Theoretical and algorithmic aspects of Discrete Geometry: inverse problems and tilings of the plane

Dissertation (monograph) approved by the
Doctoral Degree Committee of
Hamburg University of Technology

in pursuit of the academic degree of

Doktorin der Naturwissenschaften (Dr. rer. nat.)

written by
Michela Ascolese

from
Castellammare di Stabia (NA), Italy

2025

Reviewers: Prof. Dr. Anusch Taraz
Prof. Dr. Yan Gerard
Prof. Dr. Lajos Hajdu

Date of oral examination: November 29th, 2024

Contents

1 Introduction **3**

1.1 Reconstruction of binary and gray-scale images . . . 4

1.2 Reconstruction of uniform hypergraphs . . . 5

1.2.1 Pattern sequences . . . 7

1.2.2 Hypergraphs defined as ideals of a partially ordered set . . . 8

1.2.3 Randomized algorithms for the generation of hypergraphs . . . 9

1.2.4 Trees and co-degree sequences . . . 10

1.3 Exact polyominoes and homogeneous configurations . . . 11

1.4 Prime double squares . . . 13

2 Sets of uniqueness for the reconstruction of binary and gray-scale images **15**

2.1 Introduction and preliminaries . . . 15

2.2 Sets of uniqueness for binary images in a finite grid . . . 22

2.2.1 Simple cycles . . . 22

2.2.2 Enlarging regions . . . 24

2.2.3 Forbidden translations and uniqueness . . . 26

2.2.4 Sets of uniqueness for gray-scale images . . . 32

2.3 Theoretical aspects for a reconstruction strategy . . . 34

2.3.1 Geometric properties of binary and gray-scale solutions . . . 35

2.3.2 A rounding theorem . . . 38

2.4 Reconstruction of binary and gray-scale images . . . 40

2.4.1 Binary images . . . 43

2.4.2 Generalization of e-BRA to gray-scale images . . . 45

3 Reconstruction of uniform hypergraphs **52**

3.1 Introduction and preliminaries . . . 52

3.2 Pattern sequences of 3-uniform hypergraphs . . . 61

3.2.1 Patterns with fixed step . . . 61

3.2.2 Generic patterns . . . 67

3.3 Order Theory and ideals, an extension of the class $\mathcal{D}$ . . . 71

3.3.1 Principal ideals . . . 72

3.3.2 Ideals with two generators . . . 79

3.4 A randomized approach for the reconstruction of uniform hypergraphs . . 85
3.4.1 Balls and boxes . . . 86
3.4.2 Correctness and integer sequences that are k-graphic . . . 89
3.4.3 A variation that ignores higher degrees . . . 94
3.5 Co-degree sequences and trees . . . 96
3.5.1 Trivial instances and ambiguity . . . 96
3.5.2 Trees . . . 98
3.5.3 Reconstruction of trees from their co-degree sequence . . . 105
3.5.4 Generalization and open problems . . . 112

4 Homogeneous configurations in the discrete plane 116
4.1 Introduction and preliminaries . . . 116
4.2 Classification of exact polyominoes . . . 123
4.2.1 Pseudo-squares and pseudo-hexagons . . . 123
4.2.2 Periodicity . . . 126
4.3 The decomposition conjecture . . . 127
4.3.1 Implementation . . . 127
4.3.2 Exhaustive generation of perfect squares . . . 129
4.3.3 Counterexamples to the decomposition conjecture . . . 129
4.4 Non-decomposability . . . 134
4.4.1 Irreducible configurations . . . 135
4.4.2 Classes of non-decomposable tiles . . . 137

5 Prime double square polyominoes 143
5.1 Introduction and preliminaries . . . 143
5.2 Standard factorization of prime double squares . . . 148
5.2.1 Possible position of consecutive equal letters . . . 148
5.2.2 The standard factorization . . . 149
5.3 Proof of the couple free conjecture . . . 153
5.3.1 Couple free factors . . . 153
5.3.2 Main result and generalization . . . 159
5.4 Characterization of prime double squares . . . 161
5.4.1 The alphabet of factors . . . 161
5.4.2 Characterization of the factors of prime double squares in $\mathcal{P}^0$ . . . 173
5.5 Generation of $\mathcal{P}^0$ . . . 175
5.5.1 Class $\mathcal{A}$ . . . 175
5.5.2 Class $\mathcal{B}$ . . . 178
5.5.3 Class $\mathcal{C}$ . . . 179
5.6 Enumeration of the class $\mathcal{P}^0$ . . . 184

Chapter 1

Introduction

This thesis covers four topics within the wide area of Discrete Mathematics: for each of them, we investigate a different open problem and contribute to its solution. Even if they could appear different at first glance, the leitmotif that connects all of them is the definition of inverse problems concerning discrete structures, represented as subsets of points of $\mathbb{Z}^2$ or as matrices having integer or binary entries.

At first, we focus our attention on an inverse problem classically studied in the field of Discrete Tomography, that is, the reconstruction of an unknown (discrete) object starting from partial information on it. Usually, the interior of the object is supposed to be inaccessible, and the information is acquired by means of X-rays, mathematically modeled by discrete lines in $\mathbb{Z}^2$, and addressed as *projections*, while the object to be reconstructed is modeled as a set of points of the discrete plane, or, equivalently, as a binary image that is represented through a binary matrix. The main drawback in the reconstruction process is that in most cases the acquired information is not sufficient for the faithful determination of the unknown image, meaning that, given a set of discrete directions in $\mathbb{Z}^2$, there exist in general different objects having the same projections along such directions, and so the reconstruction problem is ill-posed. The problem we consider in this thesis is the determination of sets of directions satisfying the uniqueness property, meaning that, when we collect the projections along the directions of such sets, a binary image can always be reconstructed with no ambiguities.

We then move from the field of Discrete Tomography to the field of Graph Theory, where a similar inverse problem is defined: the reconstruction of a simple (hyper)graph starting from the knowledge of its degree sequence and of the cardinalities of its (hyper)edges. In the previous perspective, this is equivalent to the reconstruction of a binary matrix, specifically, the incidence matrix that is used to represent a hypergraph, starting from its horizontal and vertical projections, with the further constraint that all the rows of the matrix are required to be distinct. This last condition makes the reconstruction problem to be NP-hard, so that our research consists of the investigation of instances for which a solution can be easily and quickly detected.

We also consider the inverse problem of reconstructing a generic set of points in $\mathbb{Z}^2$ using a generalized notion of projection introduced by Maurice Nivat: instead of considering discrete lines, so *linear* projections, we collect the projections using bi-dimensional win-

dows. In this sense, we fix a finite connected set of positions in $\mathbb{Z}^2$, called *polyomino*, and we move it on the discrete plane taking note of the number of points that lie inside it, in order to detect the presence of the points of the set we want to reconstruct. In this case, a special role is played by exact polyominoes, i.e., those polyominoes that can be used to tile the discrete plane by translation.

In the last part of the thesis, we move apart from inverse problems and we focus our attention on exact polyominoes. We consider a special class of them, called *prime double squares*, and we carry on the study of a conjecture proposed by Blondin Massé et al. in 2013. We provide a solution for it, relying on tools from Combinatorics on words. We conclude by presenting some results that may constitute a step ahead the exhaustive generation and enumeration of the class of prime double square polyominoes.

In the following, we briefly detail the main results we reached in this work, giving a first sketch of the considered problems and of the approaches we used to study each of them.

1.1 Reconstruction of binary and gray-scale images

Tomography is a discipline that fits in the wide area of inverse problems, and whose main aim is the detection of an unknown object starting from a quantitative analysis of its primary constituents along a set of (discrete) directions. Generally speaking, the task is to reconstruct the density function, or image, of an unknown object starting from some information, as an example collected by means of X-rays. More in detail, given a set of directions, we count the number of points of the object that are intercepted by lines parallel to them. The so collected data are called *projections*, and the goal is to retrieve the object starting from them. Examples of practical applications range form medicine to crystallography.

While in some cases the unknown image can be assumed to have a continuous support, as well as the collected data, this model cannot be applied in general. This is the case when the object to be reconstructed has peculiar discrete characteristics, for example in case of crystalline structures, so few materials are present, or when only a very low number of X-rays data can be collected, for example due to the fragility of the analyzed body. In these cases we speak of Discrete Tomography, and more specifically of Binary Tomography when the image that we want to detect is binary, that is, when the object is homogeneous.

When moving from a continuous to a discrete model, the inverse problem becomes ill-posed, since few projections are collected: the image to be detected is modeled by a matrix whose entries can be, for example, only equal to 0 or 1 in the binary case. Such a restriction leads to the existence of many solutions starting from the same projections, and so to ambiguity in the reconstruction process. An important research line in the field focuses on the study of such ambiguities, and on the investigation of new techniques to overcome them and go back to uniqueness of reconstruction, as occurs in the continuous case when all the projections are available. As a first step, one can impose some conditions, supposed to be known a priori, on the object to be reconstructed. These constraints restrict the number of allowed solutions for a given set of projections, and in some cases guarantee the uniqueness, if properly chosen. This may occur under various conditions:

the number of different materials that constitute the object, that can be modeled choosing integer matrices where different entries correspond to different materials or, equivalently, to the gray levels of the discrete image to be reconstructed; the dimension of the image; some geometrical properties, as connectivity or convexity. From a practical point of view, these constraints are not so restrictive, since one can suppose to know, at least in part, the nature of the body being analyzed.
Other restrictions can be imposed on the directions that are chosen to collect the projections. Our studies follow this last research line.
In particular, we investigate special sets of directions, whose projections are unambiguous and so guarantee uniqueness of reconstruction. Focusing on binary images of given size, we move from the results presented in [26, 34, 35] and generalize them, finally defining special sets of directions that we call *simple cycles of uniqueness*. Starting from our theoretical results, we investigate a reconstruction strategy that leads to the definition of an algorithm that quickly reconstructs a binary image with no ambiguity.
We can summarize our results as

Theorem (see Theorem 2.2.11 and Corollary 2.3.12)**.** Given $\mathcal{A}$ a binary image of size $m \times n$, it is always possible to select a proper set of directions, S, such that uniqueness of reconstruction from the projections collected along S is guaranteed. Moreover, there exists an algorithm that explicitly and uniquely provides the reconstruction of $\mathcal{A}$ in polynomial time.

The information that we assume to know a priori is fundamental to select a proper set of directions S, and consists of the size of the unknown image and the number of its gray levels. A full description and characterization of cycles of uniqueness, as well as techniques for their practical detection, are detailed in Chapter 2, where we also describe the reconstruction algorithm together with experimental results that underline how our theoretical outcomes can be applied in practice. As a final step, we show how to generalize our results to higher dimensions, moving from $\mathbb{Z}^2$ to $\mathbb{Z}^n$, and give some preliminary results for their application to gray-scale images.

1.2 Reconstruction of uniform hypergraphs

Among the most interesting structures studied in Discrete Mathematics, particular attention is deserved to graphs. Their importance is underlined by the existence of a full branch of mathematics called Graph Theory, that dedicates its attention to their study both from a combinatorial and an algorithmic perspective. Graphs are a useful and versatile tool, that is used for the description of relations among objects, as well as for modeling complex networks. The most natural way to visualize the (binary) relationships modeled by a graph is to represent each object with a point, namely, a vertex, and each connection with a line between the two involved vertices, namely, an edge. In our work, we will consider one of the standard tabular representations of a graph, i.e., the incidence matrix. This matrix, $A = (a_{ij})$, describes the edges of the graph pointing out their presence among vertices: each column corresponds to a vertex, each row to an edge,

and the element in position a_{ij} is defined as follows:

$$a_{ij} = \begin{cases} -1 & \text{if the edge } e_i \text{ leaves the vertex } v_j, \\ +1 & \text{if the edge } e_i \text{ enters the vertex } v_j, \\ 0 & \text{otherwise.} \end{cases}$$

We only consider simple and undirected graphs, meaning that an edge always connects distinct vertices (no loops), all the edges are distinct (no repeated edges), and that the relations between vertices are symmetric. Under this hypothesis, the incidence matrix is binary, that is, $a_{ij} = 1$ if the i-th edge contains the j-th vertex, $a_{ij} = 0$ otherwise.
Moreover, we take one step forward and consider a generalization of graphs: hypergraphs. While in graphs edges show relationships between pairs of vertices, in hypergraphs relationships among more than two vertices are also taken into account. This generalization consists of replacing an edge with a hyperedge, that is a subset of vertices that can have any cardinality. As in case of graphs, also hypergraphs can be represented through an incidence matrix, as already described. In this case, an entry of the i-th row is equal to 1 if and only if the vertex of the corresponding column belongs to the i-th hyperedge. A further information can be retrieved from this matrix: its row sums describe the cardinality of each hyperedge, while the sum of the (non-null) elements of the j-th column corresponds to the degree of the j-th vertex, that is defined as the number of hyperedges in which the vertex appears. The list of all the degrees is addressed as the *degree sequence* of the hypergraph.
The connection between Binary Tomography and Graph Theory is now clear. Through the tabular representation given by its incidence matrix, any hypergraph can be seen as a binary image. On the other hand, if we consider the vertical and horizontal projections of a binary image, in case it is an incidence matrix, these are exactly the degrees of the vertices and the cardinalities of the hyperedges of the corresponding hypergraph, respectively. Then, the following, well known, reconstruction problem can be defined:

Problem: given two sequences of integers κ and π, reconstruct (the incidence matrix of) a hypergraph having π as degree sequence and κ as the sequence of the cardinialities of the hyperedges, that is, equivalently, the reconstruction of a binary image from the horizontal and vertical projections.

Even in this case, the problem is ill-posed, and the solution may be not unique in general. However, one of the solutions can be quickly reconstructed if the required structure is a graph ([38, 49, 50, 59]), or a hypergraph in which we allow repetitions of edges [58]. When removing these constraints, the problem becomes NP-hard [33], even if we try to move closer to the classic structure of graph by asking the hyperedges to have all the same cardinality k, that is the case of simple k-uniform hypergraphs. Motivated by this recent NP-hardness result, the studies carried on in this thesis aim at the detection of classes of degree sequences for which the reconstruction problem can be solved in polynomial time. As detailed in Chapter 3, we try different approaches, each of them leading to the characterization of some of these classes.
We start with pure deterministic and algorithmic strategies, mainly based on greedy approaches, that lead us to characterize and reconstruct the so called *pattern sequences*.

Their definition starts from a generalization of a type of degree sequence that is involved in the NP-completeness proof in [33]. They are characterized by the property of uniqueness, that is rare and allows to make the reconstruction problem no more ill-posed. Such a property can be considered as a hint for a possible reconstruction strategy that works in polynomial time. Then, we find a link between degree sequences of 3-uniform hypergraphs and Order Theory. In this case, we focus on those degree sequences that we can put in relation with the down-sets of a partially ordered set, since easier to manage. Finally, we introduce randomness, and we define a randomized algorithm that reconstructs with positive probability a further class of degree sequences. These sequences are characterized by constraints on their maximum entry, similar to the cases studied by the authors in [16].
We conclude the chapter by studying a variant of the reconstruction problem, that is defined on graphs instead of hypergraphs. We consider the new notion of sequence of the co-degrees of a graph, instead of the degrees, that describe the number of common neighbors between each pair of vertices. We provide some properties concerning the case when any pair of vertices in a graph share at most one neighbor, and then we consider the reconstruction problem of a graph from its co-degree sequence.

Problem: given an integer sequence γ of length $\binom{n}{2}$, for some $n \geq 2$, find a simple graph on n vertices having γ as its co-degree sequence.

