (by Lemma 1).

On the other direction, by Proposition 5 and Lemma 1, it is clear that the parity of the sum of 1's (and 2's) before the first 3, between any two neighbor occurrences of 3 and after the last occurrence of 3 is fixed (by w) for every element w' of the generalized trace, the equivalence class $[[w]]$. However, by the calculus \mathcal{C} exactly those words w' can be obtained from w for which $|w'|_3 = |w|_3$ and the parities of the sums of 1's are the same in their respective positions. Hence, $[[w]] = \mathcal{C}(w)$. \square

We note here that the calculus gives the same equivalence classes without the rule $12 \rightleftharpoons 21$, since it can be simulated by two applications of $11 \rightleftharpoons 2$. However, we believe that \mathcal{C} is more intuitive in the presented way, and thus, the reader understands the idea behind better in this way.

Of course, there are longer sequences of steps that are equivalent to each other, e.g., 13222 is equivalent to 112123, but their equivalence is based on the use of shorter equivalences. The equivalence of every (consecutive) sequence of steps to another sequence is based on the listed short equivalences.

The system can be understood as a generalized trace [6] as we detail below. The system has some permutative (interleaving) rules, the rules by which a step to a 2-neighbor can be interchanged in the path with the previous or the next step (if it is not the same). However, the system has another types of production (representing serialisation), namely, $11 \rightleftharpoons 2$. This rule refers the change from

the unbroken 2 to two 1's. In this sense 2 represent an action in which the two 1's executed together.

In the next theorem we give a unique representation for each ball by specific regular languages.

Theorem 2. *Every digital ball $D_c(w)$ can be described by a word $w \in V^*$ such that exactly one of the following conditions holds:*

- | | |
|--------------------------|--------------------------|
| (a) $w \in 2^*$, | (b) $w \in 12^*$, |
| (c) $w \in 3(13)^*2^*$, | (d) $w \in 1(31)^+2^*$, |
| (e) $w \in (13)^+2^*$, | (f) $w \in (31)^+2^*$. |

Proof. Based on Theorem 1 using only minimal equivalent neighborhood sequences, one can easily show that any possible word $w \in V^*$ has exactly one equivalent word w' in exactly one of the listed cases. □

For all balls, their descriptions by words w fulfilling the condition of the above theorem are called *basic descriptions* or description by the *basic words*. In Fig. 3 example balls are shown for various cases. The basic words are among the shortest representations of the balls (in terms of $|w|$), the maximal (longest) representations are those words of the class that do not contain any letter 2.

4.3 The Number of Linearizations of Generalized Traces

In this subsection we count the number of words describing the same disc. We use a combinatorial approach. We distinguish six cases based on Theorem 2.

Let us start with the cases represented by words of shape 2^* and 12^* .

(a), (b): The case of balls described without 3's. In cases (a) and (b) each $w' \in [[w]]$ contains only 1's and 2's. One can also represent these balls by words of the regular language 1^* . The ball described by 1 is a triangle, and there is only one way to describe it. All other balls of these classes are hexagons (with sawtooth and straight sides, alternately, [13], see Fig. 3, first row).

Let us consider the ball described by word $w = 1^k$ (with $k \geq 1$) and let us count its possible descriptions.

- Using only 1's there is only 1 word (we may write this in the form $\binom{k}{0}$).
- Using exactly one letter 2 (in case of $k \geq 2$) we have $k - 2$ 1's and the length of these words is $k - 1$ and each permutation belongs to $[[w]]$. Their number is $\binom{k-1}{1}$
- ⋮
- Using i letter 2 (for $i \leq \frac{k}{2}$), the length of these words is $k - i$, and their number is $\binom{k-i}{i}$.

Summing up the number of these words, one gets $\sum_{i=0}^{\frac{k}{2}} \binom{k-i}{i} = F(k)$, i.e., it is the k -th Fibonacci number.