We characterize those graphs that realize binary co-degree sequences, and then solve the reconstruction problem for the class of trees.
Hereafter we detail the above described research lines.

1.2.1 Pattern sequences

If we consider an n-tuple $s = (s(1), \ldots, s(n))$ of positive and negative integers, arranged in non-increasing order, it is possible to define a 3-uniform hypergraph H_s on n vertices $v_1, \ldots, v_n$ as follows: each vertex is associated to an element of s, and the triplet (v_i, v_j, v_k) is defined as a hyperedge of H_s if and only if $s(i) + s(j) + s(k) > 0$. If the sequence s is properly chosen, the elements of the degree sequence of H_s can show a characteristic behavior. An example is given by *pattern sequences*, defined as those degree sequences obtained when s is of type $s = s^{i,t} = (it, (i-1)t, (i-2)t, \ldots, 0, -1, -2, \ldots, -(2i-1)t+1)$, with $t \geq 1$ a fixed integer, and varying the value $i \geq 1$. We point out them as $\pi^{i,t}$, referring to the parameters i and t characterizing the sequences s that define them.

Example 1.2.1. If we fix $t = 2$ and $i = 4$, the sequence $s^{4,2} = (8, 6, 4, 2, 0, -1, -2, \ldots, -13)$ generates the pattern sequence

$$\pi_1 = (55, 44, 35, 28, 22, 19, 16, 14, 12, 10, 7, \mathbf{6}, \mathbf{4}, \mathbf{4}, \mathbf{2}, \mathbf{2}, \mathbf{1}, \mathbf{1}).$$

On the other hand, if $t = 3$ and $i = 4$, the sequence $s^{4,3} = (12, 9, 6, 3, 0, -1, -2, \ldots, -20)$ generates the pattern sequence

$$\pi_2 = (93, 71, 54, 41, 32, 28, 25, 22, 19, 17, 15, 13, 11, 8, 7, \mathbf{6}, \mathbf{4}, \mathbf{4}, \mathbf{4}, \mathbf{2}, \mathbf{2}, \mathbf{2}, \mathbf{1}, \mathbf{1}, \mathbf{1}).$$

In the previous example, we put in boldface the lowest degrees of π_1 and π_2, that we address as *tail*, and whose behavior characterizes pattern sequences. Indeed, once a value t is fixed, when increasing the value of the parameter i, we observe that the last elements of the degree sequences can be grouped in $i-1$ sets of cardinality t, whose entries are given by the numerical sequence $\{a_n\}_{n\geq 1}$ whose generic element is $a_n = \lceil \frac{n+1}{2} \rceil \cdot \lfloor \frac{n+1}{2} \rfloor$. More in general, we obtain the following result:

Theorem. (see Theorem 3.2.9) For a fixed step $t \geq 1$ and index $i \geq 1$, the tail related to the i-th pattern sequence is

$$T(i) = (a_i, a_{i-1}^t, \ldots, a_2^t, a_1^t),$$

with $\{a_n\}_{n\geq 1}$ the numerical sequence whose generic element is

$$a_n = \left\lceil \frac{n+1}{2} \right\rceil \cdot \left\lfloor \frac{n+1}{2} \right\rfloor \qquad \text{for } n \geq 1.$$

In Chapter 3, we further detail this result and show that the tail of a pattern sequence actually identifies it uniquely, and that it is possible to retrieve from it the integer sequence $s^{i,t}$ that generates the corresponding hypergraph. As a consequence, we can define an algorithm for the quick reconstruction of the 3-uniform hypergraphs associated to these type of degree sequences.

After this result, we generalize and define the wider class of degree sequences $\mathcal{P}$, that are obtained similarly, starting from a sequence s of type $(s', 0, -1, \ldots, -q)$, where s' consists of non-negative integers arranged in non-increasing order and $-q = -s'(1) - s'(2) + 1$. In this more general case, the notion of tail, as well as the nice regularity of its elements, is lost. However, a polynomial time algorithm for the reconstruction of the hypergraphs associated to the sequences in this class is provided.
We can summarize our results as

Theorem (see Theorems 3.2.10 and 3.2.16)**.** If $\pi \in \mathcal{P}$, then the 3-hypergraph having π as its degree sequence is unique, and can be reconstructed in polynomial time.

The complete study of the sequences in $\mathcal{P}$, and the algorithms that solve the associated reconstruction problem, in polynomial time, are described in detail in Chapter 3, Section 3.2.

1.2.2 Hypergraphs defined as ideals of a partially ordered set

The definition of pattern sequences is given starting from an integer sequence s and defining a 3-uniform hypergraph H_s according to the rule: $(v_i, v_j, v_k) \in H_s$ if and only if $s(i) + s(j) + s(k) > 0$. From this construction, a trivial property follows:

Property 1.2.2. Given π_s the degree sequence obtained from $s = (s(1), \ldots, s(n))$ and H_s the related hypergraph, if (v_i, v_j, v_k) is an edge of H_s, then $(v_i, v_j, v_{k'})$ is an edge of H_s, for all $j + 1 \leq k' \leq k$.

This property emphasizes a descending structure in the hypergraphs thus constructed, that suggests a relation with the ideals of a partially ordered set. Starting from this idea, we introduce a new class of degree sequences, as well as a quick reconstruction strategy for some of them.

For a fixed integer $n \geq 3$, we define the set of triplets $\Omega_n = \{(a_1, a_2, a_3) \text{ s.t. } 1 \leq a_1 < a_2 < a_3 \leq n\}$ and the following order relation on them,

$$(a_1, a_2, a_3) \preceq (b_1, b_2, b_3) \text{ if and only if } a_i \leq b_i, \text{ with } i \in \{1, 2, 3\}.$$

We address the pair $\mathcal{T}_n = (\Omega_n, \preceq)$ as the partially ordered set of triplets, and consider the set $\mathcal{I}_n$ of its ideals. Each triplet (a_1, a_2, a_3) naturally corresponds to the hyperedge $(v_{a_1}, v_{a_2}, v_{a_3})$ in the complete 3-uniform hypergraph on n vertices. As a consequence, the elements in $\mathcal{I}_n$ are in bijection with a class of 3-uniform hypergraphs whose hyperedges have a descending structure induced by the ideals.
We dedicate Section 3.3 of Chapter 3 to the study of the degree sequences of the hypergraphs in bijection with $\mathcal{I}_n$, specifically, to the case of ideals with one or two generators. We briefly summarize our main results in the following theorems.

Theorem (see Theorem 3.3.5)**.** Let $g = (a, b, n) \in \Omega_n$. The i-th entry of the degree sequence of the hypergraph associated to the principal ideal $\downarrow\{g\}$ is

$$d_i = \begin{cases} \frac{(b-1)(2n-b-2)}{2} & \text{if } 1 \leq i \leq a, \\ \frac{a(2n-a-3)}{2} & \text{if } a+1 \leq i \leq b, \\ \frac{a(2b-a-1)}{2} & \text{if } b+1 \leq i \leq n. \end{cases}$$

Theorem. (see Theorems 3.3.12 and 3.3.14) Given an integer sequence $\pi = (d_1, \ldots, d_n)$, if there exists a principal ideal $\downarrow\{g\} \in \mathcal{I}_n$ realizing π, then π is unique. Moreover, the unique 3-uniform hypergraph having π as degree sequence can be reconstructed in polynomial time.

The degree sequences realized by the ideals of $\mathcal{T}_n$ with two generators have been also characterized (Theorem 3.3.17), and we implemented a quick algorithm that reconstructs the corresponding hypergraph in the most cases. Since a case analysis occurs, we decide to postpone their description to the corresponding section of Chapter 3.

1.2.3 Randomized algorithms for the generation of hypergraphs

Through the definition of deterministic algorithms, we are able to solve the reconstruction problem of uniform hypergraphs from degree sequence in polynomial time only for few specific classes of 3-uniform hypergraphs, all corresponding to some down-set of the poset $\mathcal{T}_n$. To enlarge the set of solvable instances, we relax such a rigorous structure of the incidence matrix, and also consider a general uniformity value $k \geq 3$. Since we still require a simple hypergraph as a solution, the problem continues to have an NP-hard complexity, so the goal is again the investigation of classes of instances that are solvable in polynomial time. We decide to try a different approach, introducing randomness. This new perspective leads to the definition of randomized algorithms, that give rise to some

conditions that, if satisfied by a given integer sequence, immediately reveal the existence of (at least one) k-uniform hypergraph having the input sequence as its degrees. In particular, the required constraints concern the highest entry of the integer sequence, i.e., the highest degree of the related hypergraph, that we bound in terms of the uniformity value, k, and of the number of edges, σ.
In other words, we make use of the *probabilistic method* to show the existence of a structure satisfying prescribed constraints, in this case a simple k-uniform hypergraph with a given degree sequence, thus characterizing a class of k-graphic sequences. Moreover, the randomized algorithm we define is also constructive, meaning that, if it exists, it provides a solution to the reconstruction problem with a positive probability, that tends to one as the number of vertices of the hypergraph tends to infinity. We recall here our main result:

Theorem (see Theorem 3.4.6 and Remark 3.4.7)**.** For $n \in \mathbb{N}$ and $k \geq 3$, let $\pi = (d_1, \ldots, d_n)$ be a non-increasing integer sequence such that $k(k+1)d_{k+2} \leq \sigma$, with $\sigma = \sum_{i=1}^{n} d_i$. Then, there is a polynomial time randomized algorithm that always returns a k-hypergraph H with degree sequence π, and satisfies

$$\mathbf{P}(H \text{ is simple}) \geq 1 - \frac{k+1}{2}\left(\frac{3k}{2}\right)^{k-2} \frac{d_1^k}{\sigma^{k-2}}.$$

Therefore, $\mathbf{P}(H \text{ is simple}) \overset{n\to\infty}{\longrightarrow} 1$ holds if $d_1^k = o(\sigma^{k-2})$.

From a more practical point of view, we find that if the degrees are either sufficiently small or close to each other, then a simple hypergraph realizing the required degree sequence exists.

Corollary (see Corollaries 3.4.8 and 3.4.9)**.** Let $k \geq 3$ and define $\rho := \frac{d_1}{d_n}$. If

$$d_1^2 \left(\frac{\rho}{n}\right)^{k-2} \to 0,$$

or

$$d_1 \leq Cn^{\alpha} \text{ for some } C > 0 \text{ and } \alpha < 1 - \frac{2}{k},$$

then $\mathbf{P}(H \text{ is simple}) \overset{n\to\infty}{\longrightarrow} 1$ holds.

The detailed description of our approach, and the definition of the mentioned randomized algorithm, can be found in Section 3.4 of Chapter 3.

1.2.4 Trees and co-degree sequences

The last part of Chapter 3 is dedicated to a variant of the reconstruction problem that brings us back to the classic structure of graph. We generalize the concept of degree of a vertex and consider the co-degrees, that describe the number of common neighbors between pairs of vertices. The reconstruction problem is defined exactly the same way.

Problem: given a sequence of integers, find a simple graph having those values as its co-degrees.

Up to our knowledge, the problem has not been studied yet. As a starting point, we focus on the easier instances of binary sequences. We characterize all those graphs realizing this type of co-degree sequences (see Theorem 3.5.15), and solve the reconstruction problem for the class of trees, a special case of graphs having binary co-degree sequence.
We first define the so called *canonical trees*, that is a special structure that we can give to a tree through the iterative application of an operation that does not alter its co-degree sequence. Then, using an algorithm that makes use of dynamic programming, we provide a solution to the reconstruction problem in polynomial time, constructing for a given integer sequence γ, if it exists, a canonical tree having γ as its co-degree sequence.
We postpone the definition of canonical trees to Section 3.5, since it is reached constructively by performing simple operations on a given tree T. Our results can be summarized as follows:

Theorem (see Theorem 3.5.13)**.** Given a binary sequence γ of length $\binom{n}{2}$, for some $n \geq 2$, it is possible to establish in polynomial time if there exists a tree T on n nodes having γ as the sequence of its co-degrees. Moreover, it is possible to reconstruct a solution having a canonical structure in polynomial time.

As detailed in Section 3.5, not all binary co-degree sequences can be realized by trees. We will provide some examples of such a situation, and present some preliminary results that we achieved during our research about the general case.

1.3 Exact polyominoes and homogeneous configurations

So far, we made use of the notion of linear projection for the analysis and inspection of unknown objects, equivalently modeled as binary images or binary matrices.
In 2005, Maurice Nivat introduced the notion of *rectangular scan*, that is a new concept of bi-dimensional projection, generalizing the linear projection classically employed in tomography. The idea is to replace X-rays with rectangles, and then use them as a probe to inspect the discrete plane $\mathbb{Z}^2$ and its binary configurations. A *binary configuration* is defined as a subset A of points of $\mathbb{Z}^2$, where the symbol 1 points out the presence of a point of A, while 0 its absence. Instead of counting the number of points in A intercepted by a straight line, we move a rectangle W over the discrete plane, and then count, for each position, the number of 1s that are revealed by the probe. The configuration A is defined *h-homogeneous* with respect to the rectangle W if, for each possible position of W in $\mathbb{Z}^2$, the number of elements that occur into the rectangle is always equal to h.
Rectangles and homogeneous configurations have been studied in depth, as well as their properties, in [56]. Among the others, an important result is the following decomposition theorem:

Theorem (Nivat [56])**.** If a configuration A is h-homogeneous with respect to a rectangle W, then it is possible to decompose A in h disjoint configurations such that they are all 1-homogeneous with respect to W.

Example 1.3.1. If we consider the rectangle of size 4×3, three possible homogeneous configurations are depicted in Figure 1.1(a), (b) and (c). The decomposition of the 2-homogeneous one is also provided, in (e) and (f).

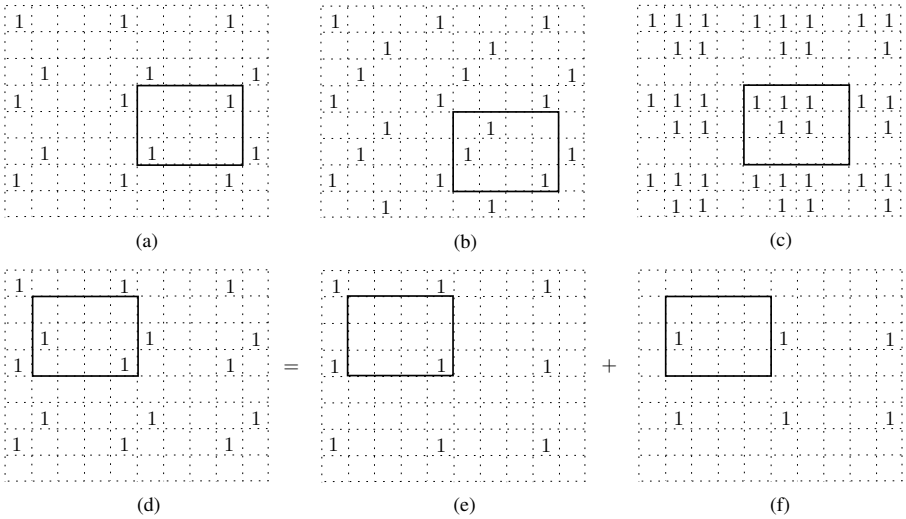


Figure 1.1: On the top, from left to right, three homogeneous configurations w.r.t. the rectangle of size 4×3, with homogeneity values $h = 2, 3$ and 5, respectively. On the bottom, the decomposition of the 2-homogeneous configuration in two disjoint 1-homogeneous ones.

Later, the concept of bi-dimensional projection was further generalized, considering as a probe not only rectangles, but any polyomino, namely, a finite set of 4-connected points in $\mathbb{Z}^2$. Since dealing with the discrete plane, exact polyominoes acquired significant relevance in its inspection. These polyominoes, also called *tiles*, are defined as those ones that can be used to create a tessellation of $\mathbb{Z}^2$, meaning that it is possible to cover the whole discrete plane using infinitely many copies of the same tile placed side by side, as shown in Figure 1.2. A rigorous definition of exact polyominoes, together with their classification in classes, is given in Chapter 4.

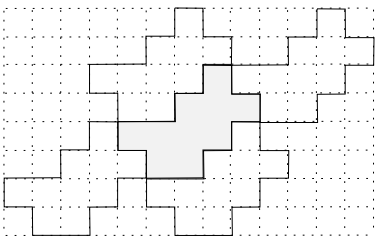


Figure 1.2: An exact polyomino can always be surrounded with copies of itself, and then used to cover the whole plane $\mathbb{Z}^2$ by translation.

As in case of rectangles, if we consider an exact polyomino W and a configuration A, it is possible to inspect A using W as a probe, and similarly to extend the definition

of h-homogeneous configuration to any tile. More in detail, exact polyominoes become relevant when dealing with homogeneous configurations due to the following result:

Theorem (Nivat [56]). Given a polyomino W, there exists a 1-homogeneous configuration w.r.t. W if and only if the polyomino is exact. Moreover, such a configuration is periodic w.r.t. the directions of periodicity of W.

Since exact polyominoes characterize homogeneous configurations, in some sense, the research line in the field moved in the direction of tiles, and in 2005 Frosini and Nivat conjectured that the decomposition theorem, that holds for rectangles, could be extended to any exact tile [42]. In fact, these are the only ones for which 1-homogeneous configurations exist, and so for which a decomposition result could be achieved.
Chapter 4 of this thesis is dedicated to the study of such a conjecture. We start with an overview of the classes in which exact polyominoes can be classified, and then present an algorithm that is used for the exhaustive generation of them and of the corresponding homogeneous configurations. Using our algorithm, we are able to provide a counterexample to the conjecture, from which we get the following result:

Theorem (see Section 4.3.3). The decomposition theorem, proven for rectangles in [56], does not hold for a general exact tile.

This result opens a new research line in the field, mainly concerning the possibility of a new classification of tiles, that could be divided in classes looking at the homogeneity values for which the decomposition theorem holds or not. Some examples of these classes are characterized and described at the end of Chapter 4.
We furthermore present a new conjecture, stating that the decomposition theorem holds when the considered tile W is exact and its area is a prime number.

1.4 Prime double squares

Combinatorics is a huge area of Discrete Mathematics mainly dealing with counting, but that finds many applications in the most different fields, from pure mathematics to the analysis of algorithms. The branch of Combinatorics we focus on is Combinatorics on words, from which we borrow basic tools to describe, generate, and finally enumerate a class of discrete objects, thus resulting in the most classical area of Enumerative Combinatorics.

Moving from the classification of exact tiles analyzed in Chapter 4, a specific class of them arouse our curiosity: double squares. The peculiarity of these polyominoes is that each of them can be used to tile the discrete plane by translation in two different ways, that is the reason why many studies were carried on to mark out their characteristics ([19, 20, 21]). An example of double square is given in Fig. 1.3.
Combinatorics on words plays a special role in these studies, since polyominoes are coded and uniquely identified through their boundary word. With the aim of reaching their exhaustive generation, the authors in [21] defined the class of *prime double squares*, and marked out many properties that are typical of their boundary words. In the same paper, they stated the following

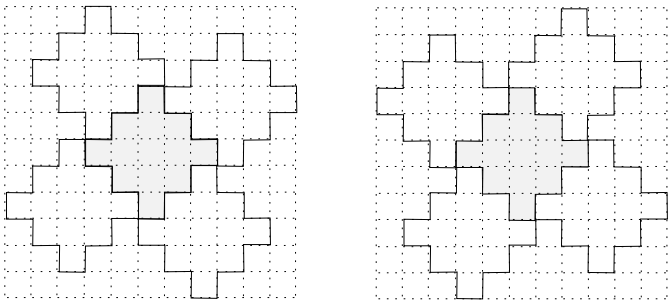


Figure 1.3: A double square tile, together with its two different tilings of $\mathbb{Z}^2$.

Conjecture: two consecutive equal letters are never present in the boundary word that identifies a prime double square.

Starting from this conjecture, in Chapter 5 we carry on the study of the properties of prime double squares, specifically of their boundary words. We provide a proof to the conjecture, that leads to a complete characterization of the boundary words of these special tiles. This important result automatically leads to the generation with no repetitions of a subclass of prime double squares, and finally to their enumeration with respect to the semi-perimeter. We can summarize the main results of the chapter as

Theorem (see Theorem 5.3.13)**.** Let P be the boundary word of a polyomino, defined on an alphabet Σ, and let be $\alpha \in \Sigma$. If P identifies a prime double square, then the factor $\alpha\alpha$ does not occur in the word P.

Moreover, a complete characterization of a subclass of prime double squares is provided in Section 5.4, as well as their exhaustive generation with no repetitions, Section 5.5, and their enumeration with respect to the semi-perimeter, Section 5.6.

Chapter 2

Sets of uniqueness for the reconstruction of binary and gray-scale images

In this chapter, we consider a discrete approach to the problem of the reconstruction of images of given size starting from valid sets of directions. We face the problem of *ambiguity*, very puzzling and relevant in the field of inverse problems, and develop some techniques for the construction of sets of directions that guarantee uniqueness. We first focus on binary images, and then move to the general case of more gray levels. In addition to important theoretical results, we also provide a practical reconstruction strategy through the implementation of an algorithm, whose efficiency is supported by the exhibition of experimental results.
The results described in this chapter are joint work with P. Dulio and S.M.C. Pagani in [2] and [3].

2.1 Introduction and preliminaries

In this section, we give the motivations of our work together with the notations and the mathematical tools we are going to use.

General framework

Tomography is the branch of inverse problems dealing with the reconstruction of unknown objects using data acquired by means of X-ray radiation. The unknown object is represented by a density function $f : \mathbb{R}^3 \to \mathbb{R}$, whose support consists of the points of the body we are studying, and that assumes different values depending on the different materials that constitute the object. As a first simplification, we can assume to be in the 2D case, since a 3D reconstruction can always be achieved through the reconstruction of a large number of suitable 2D slices. The collection of data is made by means of parallel X-rays, that, starting from sources placed at a certain distance, hit the object with prescribed slopes, or angles, so reducing their intensity according to the different

densities of the body. The attenuation of the X-rays' intensity is measured by means of detectors and collected as a *projection*. From a theoretical point of view, if we collect projections along all the possible angles in the interval $[0, \pi)$, then the density function of the object can be easily and univocally computed using the analytic inversion of the *Radon transform* [57]. We do not go into details, since the study of the continuous case is beyond the scope of this thesis. Anyway, it is clear that the collection of data along all possible directions is impossible in practice, and that some discretization is required. As a matter of fact, the image is modeled as a set of *pixels*, whose size is determined by the quality of the acquisition system, and reflects in the image resolution. Consequently, in each pixel the density function f is constant. In this way, each point encodes the density of the corresponding pixel, so a density function $f : \mathcal{A} \subset \mathbb{Z}^2 \to \mathbb{R}$ is obtained, where $\mathcal{A}$ is a finite subset of weighted points, in fact representing the object itself. In other words, the image can be modeled as a matrix of size $m \times n$, whose entries are real values that represent the densities of the materials corresponding to the different pixels. As part of the discretization process, the number l of these materials is also few in general, and reduces to $l = 2$ if the unknown body is assumed to be homogeneous. In this case, we speak of Binary Tomography, and the detection of the presence or absence of material in the points of $\mathcal{A}$ reflects in the reconstruction of an $m \times n$ *binary* image. We call $N := mn$ the number of pixels of the image and $||f|| = \max_{(\xi,\eta)\in\mathcal{A}}\{|f(\xi, \eta)|\}$ the norm of the density function. We mainly work in the binary framework, but some results about the more general case $l > 2$ will also be given.

Hajdu and Tijdeman [48] observed that the set of binary solutions is precisely the set of shortest vector solutions in the set of functions $f : \mathcal{A} \subset \mathbb{Z}^2 \to \mathbb{Z}$ with the given line sums, provided that such solutions exist. This inspires us to focus on integer valued density functions, namely, we assume that the line sums are integers. It is not obvious that integer solutions exist for a given tomographic problem. However, this is guaranteed by the following result of Stolk (Corollary 3.2.10 in [60], see also Lemma 2.3 in [32]):

Theorem 2.1.1 (Stolk [60])**.** Let $\mathcal{A}, S$ be given. Suppose all the line sums in the directions of S are integers, and there exists a real function $g : \mathcal{A} \to \mathbb{R}$ satisfying the line sums. Then, there exists a function $f : \mathcal{A} \to \mathbb{Z}$ satisfying the line sums.

In particular, when dealing with an image on l gray levels, we always mean a function $f : \mathcal{A} \to \{0, 1, \ldots, l - 1\}$, with $l = 2$ in the binary case.

We also work under the hypothesis of perfect acquisition of the data, meaning that no noise is present in our modeling.

Modeling the tomographic problem: the grid model

As already mentioned, the modeling of the tomographic problem goes through a discretization process, that can take place at more levels, so leading to different models. For our research, we make use of the *grid model*.

> *Discretization of the image.* The pixels that constitute the grid $\mathcal{A}$, as well as the object to be reconstructed, have a finite size in general, given by the fact that the

acquisition of data is made through a finite number of detectors. So, the density function f can be correctly reconstructed up to the size of the pixels. Since we do not have any information inside pixels, each of them can be collapsed in a single point of the discrete lattice $\mathbb{Z}^2$, so that the reconstruction problem is studied in such a discrete space.

Discretization of X-rays. X-rays are discretized as well, and become part of $\mathbb{Z}^2$ as discrete lattice lines, in other words, sets of points of $\mathbb{Z}^2$ belonging to the same continuous line (see the formal definition below).

Discretization of the density function. The object is supposed to be constituted of l different materials, so that the density function is $f : \mathcal{A} \subset \mathbb{Z}^2 \to \{0, 1, \ldots, l-1\}$. In case of homogeneous object, $l = 2$ and we speak of Binary Tomography.

The grid model finally leads to the formulation of the reconstruction problem as the detection of the solution of a linear system.

A *lattice direction* $u = (a, b)$ is defined as a pair of co-prime integers, with $a = 1$ if $b = 0$ and $b = 1$ if $a = 0$ (horizontal and vertical direction, respectively). Without loss of generality, we can always assume $a \geq 0$. We denote $S = \{u_1, \ldots, u_d\}$ a set of directions, with $h = \sum_{i=1}^{d} a_i$ and $k = \sum_{i=1}^{d} |b_i|$. A line of direction (a, b) is given by the equation $ay = bx + t$, $t \in \mathbb{Z}$, and, given a function $f : \mathcal{A} \subset \mathbb{Z}^2 \to \mathbb{Z}$, the *projection* along the previous lattice line is $\sum_{a\eta = b\xi + t} f(\xi, \eta)$. In other words, if the object is homogeneous, meaning that f only assumes value 0 or 1, the projection counts the number of points of the object that are intercepted by X-rays taken along the direction (a, b). Once a set of directions S is chosen, we can take all the measurements along S and collect them in the vector of projections $\mathbf{p} \in \mathbb{N}^s$. Now, the reconstruction problem consists in solving the linear system

$$\mathbf{A}\mathbf{x} = \mathbf{p},$$

where $\mathbf{x} \in \{0, 1\}^N$ is a vector of N unknowns, each of them representing a pixel of the image to be detected, and $\mathbf{A}$ is the projection matrix. According to the grid model, $\mathbf{A}$ is binary as well, since the generic entry a_{ij} is equal to 1 if the j-th pixel is intercepted by the i-th X-ray, and equal to 0 otherwise.

We finally recall some definitions of norm. For a vector $\mathbf{x} \in \mathbb{R}^N$, we denote $||\mathbf{x}||_2 = \left(\sum_{i=1}^{N} x_i^2\right)^{\frac{1}{2}}$ its Euclidean norm, and $||\mathbf{x}||_1 = \sum_{i=1}^{N} |x_i|$. For a matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, the Frobenius norm is $||\mathbf{A}||_F = \left(\sum_{i=1}^{M} \sum_{j=1}^{N} |a_{ij}|^2\right)^{\frac{1}{2}}$.

Example 2.1.2. In Figure 2.1 is depicted a set of lattice points that represent a homogeneous object in the grid model. Black points stand for the presence of the body, white points for its absence. To compute the projections along a given direction, we need to count the number of black points that are intercepted by each line. Choosing the directions $u_1 = (1, 0)$, $u_2 = (0, 1)$ and $u_3 = (2, 1)$, we scan the object from left to right for what concerns u_1, from the bottom to the top for u_2, and from the bottom-left corner along u_3. The obtained vector of projections is

$$\mathbf{p} = [5, 5, 5, 7, 5, 4, 4; 4, 4, 6, 3, 5, 2, 2, 5, 4; 5, 3, 3, 0, 2, 1, 1, 1, 0, 2, 3, 1, 2, 3, 2, 1, 1, 2, 1, 1, 0, 0].$$

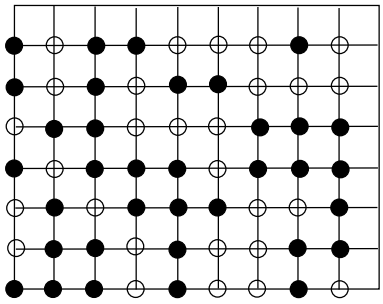


Figure 2.1: The discretization of a homogeneous object consisting of 9×7 pixels. According to the grid model, each pixel collapses in a point of the integer lattice $\mathbb{Z}^2$. In this case, the coordinates are given by the bottom-left corner of the square representing the pixel.

Ghosts in Discrete Tomography

Ghosts as solutions of homogeneous linear systems. We recall that one of the constraints that we have to face when we work in Discrete Tomography is the few number of projections, so in general the linear system that we have to solve is strongly underdetermined. Fixing a set of directions S, if we consider the associated homogeneous system $\mathbf{Ax} = \mathbf{0}$ we can find many non-trivial integer solutions in general, meaning that there exist many non-trivial density functions having zero line sums along all the directions in S. These objects are called *S-ghosts*, and are the reason of ambiguity in the reconstruction process. More specifically, we are interested in the study of *binary ghosts*, so ghosts whose density function assumes value in the set $\{0, \pm 1\}$ only, since these are those ones that can lead to the existence of different binary solutions (see Example 2.1.3).