(c): The case of balls described by basic word of the form $w = \mathbf{3(13)^{n-1}2^m}$. In this case there are two parameters: $n > 0$ (gives the number of 3's) and $m \geq 0$. The shape is a hexagon with six straight sides if $n = 1$ (see Fig. 3, with $w = 322$); it is an enneagon with six hilly and three straight sides if $n > 1, m = 0$. In all other cases it is a dodecagon with six hilly and six straight sides [13]. By applying \mathcal{C} on w , we can establish the following fact. In each word $w' \in [[w]]$, before the first occurrence of 3 there could be an even number of 1's (maybe with some 2's also), after the last occurrence of letter 3 there could also be only even number of 1's (maybe with some 2's), and between any two occurrences of 3's there could be only odd number of 1's (maybe with some 2's). Let us apply \mathcal{C} on w by moving every 2 to their 'final' positions (with respect to the positions of 3's): then one may obtain a word $w'' = 2^{k_0}312^{k_1} \dots 32^{k_n}$ of the form $2^*3(12^*3)^*2^*$, where $\sum_{i=0}^n k_i = m$. Then by applying the rule $11 \Leftrightarrow 2$ (in any directions), the number and positions of 1's and 2's can be obtained in w' . The number of region equivalent words at each maximal subword not containing letter 3 is computed analogously to the number of words in the previous case. Thus, the cardinality of the set $[[w]]$ can be computed by the formula

$$\sum_{\substack{k_0, k_1, \dots, k_n \\ k_0 + \dots + k_n = m}} F(2k_0)F(2k_n) \prod_{i=1}^{n-1} F(2k_i + 1).$$

(d): The case of balls described by basic words $w = \mathbf{1(31)^n2^m}$. In this case there are two parameters again: $n > 0$ and $m \geq 0$. The shape of the ball is hexagon with six sawtooth sides if $n = 1$ (Fig. 3, $w = 131$); it is an enneagon with six hilly and three sawtooth sides if $n > 1, m = 0$; else it is a dodecagon with six hilly and six sawtooth sides [13].

In this case the number of 1's is odd in every maximal subword of w' not containing any letter 3. Consequently, the number of words $w' \in [[w]]$ is

$$\sum_{\substack{k_0, k_1, \dots, k_n \\ k_0 + \dots + k_n = m}} F(2k_0 + 1)F(2k_n + 1) \prod_{i=1}^{n-1} F(2k_i + 1) = \sum_{\substack{k_0, k_1, \dots, k_n \\ k_0 + \dots + k_n = m}} \prod_{i=0}^n F(2k_i + 1).$$

(e): The case of balls described by basic words $w = \mathbf{(13)^n2^m}$. The two parameters are $n > 0$ and $m \geq 0$. The shapes of the balls are similar to the shapes at case (c) with the given values of parameters. The number of words $w' \in [[w]]$ is given by

$$\sum_{\substack{k_0, k_1, \dots, k_n \\ k_0 + \dots + k_n = m}} F(2k_0 + 1)F(2k_n) \prod_{i=1}^{n-1} F(2k_i + 1) = \sum_{\substack{k_0, k_1, \dots, k_n \\ k_0 + \dots + k_n = m}} F(2k_n) \prod_{i=0}^{n-1} F(2k_i + 1).$$

(f): The case of balls described by basic words $w = \mathbf{(31)^n2^m}$, with parameters $n > 0$ and $m \geq 0$. The shapes of the balls are similar to case (d) with

the same parameter values, see, e.g., Fig. 3 $w = 31312$. The number of words $w' \in [[w]]$ is computed as

$$\sum_{\substack{k_0, k_1, \dots, k_n \\ k_0 + \dots + k_n = m}} F(2k_0) \prod_{i=1}^n F(2k_i + 1).$$

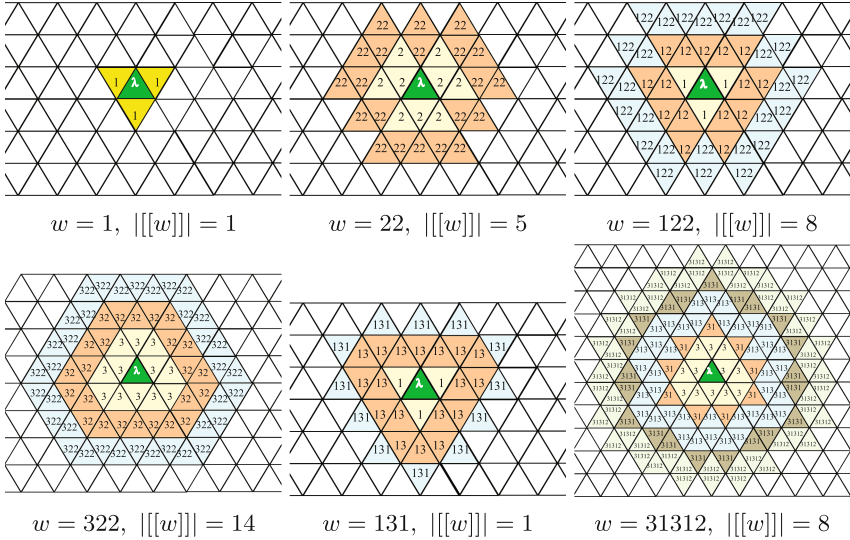


Fig. 3. Example balls with their basic words w and the cardinality of $[[w]]$.