Example 2.1.3. Let us consider an image of size 3×3 and the set of directions $S = \{(1,0),(0,1)\}$. After the acquisition of data, we get the vector of projections

$$\mathbf{p} = [2,2,2;1,3,2]$$

and set up the following linear system to solve the tomographic problem:

$$\begin{cases} x_1 + x_2 + x_3 = 2 \\ x_4 + x_5 + x_6 = 2 \\ x_7 + x_8 + x_9 = 2 \\ x_1 + x_4 + x_7 = 1 \\ x_2 + x_5 + x_8 = 3 \\ x_3 + x_6 + x_9 = 2 \end{cases}$$

where the pixels $x_1, \ldots, x_9$ are encoded row-by-row, starting from the top-left corner. We now consider the associated homogeneous system $\mathbf{Ax} = \mathbf{0}$. There exist (at least) two different non-trivial solutions,

$$\mathbf{g_1} = [-1,0,+1,0,0,0,+1,0,-1],$$
$$\mathbf{g_2} = [+1,0,-1,0,0,0,-1,0,+1],$$

both assuming values in the set $\{0, \pm 1\}$. In other words, $\mathbf{g_1}$ and $\mathbf{g_2}$ are two binary S-ghosts, that give rise to ambiguity in the resolution of the tomographic problem we set. As an example, the following binary matrices have both projections equal to $\mathbf{p}$ along the horizontal and vertical directions, but represent two different (binary) images, namely, two different solutions:

$$\mathbf{x_1} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \qquad \mathbf{x_2} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

Notice that $\mathbf{x_2} = \mathbf{x_1} + \mathbf{g_2}$.

Geometric description of ghosts. Ghosts can be associated to sets of weighted lattice points, having zero line sums along prescribed directions. Given a set of directions S, we call a *weakly bad configuration with respect to* S a pair of sets (Z, W) consisting of $2k$ lattice points, $z_1, \ldots, z_k \in Z$ and $w_1, \ldots, w_k \in W$, not necessarily distinct, and counted with their multiplicity, such that for each direction $u = (a, b) \in S$, a line with direction u meets the same number of points in Z and W. In other words, if we consider the configuration (Z, W) as an image, and we give a weight to the points according to their multiplicity, considering a negative weight for the points in Z and a positive weight for those ones in W, then a weakly bad configuration w.r.t. S corresponds to an S-ghost. If all the points in Z and W are distinct, we speak of *bad configuration*. Since in this case all the lattice points have weight equal to 0 or ± 1, the associated ghost is binary.

Example 2.1.4. The set B of black points in Fig. 2.1, and the set BR consisting of black and red points in Fig. 2.2, represent homogeneous objects having the same projections along the directions $u_1 = (1, 0)$, $u_2 = (0, 1)$ and $u_3 = (2, 1)$.
This implies that the set $G = B \setminus BR$ is a binary ghost. It consists of the circled and filled red points in Fig. 2.2, corresponding, respectively, to the lists Z and W composing the bad configuration.

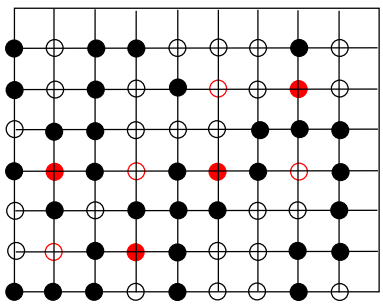


Figure 2.2: A different solution to the tomographic problem presented in Example 2.1.2.

An example of weakly bad configuration can be found in Fig. 2.4(a), where a point of weight two is present.

Ghosts are not easy to avoid in general, as highlighted by the following result.

Theorem 2.1.5 (Hajdu and Tijdeman [48]). For any $d \in \mathbb{N}$, and for any set of directions $S = \{u_1, \ldots, u_d\}$, there exists a ghost with respect to S.

To show this, it suffices to start with two points of opposite weight, z_1 and w_1, lying on a line along the direction u_1. Then, we consider a copy of these two points along the direction u_2, but changing the sign of the weights. Iterating this construction for d times leads to the desired configuration. Example 2.1.6 shows the iterative construction of a bad configuration for the directions in $S = \{(1,0),(0,1),(2,3),(4,5)\}$.

Example 2.1.6. We provide an example of the iterative procedure that leads to the construction of a binary ghost along the set of directions $S = \{(1,0),(0,1),(2,3),(4,5)\}$. First of all, a bad configuration along $u_1 = (1,0)$ is constructed, Fig. 2.3(a). Then, such a configuration is translated along the direction $u_2 = (0,1)$, changing the sign of the weights of the involved lattice points, Fig. 2.3(b). The same is repeated along the direction $u_3 = (2,3)$, Fig. 2.3(c), and then along $u_4 = (4,5)$, finally reaching the S-binary ghost depicted in Fig. 2.3(d).

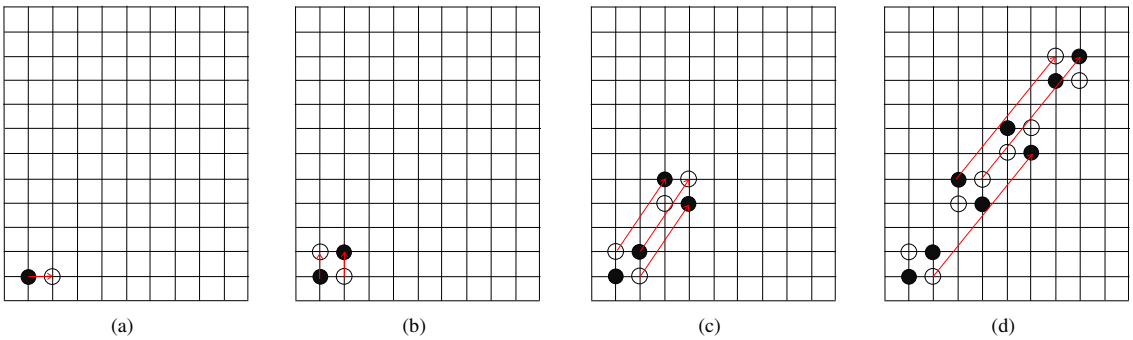


Figure 2.3: The iterative construction of a (weakly) bad configuration along a prescribed set of directions.

The previous construction is always possible, provided that we can move in the whole lattice $\mathbb{Z}^2$. Differently, one can confine the tomographic problem in a grid $\mathcal{A}$ of size $m \times n$. In view of uniqueness of reconstruction, we want to investigate how to choose S such that a (weakly) bad configuration does not exist *inside* the grid $\mathcal{A}$. We will actually focus on bad configurations, since we only need to avoid binary ghosts.
The problem can be faced from different perspectives: as an example, uniqueness of reconstruction inside a finite grid of given size $m \times n$ is guaranteed by the Katz condition [53], stating that it is not possible to have a ghost entirely contained in $\mathcal{A}$ if $h = \sum_{(a_i,b_i)\in S} a_i \geq m$ or $k = \sum_{(a_i,b_i)\in S} |b_i| \geq n$. In this case, the explicit reconstruction is obtained by means of the Mojette transform (see [46]).
On the other hand, one can assume that the set of directions S is *valid* inside the grid $\mathcal{A}$, meaning that $h < m$ and $k < n$ hold at the same time. We work under this hypothesis, so looking for further constraints on the set S to avoid ambiguities. On one hand, the hypothesis of validity allows to work with a reduced number of directions, that are also not too long (in the sense of Euclidean norm) with respect to the grid size, in general,

and so satisfy the characteristics that are required from a practical use. On the other side, since such sets of directions produce ghosts, then further information is needed to solve the ambiguity problem of reconstruction. As an example of prior information, the knowledge of the number of gray levels l will be a key point for the selection of suitable sets of directions that guarantee uniqueness.

An algebraic approach to the study of ghosts

A further description of ghosts can be given in terms of special polynomials. This approach has been introduced by Hajdu and Tijdeman in [48], and here recalled.
Given a direction $u = (a, b)$, we consider

$$f_{(a,b)}(x, y) = \begin{cases} x^a y^b - 1 & \text{if } a \neq 0, b > 0, \\ x^a - y^{-b} & \text{if } a \neq 0, b < 0, \\ x - 1 & \text{if } a = 1, b = 0, \\ y - 1 & \text{if } a = 0, b = 1. \end{cases}$$

For the sake of brevity, we will write $\mathbf{x}^u$ instead of $x^a y^b$. Given a set of directions $S = \{u_1, \ldots, u_d\}$, we define

$$F_S(x, y) = \prod_{i=1}^{d} f_{(a_i,b_i)}(x, y).$$

Using the notation $u(A) = \sum_{u \in A} u$, with $A \subseteq S$, it results

$$F_S(x, y) = \sum_{A \subseteq S} (-1)^{|S|-|A|} \mathbf{x}^{u(A)}.$$

Remark 2.1.7. We underline that the sign of the monomial $\mathbf{x}^{u(A)}$ depends on the cardinality of the set A, more precisely from its parity. This will be a key point in our approach for the detection of sets of uniqueness.

Given a density function $g : \mathcal{A} \to \mathbb{Z}$, its *generating function* is the polynomial

$$G_g(x, y) = \sum_{(\xi,\eta) \in \mathcal{A}} g(\xi, \eta) x^\xi y^\eta,$$

that associates to the lattice point (ξ, η) the monomial $m x^\xi y^\eta$, with m the weight, or multiplicity, of the corresponding point. So, we have a correspondence between generating functions and sets of (weighted) points in the integer lattice.
Conversely, given a polynomial $G(x, y)$, we denote by $G^+(x, y)$ and $G^-(x, y)$ the polynomials given by the sum of the monomials in $G(x, y)$ with positive and negative coefficients, respectively. The corresponding sets of lattice points will be $\mathcal{G}, \mathcal{G}^+$ and $\mathcal{G}^-$. If g is a ghost, then the corresponding pair $(\mathcal{G}^-, \mathcal{G}^+)$ is a (weakly) bad configuration.

Example 2.1.8. The polynomial associated to the set $S = \{(1,0),(0,1),(2,3),(4,5)\}$ is $F_S(x,y) = (x-1)(y-1)(x^2y^3-1)(x^4y^5-1) = x^7y^9 - x^7y^8 - x^6y^9 + x^6y^8 - x^5y^6 + x^5y^5 + x^4y^6 - x^4y^5 - x^3y^4 + x^3y^3 + x^2y^4 - x^2y^3 + xy - x - y + 1$.

Looking at the configuration in Fig. 2.3(d), it is immediately clear the bijection between the monomials in F_S and the lattice points in $\mathcal{F}_S$. Monomials with coefficient -1, respectively $+1$, correspond to white, respectively black, lattice points whose coordinates equal the exponents of x and y. As an example, the origin corresponds to the monomial $x^0y^0 = +1$, while the white point $(7,8)$ is associated to the monomial $-x^7y^8$. In this case, $\mathcal{F}_S$ is a bad configuration, and in fact all the coefficients in F_S are equal to 1 in absolute value (no multiple point occurs).

Theorem 2.1.9 (Hajdu and Tijdeman [48])**.** Given a set of directions S and a density function $g : \mathcal{A} \to \mathbb{Z}^2$, g has zero line sums along the directions in S if and only if $G_g(x,y) = H(x,y)F_S(x,y)$ for some $H(x,y) \in \mathbb{Z}[x,y]$.

The previous theorem basically states that each point of the discrete lattice whose coordinates are given by $u(A)$, for any $A \subseteq S$, can be associated to a monomial of $F_S(x,y)$, up to possible translations defined by the polynomial $H(x,y)$. In particular, looking at the monomials in $F_S(x,y)$, we can immediately retrieve the basic (weakly) bad configuration w.r.t. S. We denote such configuration by $\mathcal{F}_S$.

2.2 Sets of uniqueness for binary images in a finite grid

We dedicate this section to the description of a family of valid sets of directions for the unique reconstruction of binary images of given size. Starting from an important previous result in the field, by Brunetti et al. [26], we introduce the concept of *simple cycle of uniqueness*, and necessary and sufficient conditions for their construction.

2.2.1 Simple cycles

Let $S = \{u_1, u_2, u_3, u_4\}$, with $u_4 = u_1 + u_2 + u_3$ or $u_4 = u_1 + u_2 - u_3$, and $u_i = (a_i, b_i)$ for $i = 1, \ldots, 4$; $\widehat{S} = \pm(u_i - u_4)$ s.t. $u_i \in \{u_1, u_2, u_4 - u_1 - u_2\}$; $D = \pm S \cup \widehat{S}$; $A = \{(a,b) \in D$ s.t. $|a| > |b|\}$ and $B = \{(a,b) \in D$ s.t. $|b| > |a|\}$.

Theorem 2.2.1 (Brunetti et al. [26])**.** Let $S = \{u_1, u_2, u_3, u_4 = u_1 + u_2 \pm u_3\}$ be a valid set for the grid $\mathcal{A} = \{(\xi,\eta) \in \mathbb{Z}^2, 0 \le \xi < m, 0 \le \eta < n\}$, and $\sum_{i=1}^4 a_i = h$, $\sum_{i=1}^4 |b_i| = k$. Suppose that $g : \mathcal{A} \to \mathbb{Z}$ has zero line sums along the lines in the directions in S, and $||g|| \le 1$. Then, g is identically zero if and only if

$$\min_{|a|} A \ge \min\{m-h, n-k\} \text{ and } \min_{|b|} B \ge \min\{m-h, n-k\},$$

and

$$\begin{aligned} m-h < n-k, &\Rightarrow \forall (a,b) \in B \quad |a| \ge m-h \text{ or } |b| \ge n-k, \\ n-k < m-h, &\Rightarrow \forall (a,b) \in A \quad |a| \ge m-h \text{ or } |b| \ge n-k, \end{aligned}$$

where, if one of the sets A, B is empty, the corresponding condition drops.

The theorem, that extends some preliminary results obtained in [25], provides necessary and sufficient conditions for the selection of a set of four directions, valid in a grid of size $m \times n$, for the reconstruction of a binary image with no ambiguities. The authors actually characterize sets already introduced by Hajdu in [47].
Despite the relevance of the above result, some drawbacks stimulate us to look for other sets of uniqueness with similar but more general properties:

- ▷ Even if four directions represent a minimal choice for uniqueness (see Theorem 2.4 in [47]), larger sets could be desirable in view of explicit reconstruction algorithms.
- ▷ Using only four directions reflects in having large Euclidean norms, or very skew slopes, which is undesirable from a practical point of view, since facilitates the introduction of artifacts in the reconstruction process.
- ▷ The result with four directions provides uniqueness for binary images, but cannot be applied in case of an increasing number of gray levels, even when $l = 3$.

We define a *cycle* to be a set of lattice directions $S = \{u_1, \dots, u_d\}$ such that there exists a partition of $S = I \dot{\cup} J$ with $u(I) = u(J)$. A cycle is called *simple* if no other pair $\{A, B\}$ of disjoint subsets exists such that $u(A) = u(B)$, that is, no other proper subset of S is a cycle. By definition, we can always suppose, up to a reordering of indices, that the direction u_d is obtained as sums and differences of the others, being $u_d = \sum_{i=1}^{k} u_i - \sum_{j=k+1}^{d-1} u_j$, with $I = \{u_1, \dots, u_k\}$ and $J = \{u_{k+1}, \dots, u_d\}$. We underline that the sets of uniqueness characterized in Theorem 2.2.1 are all simple cycles of cardinality $d = 4$, that is the minimum number of directions that allow such a construction (see [47], Theorem 2.4).
We are interested in the weakly bad configurations associated to simple cycles, in particular when the cardinality of the set is even. Indeed, this is the only case in which we can avoid binary ghosts, according to the following result.

Proposition 2.2.2 ([2]). Let $S = I \dot{\cup} J = \{u_1, \dots, u_d\}$ be a simple cycle. Then, the associated weakly bad configuration $\mathcal{F}_S$ has a multiple point if and only if d is even. Moreover, in this case there is exactly one coefficient of $F_S(x, y)$ not in $\{0, \pm 1\}$, and its value is either $+2$ or -2.

Proof. We have to distinguish two cases, regarding the parity of the number of directions in the set S. If d is odd, then $|I|$ and $|J|$ have different parities, being a partition of S. By Remark 2.1.7, it follows that the corresponding monomials in $F_S(x, y)$, $\mathbf{x}^{u(I)}$ and $\mathbf{x}^{u(J)}$, have opposite coefficient, and therefore vanish being $u(I) = u(J)$.
On the other hand, if d is even then the coefficients of the monomials $\mathbf{x}^{u(I)}$ and $\mathbf{x}^{u(J)}$ have the same sign, since $|I|$ and $|J|$ have the same parity. So, they are both equal to $+1$ or -1, thus resulting in the monomial $\pm 2\mathbf{x}^{u(I)}$. Since the cycle S is simple by hypothesis, there is no other pair of sets $\{A, B\} \neq \{I, J\}$ such that $u(A) = u(B)$, so no other pair of monomials that sum to a monomial with coefficient greater than one in absolute value. □

We underline that the double point is defined by the partition of the set $S = I\dot{\cup}J$, and its coordinates in the discrete lattice are given by $u(I)$. From now on, we will assume $d \geq 4$ and even, since other cases are not relevant for the results we want to achieve.

Example 2.2.3. The set of directions $S_1 = \{(1,-4),(6,5),(2,1),(3,2),(4,1),(2,-1)\}$ is an even simple cycle valid in a grid of size 19×15. Its unique double point has coordinates $\lambda_\delta = (9,2)$, and it is related to the partition $I = \{(1,-4),(6,5),(2,1)\}$, $J = \{(3,2),(4,1),(2,-1)\}$. The polynomial associated to the weakly bad configuration $\mathcal{F}_{S_1}$, depicted in Fig. 2.4(a), is

$F_{S_1} = (x - y^{-4})(x^6y^5 - 1)(x^2y - 1)(x^3y^2 - 1)(x^4y - 1)(x^2 - y^{-1}) =$
$x^{18}y^9 - x^{17}y^{13} - x^{16}y^{10} - x^{16}y^8 + x^{15}y^{14} + x^{15}y^{12} - x^{15}y^7 + x^{14}y^{11} + x^{14}y^9 - x^{14}y^8 - x^{13}y^{13} + x^{13}y^{12} + x^{13}y^8 + x^{13}y^6 - x^{12}y^{12} - x^{12}y^{10} + x^{12}y^9 + x^{12}y^7 - x^{12}y^4 - x^{11}y^{13} - x^{11}y^{11} + x^{11}y^8 - x^{11}y^7 + x^{11}y^6 + x^{10}y^{11} - x^{10}y^{10} - x^{10}y^8 + x^{10}y^5 + x^{10}y^3 + x^9y^{12} - x^9y^9 - \mathbf{2x^9y^7} - x^9y^5 + x^9y^2 + x^8y^{11} + x^8y^9 - x^8y^6 - x^8y^4 + x^8y^3 + x^7y^8 - x^7y^7 + x^7y^6 - x^7y^3 - x^7y - x^6y^{10} + x^6y^7 + x^6y^5 - x^6y^4 - x^6y^2 + x^5y^8 + x^5y^6 + x^5y^2 - x^5y - x^4y^6 + x^4y^5 + x^4y^3 - x^3y^7 + x^3y^2 + x^3 - x^2y^6 - x^2y^4 - xy + y^5$.

The exponents of the monomials in F_{S_1} correspond to the coordinates of the points of the bad configuration up to a translation of $v = (0, \sum_{(a,b)\in S, b<0} |b|) = (0,5)$, which does not alter any of the following arguments (see Theorem 2.1.9).

Example 2.2.4. The set $S_2 = \{(3,1),(1,2),(1,-3),(1,3),(2,3)\}$ is valid in a grid of size 15×15. Its directions form an odd simple cycle, defined by the partition $I = \{(3,1),(1,2)\}$, $J = \{(1,-3),(1,3),(2,3)\}$. According to Proposition 2.2.2, $\mathcal{F}_{S_2}$ is a bad configuration, where no multiple points occur, see Fig. 2.4(b). Indeed, the corresponding polynomial has all coefficients equal to ± 1,

$F_{S_2} = x^8y^9 - x^7y^{12} - x^7y^7 - x^7y^6 + x^6y^{10} + x^6y^9 - x^6y^6 + x^6y^4 + x^5y^9 - x^5y^8 - x^5y^7 + x^5y^4 + x^5y^3 + x^4y^{11} - x^4y^7 + x^4y^5 - x^4y - x^3y^9 - x^3y^8 + x^3y^5 + x^3y^4 - x^3y^3 - x^2y^8 + x^2y^6 - x^2y^3 - x^2y^2 + xy^6 + xy^5 + x - y^3$.

Even in this case, the exponents of the monomials correspond to the coordinates up to a translation along $v = (0, \sum_{(a,b)\in S, b<0} |b|) = (0,3)$.

2.2.2 Enlarging regions

Before going on in our work about simple cycles, we need to introduce the concept of *enlarging region*, useful when studying sets of valid directions in a grid of given size. We entirely dedicate this section to its definition and description, since it plays a special role both in the detection of sets of uniqueness and in the implementation of an explicit reconstruction strategy.

Let us consider a finite grid $\mathcal{A}$ of size $m \times n$, and $S = \{u_1, \ldots, u_d\}$ a set of directions valid in $\mathcal{A}$, with $u_i = (a_i, b_i)$ for $i = 1, \ldots, d$. We recall that $h = \sum_{i=1}^d a_i$ and $k = \sum_{i=1}^d |b_i|$. The *enlarging region* associated to $\mathcal{F}_S$ is the rectangle

$$E = \{(\xi, \eta) \text{ s.t. } 0 \leq \xi \leq m - h - 1, 0 \leq \eta \leq n - k - 1\}.$$

Each point of the (weakly) bad configuration $\mathcal{F}_S$ can be paired with its own enlarging region. Formally, given a point $\lambda \in \mathcal{F}_S$, the *enlarging region associated to* λ is the set of

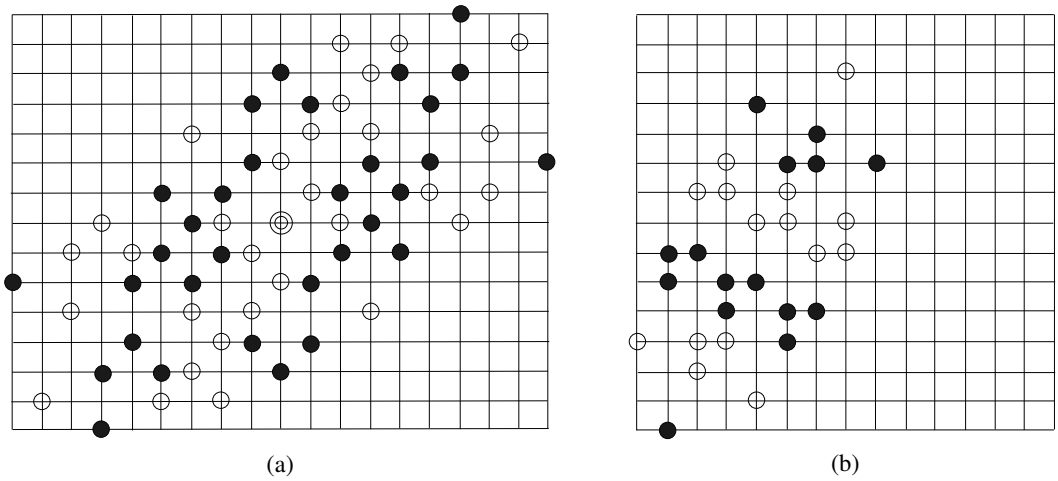


Figure 2.4: The (weakly) bad configurations associated to the sets S_1 (on the left) and S_2 (on the right) described in Examples 2.2.3 and 2.2.4.

lattice points $\lambda + E$, obtained by translating each point of the rectangle E by the vector whose coordinates are given by λ. In a general framework, enlarging regions related to different points can overlap. If this does not happen for some point, let us say λ, we say that the enlarging region associated to λ is *free*. Figure 2.5 shows an example of enlarging regions associated to the points of a weakly bad configuration.
In practice, the rectangle E characterizes the region inside which we can move a point of $\mathcal{F}_S$ still remaining inside the grid $\mathcal{A}$. Recalling how binary ghosts can be constructed (see Example 2.1.6), the idea is to look for sets S such that the construction of S-binary ghosts is not possible without exceeding the grid size. This means that, if we move the points of $\mathcal{F}_S$ inside their respective enlarging region, then the configuration keeps at least one point with multiplicity strictly greater than one.
When considering the sets of uniqueness characterized by Theorem 2.2.1, the enlarging regions of the associated weakly bad configurations show a peculiar behavior. Indeed, the following result holds:

Theorem 2.2.5 (Dulio and Pagani [34, 35])**.** Let $\mathcal{A}$ be a grid of size $m \times n$, S a set of uniqueness for $\mathcal{A}$ defined as in Theorem 2.2.1, and λ_δ the unique point in $\mathcal{F}_S = \{\lambda_0, \dots, \lambda_{14}\}$ that is counted twice. Then,

$$(\lambda_t + E) \cap (\lambda_\delta + E) = \emptyset \text{ for all indices } t \neq \delta.$$

Theorem 2.2.5 states that, in case S is a set of uniqueness chosen as in Theorem 2.2.1, then the enlarging region related to the unique double point in $\mathcal{F}_S$ does not intersect any other enlarging region, see Fig. 2.5.

Example 2.2.6. The four directions $S = \{(3,-2),(3,-1),(1,3),(5,-6)\}$ constitute a set of uniqueness in a grid of size 15×15, according to Theorem 2.2.1. The corresponding weakly bad configuration is depicted in Fig. 2.5, where the enlarging region associated to some pixels is highlighted in red. In this case, E is a rectangle of size 2×2. According

to Theorem 2.2.5, the enlarging region associated to the unique multiple point (double circled) does not intersect the others, that may overlap in general.

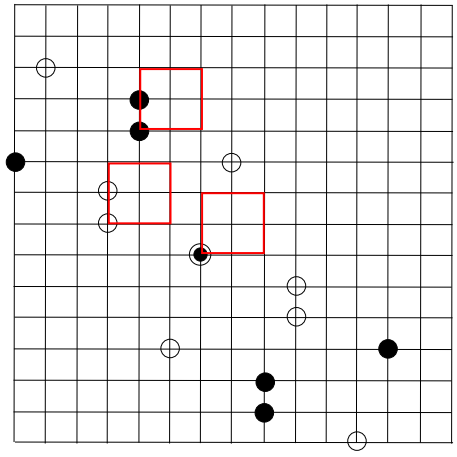


Figure 2.5: The weakly bad configuration associated to $S = \{(3,-2),(3,-1),(1,3),(5,-6)\}$. In general, different enlarging regions may overlap, while the enlarging region associated to the unique double point is always free.

Remark 2.2.7. Sometimes it is possible to choose a set of directions S such that no overlapping occurs among all the enlarging regions associated to the points of $\mathcal{F}_S$ (see Fig. 2.10 for an example).

2.2.3 Forbidden translations and uniqueness

Even simple cycles are the easiest way to construct a weakly bad configuration having only one multiple point, so in this sense are the easiest sets of directions that we can study when looking for uniqueness results. Proposition 2.2.2 provides a necessary condition for the construction of a weakly bad configuration, but the condition is not sufficient. Indeed, as we showed in Example 2.1.6, some translations which cancel the double point can be applied to $\mathcal{F}_S$, still keeping the bad configuration inside the grid $\mathcal{A}$. Then, the next step will be the characterization of the forbidden translations inside the grid, so to provide conditions that are also sufficient for the existence of a double point. The proof, also sketched in [2], follows the same steps used by the authors in [26], but we decide to report it entirely for the sake of completeness.

Let us consider a vector $w \in \mathbb{Z}^2$, that will represent a generic translation of the weakly bad configuration $\mathcal{F}_S$ associated to an even simple cycle. We call f_-^w and f_+^w the density functions associated to the generating functions $G_{f_-^w}(x,y) = (\mathbf{x}^w - 1)F_S(x,y)$ and $G_{f_+^w}(x,y) = (\mathbf{x}^w + 1)F_S(x,y)$, respectively. Our aim is to characterize those vectors w leading to bad configurations for f_-^w and f_+^w. We use the standard notation $\equiv_2$ to describe the mutual parity of numbers, crucial point when considering the cardinality of the sets that are involved in our proofs.

Lemma 2.2.8 ([2]). Let $S = \{u_1, \dots, u_d\}$ be an even simple cycle, with $I \dot{\cup} J$ the partition that individuates the unique double point $u(I) = u(J)$ in the corresponding weakly bad configuration. Then,

1. $||f_-^w|| \leq 1$ if and only if $w \in \{\pm(u(I') - u(J'))$ s.t. $I' \subseteq I, J' \subseteq J, |I'| \equiv_2 |J'|\}$,
2. $||f_+^w|| \leq 1$ if and only if $w \in \{\pm(u(I') - u(J'))$ s.t. $I' \subseteq I, J' \subseteq J, |I'| \not\equiv_2 |J'|\}$.

Proof. The proof of both statements proceeds analogously, so we focus on f_-^w only.
First, suppose that $||f_-^w|| \leq 1$. We have that all the coefficients in the corresponding polynomial $G_{f_-^w}(x, y)$ are equal to 0 or ± 1. By definition, $G_{f_-^w}(x, y) = \mathbf{x}^w F_S(x, y) - F_S(x, y)$, so we can write

$$G_{f_-^w}(x, y) = \sum_{A \subseteq S} (-1)^{|A|} \mathbf{x}^{w+u(A)} - \sum_{B \subseteq S} (-1)^{|B|} \mathbf{x}^{u(B)}. \tag{2.1}$$

We also recall that $|S|$ is even, so that $(-1)^{|S|-|A|} = (-1)^{|A|}$ in $F_S(x, y)$.
By hypothesis, S is an even simple cycle, with the unique double point in correspondence of $u(I) = u(J)$. Then, by Proposition 2.2.2, there are only two monomials in (2.1) with coefficient greater than one in absolute value, and are precisely $(-1)^{|I|} 2\mathbf{x}^{w+u(I)}$ and $(-1)^{|I|} 2\mathbf{x}^{u(I)}$, given by the first and second sum, respectively. On the other hand, we are assuming $||f_-^w|| \leq 1$. It follows that the monomial $(-1)^{|I|} 2\mathbf{x}^{w+u(I)}$ in the first sum vanishes with some monomial of the second sum having same exponent and same sign, namely, there exists a subset $K \subset S$ such that $|K| \equiv_2 |I|$ and $w + u(I) = u(K)$, meaning $w = u(K) - u(I)$. Since the sets I and J define a partition of S, there exist disjoint subsets $T \subseteq I$ and $J_1' \subseteq J$ such that $K = T \cup J_1'$. We then have

$$w = u(T) + u(J_1') - u(I) = u(J_1') - u(I_1'),$$

where I_1' is the complement of T in I. Moreover, by considerations on the cardinality we see that $|I_1'| \equiv_2 |J_1'|$, as required.
Similarly, the monomial $(-1)^{|I|} 2\mathbf{x}^{u(I)}$ in the second sum vanishes with some monomial in the first sum having same sign and exponent, that is, there exists $H \subset S$ such that $|H| \equiv_2 |I|$ and $w + u(H) = u(I)$. Since I and J define a partition of S, there exist $Q \subseteq I$ and $J_2' \subseteq J$ such that $H = Q \cup J_2'$, and then

$$w = u(I) - u(H) = u(I) - u(Q) - u(J_2') = u(I_2') - u(J_2'),$$

where I_2' is the complement of Q in I. Again, we underline that $|I_2'| \equiv_2 |J_2'|$, as required.
In both cases, we showed that $w \in \{\pm(u(I') - u(J'))$ s.t. $I' \subseteq I, J' \subseteq J, |I'| \equiv_2 |J'|\}$.