5 Concluding Remarks

As we have seen instead of the binomial coefficients that are obtained on the square grid (lattice), more complex formulae are required to compute the number of words defining the same ball on the triangular tiling. Thus, these numbers can be considered, as a kind of generalizations of the binomial coefficients using the triangular tiling, and actually, they are the cardinalities of the sets representing generalized traces (defined also by special associative calculus) that describe digital balls on the triangular tiling. In this way an interesting relation between some parallel processes and shortest paths in the triangular tiling is established.

References

1. Das, P.P., Chakrabarti, P.P., Chatterji, B.N.: Distance functions in digital geometry. *Inf. Sci.* **42**, 113–136 (1987)
2. Deutsch, E.S.: Thinning algorithms on rectangular, hexagonal and triangular arrays. *Comm. ACM* **15**, 827–837 (1972)

3. Diekert, V., Rozenberg, G. (eds.): *The Book of Traces*. World Scientific, Singapore (1995)
4. Herendi, T., Nagy, B.: *Parallel Approach of Algorithms*. Typotex, Budapest (2014)
5. Farkas, J., Baják, S., Nagy, B.: Approximating the Euclidean circle in the square grid using neighbourhood sequences. *Pure Math. Appl.* **17**, 309–322 (2006)
6. Janicki, R., Kleijn, J., Koutny, M., Mikulski, L.: Generalising traces, CS-TR-1436. Technical report, Newcastle University, Partly Presented at LATA 2015 (2014)
7. Mateescu, A., Rozenberg, G., Salomaa, A.: Shuffle on trajectories: syntactic constraints. *Theoret. Comput. Sci.* **197**, 1–56 (1998)
8. Mazurkiewicz, A.W.: *Trace Theory: Advances in Petri Nets 1986*. LNCS, vol. 255, pp. 279–324. Springer, Heidelberg (1987)
9. Nagy, B.: Finding shortest path with neighborhood sequences in triangular grids. In: 2nd IEEE R8-EURASIP International Symposium ISPA, Pula, Croatia, pp. 55–60 (2001)
10. Nagy, B.: Metrics based on neighbourhood sequences in triangular grids. *Pure Math. Appl.* **13**, 259–274 (2002)
11. Nagy, B.: Shortest path in triangular grids with neighbourhood sequences. *J. Comput. Inf. Technol.* **11**, 111–122 (2003)
12. Nagy, B.: Distance functions based on neighbourhood sequences. *Publications Math. Debrecen* **63**, 483–493 (2003)
13. Nagy, B.: Characterization of digital circles in triangular grid. *Pattern Recogn. Lett.* **25**, 1231–1242 (2004)
14. Nagy, B.: Metric and non-metric distances on \mathbb{Z}^n by generalized neighbourhood sequences. In: 4th ISPA: International Symposium on Image and Signal Processing and Analysis, Zagreb, Croatia, pp. 215–220 (2005)
15. Nagy, B.: Distances with neighbourhood sequences in cubic and triangular grids. *Pattern Recogn. Lett.* **28**, 99–109 (2007)
16. Nagy, B.: Distance with generalized neighbourhood sequences in nD and ∞D . *Discrete Appl. Math.* **156**, 2344–2351 (2008)
17. Nagy, B., Strand, R.: Approximating Euclidean circles by neighbourhood sequences in a hexagonal grid. *Theoret. Comput. Sci.* **412**, 1364–1377 (2011)
18. Rosenfeld, A., Pfaltz, J.L.: Sequential operations in digital picture processing. *J. ACM* **13**, 471–494 (1966)
19. Rosenfeld, A., Pfaltz, J.L.: Distance functions on digital pictures. *Pattern Recogn.* **1**, 33–61 (1968)
20. Stojmenovic, I.: Honeycomb networks: topological properties and communication algorithms. *IEEE Trans. Parallel Distrib. Syst.* **8**, 1036–1042 (1997)
21. Yamashita, M., Honda, N.: Distance functions defined by variable neighborhood sequences. *Pattern Recogn.* **17**, 509–513 (1984)