On the other hand, assume that $w \in \{\pm(u(I') - u(J'))$ s.t. $I' \subseteq I, J' \subseteq J, |I'| \equiv_2 |J'|\}$, and we want to show that in this case $||f_-^w|| \leq 1$.
Let us consider two monomials having the same exponent in the first and second sum, given by $w + u(A) = u(B)$ for some $A, B \subset S$. By the hypothesis on w, we get $u(B) - u(A) = u(I') - u(J')$ for proper $I' \subseteq I$ and $J' \subseteq J$. We consider the following disjoint subsets:

$$\begin{array}{ll} B_I = (B \setminus C) \cap I & A_I = (A \setminus C) \cap I \\ B_J = (B \setminus C) \cap J & A_J = (A \setminus C) \cap J, \end{array}$$

where $C = A \cap B$. It holds

$$w = u(I') - u(J') = u(B) - u(A) = u(B \setminus C) - u(A \setminus C) = u(B_I) + u(B_J) - u(A_I) - u(A_J).$$

Being $A_I, B_I, I' \subset I$ and $A_J, B_J, J' \subset J$, with $I \cap J = \emptyset$, $A_I \cap B_I = \emptyset$ and $A_J \cap B_J = \emptyset$, we must have $I' = B_I \cup (-A_I)$ and $-J' = B_J \cup (-A_J)$, where the notation $-A$ points out the set given by the elements in A after changing the sign of all their entries.
Finally, some considerations on the cardinality of these sets occur. We recall that, by hypothesis on w, we have $|I'| \equiv_2 |J'|$, and then $|B_I \cup (-A_I)| \equiv_2 |B_J \cup (-A_J)|$. Being the sets mutually disjoint, it holds

$$|B_I| + |A_I| \equiv_2 |B_J| + |A_J|.$$

If $|B_I| \equiv_2 |B_J|$, then $|A_I| \equiv_2 |A_J|$, and consequently $|A| \equiv_2 |B|$. On the other hand, if $|B_I| \not\equiv_2 |B_J|$, then $|A_I| \not\equiv_2 |A_J|$, and consequently $|A| \not\equiv_2 |B|$. In both cases, the monomials $(-1)^{|A|}\mathbf{x}^{w+u(A)}$ and $(-1)^{|B|}\mathbf{x}^{u(B)}$ have the same coefficient and the same exponent, and so can mutually cancel in $G_{f_-^w}$, or sum to a monomial having coefficient equal to ± 1. Therefore, $||f_-^w|| \leq 1$, and the statement follows. □

So, we have a characterization of the single translations that we can apply to $\mathcal{F}_S$ to cancel the unique multiple point of the weakly bad configuration. We address the set of forbidden translations as

$$D = \{\pm(u(I') - u(J')) \text{ s.t. } I' \subseteq I, J' \subseteq J\}.$$

Notice that, as expected, if $d = 4$ the set D coincides with the set defined before Theorem 2.2.1.

Example 2.2.9. Let us consider the even simple cycle analyzed in Example 2.2.3,

$$S = \{(1, -4), (6, 5), (2, 1), (3, 2), (4, 1), (2, -1)\}.$$

We refer to Fig. 2.4(a) for the associated weakly bad configuration. The set S is valid in a grid $\mathcal{A}$ of size 20×20, but is not of uniqueness. Indeed, it is possible to construct a binary ghost, still remaining inside $\mathcal{A}$, through the translation $w = (0, 2)$. According to Lemma 2.2.8, we have

$$G_{f_-^w}(x, y) = (\mathbf{x}^w - 1)F_S(x, y) = (y^2 - 1)F_S(x, y) = x^{18}y^{11} - x^{18}y^9 - x^{17}y^{15} + x^{17}y^{13} - x^{16}y^{12} + x^{16}y^8 + x^{15}y^{16} - x^{15}y^{12} - x^{15}y^9 + x^{15}y^7 + x^{14}y^{13} - x^{14}y^{10} - x^{14}y^9 + x^{14}y^8 - x^{13}y^{15} + x^{13}y^{14} + x^{13}y^{13} - x^{13}y^{12} + x^{13}y^{10} - x^{13}y^6 - x^{12}y^{14} + x^{12}y^{11} + x^{12}y^{10} - x^{12}y^7 - x^{12}y^6 + x^{12}y^4 - x^{11}y^{15} + x^{11}y^{11} + x^{11}y^{10} - x^{11}y^9 + x^{11}y^7 - x^{11}y^6 + x^{10}y^{13} - x^{10}y^{12} - x^{10}y^{11} + x^{10}y^8 + x^{10}y^7 - x^{10}y^3 + x^9y^{14} - x^9y^{12} - x^9y^{11} - x^9y^9 + x^9y^7 + x^9y^5 + x^9y^4 - x^9y^2 + x^8y^{13} - x^8y^9 - x^8y^8 + x^8y^5 + x^8y^4 - x^8y^3 + x^7y^{10} - x^7y^9 + x^7y^7 - x^7y^6 - x^7y^5 + x^7y - x^6y^{12} + x^6y^{10} + x^6y^9 - x^6y^6 - x^6y^5 + x^6y^2 + x^5y^{10} - x^5y^6 + x^5y^4 - x^5y^3 - x^5y^2 + x^5y - x^4y^8 + x^4y^7 + x^4y^6 - x^4y^3 - x^3y^9 + x^3y^7 + x^3y^4 - x^3 - x^2y^8 + x^2y^4 - xy^3 + xy + y^7 - y^5,$$

where no coefficient greater than one (in absolute value) is present. Referring to Fig. 2.4(a), we can see that if we repeat the construction described in Example 2.1.6, using the further translation $w = (0, 2)$, the unique double point vanishes, and we obtain a bad configuration still remaining inside a grid of 20×20 pixels.

Now that we found the set of single movements that give rise to a bad configuration, we are able to describe a generic function g assuming only values in $\{0, \pm 1\}$ inside our grid. In other words, we can characterize, through the polynomial of the corresponding generating function, any S-binary ghost inside the grid $\mathcal{A}$.

Theorem 2.2.10 ([2]). Let $S = \{u_1, \ldots, u_d\}$ be an even simple cycle, with $I \dot{\cup} J$ the partition that individuates the unique double point $u(I) = u(J)$ in the corresponding weakly bad configuration. Let $g : \mathbb{Z}^2 \to \mathbb{Z}$ be a non-trivial function having zero line sums along the lines in the directions in S. If $||g|| \leq 1$, then there exists $r \in \mathbb{N}$ such that

$$G_g(x, y) = \sum_{t=1}^{r} (\delta_t \mathbf{x}^{z_t} + \mu_t \mathbf{x}^{v_t}) F_S(x, y),$$

where $\delta_t, \mu_t \in \{\pm 1\}$ and $z_t - v_t \in D$.

The proof follows the same arguments as in [25] and [27], but we present it also here to underline the analogies among the defined set D, the notion of independence ([27]), and the notion of simple cycle. We also underline that such analogies are important for a possible generalization of our results in many directions, as we will discuss at the end of the chapter.

Proof. By Theorem 2.1.9, $G_g(x, y) = H(x, y) F_S(x, y)$ for some polynomial $H(x, y)$, that we can write explicitly as sum of single monomials, thus resulting in $G_g(x, y) = \sum_{t=1}^{r} \delta_t \mathbf{x}^{z_t} F_S(x, y)$, for some z_t and with $\delta_t = \pm 1$, for all $t = 1, \ldots, r$.
Let us consider the unique monomial in $F_S(x, y)$ with multiple coefficient, $\pm 2\mathbf{x}^{u(I)}$. This corresponds in $G_g(x, y)$ to the translated point $u(I) + z_t$, for all $t = 1, \ldots, r$. Being $||g|| \leq 1$ by hypothesis, the double coefficient must be reduced by adding a monomial of opposite sign, that is, for each $t = 1, \ldots, r$ there exists a point $v_t \neq z_t$, whose corresponding monomial in $H(x, y)$ has coefficient $\mu_t = \pm 1$, such that $u(I) + z_t = v_t + u(A_t)$ for some $A_t \subseteq S$. It follows that

$$(\delta_t \mathbf{x}^{z_t} + \mu_t \mathbf{x}^{v_t}) F_S(x, y) = \mathbf{x}^{v_t} (\delta_t \mathbf{x}^{z_t - v_t} + \mu_t) F_S(x, y)$$

has no multiple points and so, by Lemma 2.2.8, $z_t - v_t \in D$ for all $t = 1, \ldots, r$.
We finally have that $G_g(x, y) = \sum_{t=1}^{r} (\delta_t \mathbf{x}^{z_t} + \mu_t \mathbf{x}^{v_t}) F_S(x, y)$, with $\delta_t, \mu_t \in \{\pm 1\}$ and $z_t - v_t \in D$, and so the statement follows. □

We are now ready to state and prove the main result of this section, that consists in the characterization of simple cycles of uniqueness for the reconstruction, with no ambiguities, of binary images in a grid of given size.

Theorem 2.2.11 ([2]). Let $S = \{u_1, \ldots, u_d\}$ be an even simple cycle, with $u_i = (a_i, b_i)$ for $i = 1, \ldots, d$, $\sum_{i=1}^{d} a_i = h$ and $\sum_{i=1}^{d} |b_i| = k$, and valid for a lattice grid $\mathcal{A} = \{(\xi, \eta) \in \mathbb{Z}^2 \text{ s.t. } 0 \leq \xi < n, 0 \leq \eta < m\}$. Then, S is a set of uniqueness for $\mathcal{A}$ if and only if for each $w = (w_1, w_2) \in D$ it holds either $|w_1| \geq m - h$ or $|w_2| \geq n - k$.

Proof. We consider a function $g : \mathcal{A} \to \mathbb{Z}$ having zero line sums along the directions in S and with $||g|| \leq 1$. We have to show that g is identically zero if and only if either $|w_1| \geq m - h$ or $|w_2| \geq n - k$ for all $w \in D$.

First, assume that there exists $w = (w_1, w_2) \in D$ with $|w_1| < m - h$ and $|w_2| < n - k$ at the same time, and let us show that a non-trivial binary S-ghost g exists.

Since $w \in D$, by Lemma 2.2.8 we have either $||f_-^w|| \leq 1$ or $||f_+^w|| \leq 1$. We also know that the polynomials $G_{f_-^w}(x, y)$ and $G_{f_+^w}(x, y)$ have degree less or equal than $|w_1| + h$ in x and $|w_2| + k$ in y, by our assumption on w. It follows that both f_-^w and f_+^w are non-trivial functions and, being $|w_1| + h < m$ and $|w_2| + k < n$, have support inside the grid $\mathcal{A}$. Then, we can choose $g = f_-^w$ or $g = f_+^w$, depending on which one has norm less than one.

On the other hand, let us suppose that either $|w_1| \geq m - h$ or $|w_2| \geq n - k$ holds for each $w = (w_1, w_2) \in D$, and let us consider $g : \mathcal{A} \to \mathbb{Z}$, $||g|| \leq 1$, a non-trivial function having zero line sums along S. By Theorem 2.2.10,

$$G_g(x, y) = \sum_{t=1}^{r} (\delta_t \mathbf{x}^{z_t} + \mu_t \mathbf{x}^{v_t}) F_S(x, y),$$

with $z_t - v_t \in D$ and $\delta_t, \mu_t \in \{\pm 1\}$ for all $t = 1, \ldots r$, and the polynomial must have all degrees less than m in x and less than n in y, being g defined inside the grid $\mathcal{A}$. Again by Theorem 2.2.10, for all $t = 1, \ldots, r$ the polynomial $G_g(x, y)$ contains the expression

$$(\delta_t \mathbf{x}^{z_t} + \mu_t \mathbf{x}^{v_t}) F_S(x, y) = \mathbf{x}^{v_t} (\delta_t \mathbf{x}^{z_t - v_t} + \mu_t) F_S(x, y)$$

and, being $z_t - v_t = (w_1(t), w_2(t)) \in D$, it holds by assumption either $|w_1(t)| \geq m - h$ or $|w_2(t)| \geq n - k$. Since the degrees of $F_S(x, y)$ are h in x and k in y, respectively, we get that the degrees of $G_g(x, y)$ are greater or equal than m in x or greater or equal than n in y, in contradiction with our hypothesis. It follows that g is identically zero, and this concludes the proof. □

Example 2.2.12. The set $S = \{(4, -1), (4, 1), (1, 2), (3, -4), (1, -2), (5, 8)\}$ is a simple cycle of uniqueness in a grid $\mathcal{A}$ of size 20×20. We describe here all the elements necessary to verify the assumptions of Theorem 2.2.11.

Size of the grid $\mathcal{A}$: $m \times n = 20 \times 20$.

Validity of directions: $h = 18 < 20$ and $k = 18 < 20$.

Weakly bad configuration $\mathcal{F}_S$: see Figure 2.6.

Associated polynomial:

$F_S = (x^4 - y)(x^4 y - 1)(xy^2 - 1)(x^3 - y^4)(x - y^2)(x^5 y^8 - 1) =$
$x^{18}y^{11} - x^{17}y^{13} - x^{17}y^9 + x^{16}y^{11} - x^{15}y^{15} + x^{14}y^{17} + x^{14}y^{13} - x^{14}y^{12} - x^{14}y^{10} - x^{13}y^{15} + x^{13}y^{14} + x^{13}y^{12} + x^{13}y^{10} + x^{13}y^8 - x^{13}y^3 - x^{12}y^{12} - x^{12}y^{10} + x^{12}y^5 + x^{12}y + x^{11}y^{16} + x^{11}y^{14} - x^{11}y^3 - x^{10}y^{18} - x^{10}y^{16} - x^{10}y^{14} - x^{10}y^{12} + x^{10}y^{11} + x^{10}y^7 + x^9y^{16} + x^9y^{14} - x^9y^{13} - \mathbf{2x^9y^9} - x^9y^5 + x^9y^4 + x^9y^2 + x^8y^{11} + x^8y^7 - x^8y^6 - x^8y^4 - x^8y^2 - x^8 - x^7y^{15} + x^7y^4 + x^7y^2 + x^6y^{17} + x^6y^{13} - x^6y^8 - x^6y^6 - x^5y^{15} + x^5y^{10} + x^5y^8 + x^5y^6 + x^5y^4 - x^5y^3 - x^4y^8 - x^4y^6 + x^4y^5 + x^4y - x^3y^3 + x^2y^7 - xy^9 - xy^5 + y^7.$

Double point: $\lambda_\delta = (9, 2)$, whose coordinates are given by the monomial with double coefficient up to a vertical translation along the vector $v = (0, \sum_{(a,b)\in S, b<0} |b|) = (0, 7)$.

Size of the enlarging region: $E = 1 \times 1$.

Set of forbidden translations:

$$\begin{aligned} D \;=\; & \pm\{(0,4),(0,5),(0,7),(1,-2),(1,2),(1,3),(1,5),(1,7),(1,9),(2,-6),\\ & (2,5),(2,7),(3,-8),(3,-4),(3,1),(3,3),(4,-6),(4,-1),(4,1),(4,3),\\ & (4,5),(4,6),(5,1),(5,3),(5,4),(5,8),(6,6),(7,2),(8,0),(8,4)\}. \end{aligned}$$

It is easy to check that for each $w \in D$ the conditions of Theorem 2.2.11 are satisfied. Therefore, any translation that could lead to the construction of an S-binary ghost exceeds the size of the grid $\mathcal{A}$.

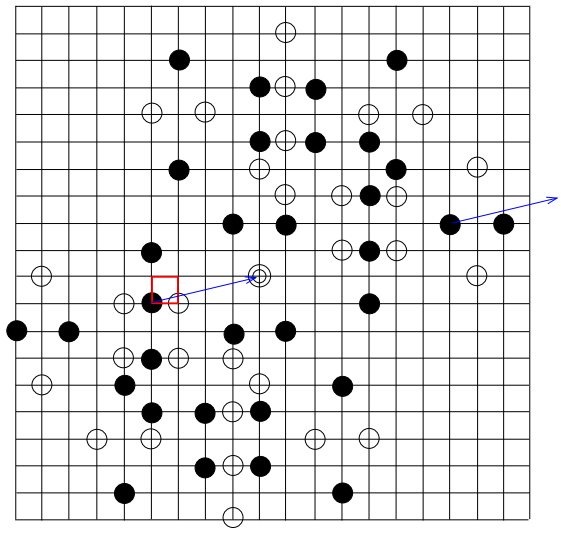


Figure 2.6: The weakly bad configuration $\mathcal{F}_S$ associated to the simple cycle of uniqueness described in Example 2.2.12. The translation along the vector $w = (4, 1) \in D$, in blue, exceeds the grid size, thus preventing the construction of a binary ghost inside $\mathcal{A}$. The same holds for all the other vectors in D.

Remark 2.2.13. Comparing Theorem 2.2.1 and Theorem 2.2.11, the latter is an extension of the previous one, where the cardinality of cycles was fixed at $d = 4$. The key points for the characterization of sets of uniqueness are the set D of forbidden translations and the enlarging region E. In fact, it is crucial that the directions in D exceed the size of E, meaning that a binary ghost cannot be constructed still remaining inside the grid $\mathcal{A}$.

Practical implications (and improvements) of our results can be summarized as follows: sets of uniqueness can be found with no restrictions about the number of directions, aside from the even cardinality, so leaving more freedom in their choice, especially concerning their length and skewness. As previously discussed, these properties are very important from the point of view of real applications.

2.2.4 Sets of uniqueness for gray-scale images

In this section, we focus on some preliminary results that we achieved for gray-scale images. Our research still bases on a uniqueness result, and aims at generalizing to $l \geq 3$ gray levels the same techniques used for the binary case.

Theorem 2.2.14 (Brunetti et al. [28]). Let $l \geq 3$, and $S = \{u_i : i = 1, \ldots, 2l\}$ be a valid set of lattice directions for a given grid $\mathcal{A}$ of size $m \times n$. Suppose that $u_i + u_{l+i} = u_j + u_{l+j}$, and both $u_i \in S$ and $u_{l+i} \in S$ cannot be written as sum of directions of $S \setminus \{u_i, u_{l+i}\}$, for $i, j = 1, \ldots, l$. Set $(a, b) = u_i + u_{l+i}$, and

$$D = \{\pm w \text{ s.t. } w = u(X) - (a, b) \neq 0, X \subseteq S\}.$$

If, for each $w = (w_1, w_2) \in D$, either $|w_1| \geq m - h$ or $|w_2| \geq n - k$ holds, then each $g : \mathcal{A} \to \mathbb{Z}$ with zero line sums along the directions in S and s.t. $||g|| \leq l - 1$ is identically zero.

First of all, we can notice many analogies between Theorem 2.2.14 and the uniqueness results provided for binary images (see Theorems 2.2.1 and 2.2.11), but it is important to underline that, for $l \geq 3$, the uniqueness conditions described in Theorem 2.2.14 are sufficient but not necessary. Indeed, the set D provided in Theorem 2.2.14 is larger than the set of translations that allow to lower the multiple point still remaining inside $\mathcal{A}$. Moreover, as shown in the following Example 2.2.15, sets of uniqueness with an odd number of directions can exist when $l \geq 3$.
Then, as far as we know, differently from the binary case, a characterization of sets of uniqueness in case of gray-scale images is not available in the literature yet.

Example 2.2.15. If we consider a grid $\mathcal{A}$ of size 34×34, the set of directions

$$S = \{(1, 1), (1, -2), (3, 4), (5, 3), (1, 6), (1, 7), (3, -10)\}$$

provides uniqueness for an image on $l = 3$ gray levels. Indeed, there are two points in the weakly bad configuration $\mathcal{F}_S$ having weight equal to 3, and it is not possible to decrease their multiplicity remaining inside $\mathcal{A}$ (see Figure 2.7).
Actually, the size of the grid can be increased in the horizontal direction to any value $m \geq 16$, still keeping the uniqueness property for the set S. Indeed, any forbidden translation exceeds the grid size vertically, if $n = 34$.

As a matter of fact, the idea that lies behind the uniqueness result provided in Theorem 2.2.14 is exactly the same carried on in case of binary images. It is essential to find a set of valid directions that gives rise to a weakly bad configuration with at least one point having multiplicity greater than or equal to l, thus preventing the existence of an S-ghost for l gray levels images inside the grid $\mathcal{A}$. So, again we need to describe the set of the corresponding forbidden translations. As we already pointed out, the set D described in the statement of the theorem does not match the set of forbidden translations, but is slightly larger.
In fact, as shown in Example 2.2.15, it is not always necessary to choose $2l$ directions for the unique reconstruction of an l-gray levels image. We now give a different interpretation

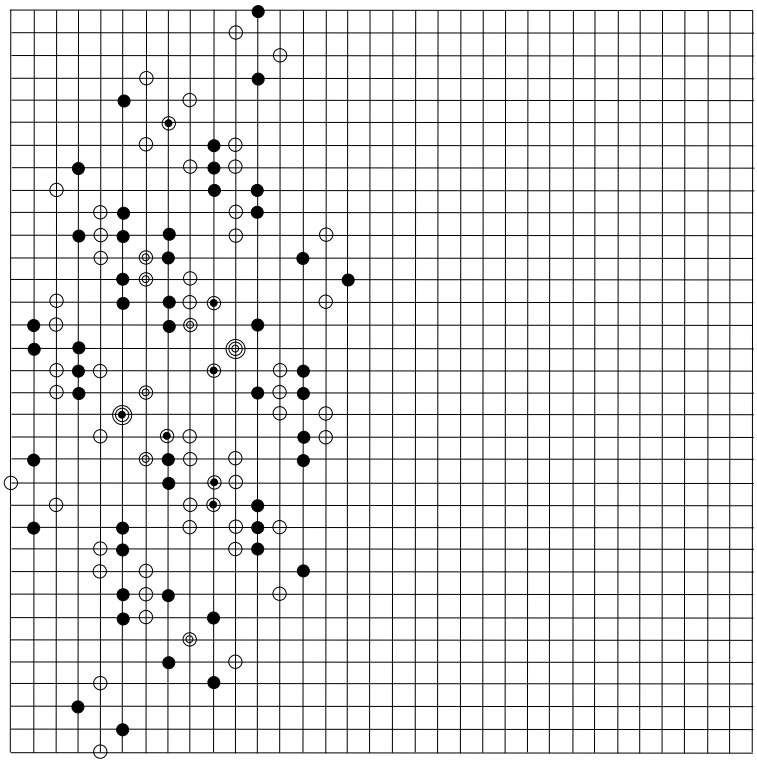


Figure 2.7: The weakly bad configuration associated to the set S described in Example 2.2.15. Double and triple points are doubly and triply circled, respectively. Any translation that may lower the multiplicity of the triple points (vertically) exceeds the size of the grid, that is 34×34 pixels. Actually, the set S is of uniqueness in any grid of size $m \times 34$, with $m \geq 16$.

to the construction of the weakly bad configuration $\mathcal{F}_S$, in terms of paths defined by subsets of directions, that allow to better visualize how multiple points with suitable weights can be easily created.
For each $I \subseteq S$, define the multisets $E_S = \{u(I) \text{ s.t. } I \subseteq S, |I| \text{ even}\}$ and $O_S = \{u(I) \text{ s.t. } I \subseteq S, |I| \text{ odd}\}$, with each point $u(I)$ counted with its proper multiplicity. Then, it results that the pair (E_S, O_S) is an S-weakly bad configuration, and each subset of directions $I \subseteq S$ can be seen as a path from the origin leading to the lattice point $u(I)$ in an even (or odd) number of steps. For a fixed lattice point $(\xi, \eta) \in \mathcal{F}_S$, each even path leading to (ξ, η) contributes with a weight of $+1$ to its final multiplicity, while each odd path contributes with -1 (or vice versa, depending on the parity of $|S|$). So, for the construction of a point with multiplicity equal to $m(\xi, \eta) = l$, it is sufficient to have l distinct even [odd] paths leading to (ξ, η), where l is apart from cancellations, meaning that every pair of paths with different parity leading to the same lattice point counts zero. This is exactly the construction that leads to simple cycles of uniqueness for binary images, where the partition $S = I \dot{\cup} J$ individuates two paths, with same parity, leading to the double point $u(I)$. It is also clear by this construction why it is not possible to have a simple cycle of uniqueness with an odd number of directions. Indeed, this is the case in which the paths given by I and J have different parity, and so contributes to the multiplicity of $u(I)$ with opposite weights, deleting it in actual fact.
In this sense, the construction provided in Theorem 2.2.14 can be seen as a generalization of simple cycles of uniqueness, where the number of directions $2l$ is the minimum possible and all the paths have length 2, the easiest case. In Example 2.2.15, we showed a set of uniqueness of odd cardinality, where the multiple point is reached by three distinct paths of odd length. We underline again that the lengths of the paths can differ each other, in general, and that their parity is the only property that matters.
When constructing these sets of directions, it becomes more difficult to control all the points of $\mathcal{F}_S$ and their behavior, as well as the computation of the set of forbidden translations D. We do not have a complete description of them at the moment, but we carried on a more theoretical study about the geometrical properties of gray-scale solutions, that give us some hints for a possible reconstruction strategy.

2.3 Theoretical aspects for a reconstruction strategy

In this section, we present a strategy for the reconstruction of a binary image of size N inside a given grid $\mathcal{A}$. While in the previous section we showed how to find sets of directions that guarantee uniqueness, here we describe a rounding method that enables us to reconstruct the unique binary image starting from the solution of $\mathbf{Ax} = \mathbf{p}$ having minimal Euclidean norm, $\mathbf{x}^* \in \mathbb{R}^N$. This is a solution of the tomographic problem having some relevant geometric properties. It is real-valued in general, and easy to compute starting from the linear system and the singular value decomposition of the matrix $\mathbf{A}$. In practice, since we deal with matrices of huge dimension, we will use numerical methods to compute an approximation $\tilde{\mathbf{x}}$ of such a solution. The same approach was already used in [34] for the implementation of the algorithm BRA, and what we present here is actually its extension to simple cycles of uniqueness.

We start recalling an important result achieved in [13] for binary solutions:

Theorem 2.3.1 (Batenburg et al. [13]). Let $\overline{\mathbf{x}}$ be a binary solution of the tomographic problem $\mathbf{Ax} = \mathbf{p}$, and $\mathbf{x}^*$ the (real-valued) solution having minimal Euclidean norm. Then,

$$||\overline{\mathbf{x}} - \mathbf{x}^*||_2 = \sqrt{\frac{||\mathbf{p}||_1}{d} - ||\mathbf{x}^*||_2^2},$$

where d is the number of employed directions.

The previous theorem states that the minimal norm solution is actually the center of a hypersphere which contains all the binary solutions of the tomographic problem $\mathbf{Ax} = \mathbf{p}$. For this reason, $\mathbf{x}^*$ is addressed as the *central solution*. Moreover, we fix the notation $\overline{\mathbf{x}}$ to point out the unique binary solution of the tomographic problem $\mathbf{Ax} = \mathbf{p}_S$, where $\mathbf{p}_S$ is the vector of projections collected along the directions of a simple cycle of uniqueness S.

2.3.1 Geometric properties of binary and gray-scale solutions

Starting from the basic idea arising from Theorem 2.3.1, we carry on a study concerning geometrical properties of gray-scale solutions, in order to get hints for possible reconstruction strategies. We consider the tomographic problem $\mathbf{Ax} = \mathbf{p}$ associated to the following parameters: N the number of pixels of the solution, l the number of its gray levels, S the set of directions along which we collect projections, and s the number of measurements, i.e., the size of the vector $\mathbf{p}$. We call $\mathcal{S}(S, s, N)$ the set of all real-valued solutions of the tomographic problem such defined. We are interested in the subset of $\mathcal{S}(S, s, N)$ of integer solutions.

Lemma 2.3.2 ([3]). All $\mathbf{x} \in \mathcal{S}(S, s, N)$ belong to the hyperplane $\pi : x_1 + \cdots + x_N = \frac{||\mathbf{p}||_1}{|S|}$.

Proof. Let us consider $\mathbf{x}$ a solution of the tomographic problem $\mathbf{Ax} = \mathbf{p}$. Since all the entries of $\mathbf{p}$ are greater than or equal to zero, it results

$$||\mathbf{p}||_1 = \sum_{i=1}^{s} p_i = \sum_{i=1}^{s} \left(\sum_{j=1}^{N} a_{ij} x_j \right) = \sum_{i=1}^{s} a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{iN}x_N.$$

On the other hand, $\mathbf{A}$ is such that in each of its column there exist exactly $|S|$ entries equal to 1, that is, $\sum_{i=1}^{s} a_{ij} = |S|$ for all $j = 1, \ldots, N$. It follows that $||\mathbf{p}||_1 = |S| \sum_{i=1}^{N} x_j$, and so the proof. □

Lemma 2.3.3 ([3]). It results $||\mathbf{A}||_F = \sqrt{N|S|}$.

Proof. It directly follows from the definition of Frobenius' norm and the fact that $\mathbf{A}$ is a binary matrix with exactly $|S|$ entries equal to 1 in each column, so that

$$||\mathbf{A}||_F^2 = \sum_{i=1}^{s} \sum_{j=1}^{N} |a_{ij}|^2 = N|S|.$$

□

Lemma 2.3.4 ([3]). The line through the origin of $\mathbb{R}^N$ and orthogonal to the hyperplane π intersects π in a point $\mathbf{C}$ corresponding to the uniform image.

Proof. We consider the parametric equations of the line through the origin and orthogonal to π, that are

$$x_i = \lambda, \qquad i = 1, \dots, N,$$

for some $\lambda \in \mathbb{R}^N$. Intersecting such a line with π, we get $\lambda = \frac{||\mathbf{p}||_1}{N|S|} = \frac{||\mathbf{p}||_1}{||\mathbf{A}||_F^2}$, and so the point $\mathbf{C} = \left(\frac{||\mathbf{p}||_1}{||\mathbf{A}||_F^2}, \dots, \frac{||\mathbf{p}||_1}{||\mathbf{A}||_F^2}\right)$, that is the uniform image.

□

As we already mentioned, the authors in [13] showed that all the binary solutions of a tomographic problem belong to a same sphere centered at the solution $\mathbf{x}^*$ with minimal Euclidean norm. A similar result holds for all real-valued solutions:

Theorem 2.3.5 ([3]). All $\mathbf{x} \in \mathcal{S}(S, s, N)$ having the same Euclidean norm belong to a same $(N-2)$-dimensional hypersphere contained in π, $\Sigma(\mathbf{C}, r_{\mathbf{x}})$, centered at the uniform image $\mathbf{C}$. Conversely, all $\mathbf{x} \in \mathcal{S}(S, s, N)$ that belong to such $(N-2)$-dimensional hypersphere have the same Euclidean norm.

Proof. We consider two solutions $\mathbf{x}$ and $\mathbf{y}$ of the tomographic problem having the same Euclidean norm, $||\mathbf{x}||_2 = ||\mathbf{y}||_2$. By Lemma 2.3.2, they both belong to π and to the $(N-1)$-dimensional hypersphere centered at the origin, that is, $\mathbf{x}, \mathbf{y} \in \Sigma(O, ||\mathbf{x}||_2) \cap \pi$. Such an intersection is an $(N-2)$-dimensional hypersphere whose center lies on the line through the origin and orthogonal to π, and so, by Lemma 2.3.4, is the uniform image. On the other hand, if we consider the $(N-2)$-dimensional hypersphere $\Sigma(\mathbf{C}, r_{\mathbf{x}}) = \Sigma(O, ||\mathbf{x}||_2) \cap \pi$, we immediately get $r_{\mathbf{x}}^2 = ||\mathbf{x}||_2^2 - ||\mathbf{C}||_2^2$ by Pythagoras' theorem, and so the thesis. □

The previous result, together with Theorem 2.3.1, leads to the following

Corollary 2.3.6 ([3]). If $\mathbf{C} \neq \mathbf{x}^*$, then binary solutions belong to an $(N-3)$-dimensional subsphere of π.

In case of gray-scale images, it is possible to give an estimate of the radius of $\Sigma(\mathbf{C}, r_{\mathbf{x}})$:

Corollary 2.3.7 ([3]). If $\mathbf{x} \in \mathcal{S}(S, s, N)$ is a gray-scale image and $\overline{\mathbf{x}} \in \mathcal{S}(S, s, N)$ is a binary image, then

$$\sqrt{q(1 - \frac{q}{N})} \leq r_{\mathbf{x}} \leq q\sqrt{\frac{N-1}{N}},$$

where $q = ||\overline{\mathbf{x}}||_2^2$ and $r_{\mathbf{x}}$ is the radius of $\Sigma(\mathbf{C}, r_{\mathbf{x}})$.

Proof. Let us consider $\mathbf{x}, \overline{\mathbf{x}} \in \mathcal{S}(S, s, N)$, with $\overline{\mathbf{x}}$ a binary image. Since binary solutions are the shortest with respect to Euclidean norm, we have that

$$r_{\mathbf{x}} \overset{\text{Thm. 2.3.5}}{=} \sqrt{||\mathbf{x}||_2^2 - ||\mathbf{C}||_2^2} \geq \sqrt{||\overline{\mathbf{x}}||_2^2 - ||\mathbf{C}||_2^2}.$$

Since $\overline{\mathbf{x}}$ is a binary image, it holds $q = \sum_{i=1}^{N} \overline{x}_i^2 = \frac{||\mathbf{p}||_1}{|S|}$ and so, by Lemma 2.3.4,

$$||\mathbf{C}||_2^2 = N\frac{||\mathbf{p}||_1^2}{||\mathbf{A}||_F^2} = \frac{||\mathbf{p}||_1^2}{N|S|^2} = \frac{q^2}{N}.$$

It follows $r_{\mathbf{x}} \geq \sqrt{q - \frac{q^2}{N}}$, and so the lower bound is given.
On the other hand, since all gray-scale solutions belong to the hyperplane $\pi : x_1 + \cdots + x_N = \frac{||\mathbf{p}||_1}{|S|}$, we have

$$||\mathbf{x}||_2^2 \leq \left(\sum_{i=1}^{N} x_i\right)^2 = \frac{||\mathbf{p}||_1^2}{|S|^2} = q^2$$

and then, by Theorem 2.3.5, the upper bound

$$r_{\mathbf{x}} \leq \sqrt{\max_{\mathbf{x}\in\mathcal{S}(S,s,N)} ||\mathbf{x}||_2^2 - ||\mathbf{C}||_2^2} \leq \sqrt{q^2 - \frac{q^2}{N}}.$$

□

We conclude the study of the set of solutions $\mathcal{S}(S, s, N)$ and their geometric properties by considering the N-dimensional hypercube H^N, having one vertex at the origin, sides of length 1 parallel to the coordinate axes, and all vertices with non-negative coordinates.

Lemma 2.3.8 ([3]). If $\mathcal{S}(S, s, N)$ contains a binary solution, then $\frac{||\mathbf{p}||_1}{|S|} \in \{0, \ldots, N\}$.

Proof. Let us consider $\mathbf{v}$ a vertex of the hypercube H^N. We have $||\mathbf{v}||_2^2 = ||\mathbf{v}||_1 \in \{0, \ldots, N\}$. Since each binary image corresponds to one of the vertices of H^N, it follows $\frac{||\mathbf{p}||_1}{|S|} = ||\mathbf{v}||_2^2$ for some vertex $\mathbf{v}$, and so the thesis. □

Theorem 2.3.9 ([3]). Let $N \geq 2$ and $0 \neq q = \frac{||\mathbf{p}||_1}{|S|} < N$. Then, π is the unique affine hyperplane of $\mathbb{R}^N$ containing the set of all vertices of H^N having two-norm $\sqrt{q}$.

Sketch of the proof. We consider all the vertices of H^N having Euclidean norm equal to $\sqrt{q}$, where we recall that $q = \frac{||\mathbf{p}||_1}{|S|}$. Let us denote this set H_q^N. By definition, each $\mathbf{v} \in H_q^N$ has q entries equal to 1, and so $\mathbf{v} \in \pi$. Then, we have to prove that there does not exist any other affine hyperplane in $\mathbb{R}^N$ containing the set H_q^N.
We define $\mathbf{e}_\kappa = (\mathbf{0}_{\kappa-1}, 1, \mathbf{0}_{N-\kappa})$ the vector having all entries equal to zero except for the κ-th entry, that is equal to 1, and we denote $\mathbf{v}_r^1 = (\mathbf{1}_{r-1}, 0, \mathbf{1}_{q-r+1}, \mathbf{0}_{N-q-1})$ for $r = 1, \ldots, q$ the vertex in H_q^N having the r-th entry and the last $N - q - 1$ entries all equal to 0. Then, we can write $\mathbf{e}_1$ as a linear combination of elements of H_q^N, that is,

$$\mathbf{e}_1 = -\frac{q-1}{q}\mathbf{v}_1^1 + \frac{1}{q}\mathbf{v}_2^1 + \cdots + \frac{1}{q}\mathbf{v}_{q+1}^1.$$

A similar argument holds for all $\mathbf{e}_\kappa$, with $\kappa = 1, \ldots, N$. It follows that H_q^N spans $\mathbb{R}^N$, thus implying that all the elements of H_q^N belong to the same affine hyperplane. □

Remark 2.3.10. Theorem 2.3.9 actually shows that each solution of the tomographic problem can be obtained as a linear combination of elements of H_q^N.

The previous remark suggests a possible approach for the computation of the solutions of the tomographic problem. We recall that we are looking for integer solutions on l gray levels, so we need a specific algorithm for their computation. On the other side, real-valued solutions can be retrieved using standard numerical techniques already available in the literature. Since all the solutions are linear combinations of elements of H_q^N, the idea is to start from the central solution $\mathbf{x}^*$, easy to compute, and then manipulate the coefficients of the corresponding linear combination to move to a suitable integer solution. In case of binary images, such a manipulation results in a rounding operation, that leads to the implementation of the algorithm BRA [34]. In case of gray-scale images, with $l \geq 3$, a correct algorithm still does not exist, up to our knowledge, but some experimental results suggest that similar techniques could work even in this case (see Section 2.4).

2.3.2 A rounding theorem

Let us consider the weakly bad configuration $\mathcal{F}_S$ associated to the simple cycle of uniqueness S. As turns out from the proof of Proposition 2.2.2, it is composed by $2^{|S|} - 1$ distinct lattice points λ_i, all of them with multiplicity equal to ± 1 or ± 2. We divide them in two sets based on the sign of their weights, thus defining the sets of indices I^- and I^+. Depending on the sign of the unique double point λ_δ, we have $|I^-| = 2^{|S|-1}$ and $|I^+| = 2^{|S|-1} - 1$, or vice versa. For a fixed vector contained in the enlarging region, $w \in E$, we consider the bad configuration $\mathcal{G}_w = \mathcal{F}_S + w$ obtained by translating a copy of $\mathcal{F}_S$ after changing the sign of the weights, and the associated binary S-ghost

$$g_w(x,y) = \begin{cases} 0 & \text{if } (x,y) \notin \mathcal{G}_w, \\ 1 & \text{if } (x,y) = \lambda_i + w, i \in I^+, \\ -1 & \text{if } (x,y) = \lambda_i + w, i \in I^-. \end{cases}$$

Then, the central solution $\mathbf{x}^*$ can be written as the sum of the unique binary solution $\overline{\mathbf{x}}$ and the S-ghost g_w, namely,

$$\mathbf{x}^*(\xi,\eta) = \overline{\mathbf{x}}(\xi,\eta) + \sum_{w \in E} \alpha_w^* g_w(\xi,\eta)$$

for all $(\xi,\eta) \in \mathcal{A}$ and some coefficients $\alpha_w^* \in \mathbb{R}$. We also recall that $g_w(\xi,\eta)$ takes value equal to the multiplicity $m(\xi,\eta)$ of the point (ξ,η), that is different from zero if and only if we are in the set $\bigcup_{w \in E} \mathcal{G}_w$. Then, it is sufficient to compute the value $\sum_{w \in E} \alpha_w^* g_w(\xi,\eta)$ for all the lattice points in $\mathcal{A}$, and then the binary solution can be immediately reconstructed starting from $\mathbf{x}^*$. The multiplicity values $m(\xi,\eta)$ are well known from the structure that we chose for S, an even simple cycle, while the values α_w^* can be computed through an easy rounding operation, as showed in the following Theorem 2.3.11. As previously mentioned, our result is a generalization of Theorem 13 in [34], and also the proof proceeds similarly.

Theorem 2.3.11 ([2]). Let S be a simple cycle of uniqueness for an $m \times n$ lattice grid $\mathcal{A}$, and let $\mathbf{x}^*$ be the central solution of the tomographic problem $\mathbf{Ax} = \mathbf{p}_S$. Then, for all $w \in E$, it results

$$\alpha_w^* = \mathbf{x}^*(\lambda_\delta + w) - \text{round}(\mathbf{x}^*(\lambda_\delta + w)),$$

with λ_δ the unique double point in $\mathcal{F}_S$ and round(x) the closest integer to x.

Proof. We recall that S is an even simple cycle, so that $m(\xi, \eta) \in \{\pm 1, \pm 2\}$ for each $(\xi, \eta) \in \mathcal{F}_S$, with one only multiple point, $\lambda_\delta \in \mathcal{F}_S$, such that $|m(\lambda_\delta)| = 2$.
We define the function $f : \mathbb{R} \to \mathbb{R}$ for all $w \in E$,

$$f(\alpha_w) = \sum_{i \in I^- \cup I^+} \big(\overline{\mathbf{x}}(\lambda_i + w) + \alpha_w m(\lambda_i)\big)^2.$$

Now, when considering a general real-valued solution $\mathbf{y}$ of the tomographic problem $\mathbf{Ax} = \mathbf{p}_S$, we have

$$\begin{aligned}
||\mathbf{y}||_2^2 &= \sum_{(\xi,\eta)\in\mathcal{A}} \mathbf{y}^2(\xi,\eta) = \sum_{(\xi,\eta)\in\mathcal{A}} \left(\overline{\mathbf{x}}(\xi,\eta) + \sum_{w \in E+(\xi,\eta)} \alpha_w g_w(\xi,\eta) \right)^2 \\
&= \sum_{(\xi,\eta)\notin\mathcal{G}_w} \overline{\mathbf{x}}^2(\xi,\eta) + \sum_{(\xi,\eta)\in\mathcal{G}_w} \left(\overline{\mathbf{x}}(\xi,\eta) + \sum_{w \in E+(\xi,\eta)} \alpha_w g_w(\xi,\eta) \right)^2 \\
&= \sum_{(\xi,\eta)\notin\mathcal{G}_w} \overline{\mathbf{x}}^2(\xi,\eta) + \sum_{w\in E} \left[\sum_{i \in I^- \cup I^+} \big(\overline{\mathbf{x}}(\lambda_i + w) + \alpha_w g_w(\lambda_i + w)\big)^2 \right] \\
&= \sum_{(\xi,\eta)\notin\mathcal{G}_w} \overline{\mathbf{x}}^2(\xi,\eta) + \sum_{w\in E} \left[\sum_{i \in I^- \cup I^+} \big(\overline{\mathbf{x}}(\lambda_i + w) + \alpha_w m(\lambda_i)\big)^2 \right] \\
&= \sum_{(\xi,\eta)\notin\mathcal{G}_w} \overline{\mathbf{x}}^2(\xi,\eta) + \sum_{w\in E} f(\alpha_w).
\end{aligned}$$

Since we are interested in the central solution $\mathbf{x}^*$, that corresponds to that one with minimal Euclidean norm, we need to minimize $||\mathbf{y}||_2^2$. This value is given by the sum of a constant positive term and $|E|$ copies of the non-negative value $f(\alpha_w)$, so it is sufficient to minimize f with respect to each variable. Computing its derivative, it results

$$\alpha_{w,min} = \alpha_w^* = -\frac{\sum_{i \in I^- \cup I^+} m(\lambda_i)\overline{\mathbf{x}}(\lambda_i + w)}{\sum_{i \in I^- \cup I^+} m^2(\lambda_i)}.$$

We make some considerations about the value assumed by α_w^*. If the double point λ_δ has negative sign, then $m(\lambda_\delta) = -2$, $|I^-| = 2^{|S|-1} - 1$ and $|I^+| = 2^{|S|-1}$. On the other hand, if its sign is positive, we have $m(\lambda_\delta) = +2$, $|I^-| = 2^{|S|-1}$ and $|I^+| = 2^{|S|-1} - 1$. Then, α_w^* achieves its minimum value when $\overline{\mathbf{x}}(\lambda_i) = 0$ for all $i \in I^-$ and $\overline{\mathbf{x}}(\lambda_i) = 1$ for all $i \in I^+$, while its maximum when $\overline{\mathbf{x}}(\lambda_i) = 0$ for all $i \in I^+$ and $\overline{\mathbf{x}}(\lambda_i) = 1$ for all $i \in I^-$. As a consequence, we get that

$$\alpha_w^* \in \left[-\frac{\sum_{i\in I^+} m(\lambda_i)}{2^{|S|}+2}, \frac{\sum_{i\in I^-} m(\lambda^i)}{2^{|S|}+2}\right] = \left[-\frac{2^{|S|-1}}{2^{|S|}+2}, \frac{2^{|S|-1}}{2^{|S|}+2}\right] \subset \left[-\frac{1}{2}, \frac{1}{2}\right]. \tag{2.2}$$

By Theorem 2.2.5, we know that there exists a unique α_w to be computed for each pixel in $\lambda_\delta + E$, and therefore, by Eq. (2.2),

$$\text{round}(\mathbf{x}^*(\lambda_\delta + w)) = \text{round}(\overline{\mathbf{x}}(\lambda_\delta + w) + \alpha_w^*) = \overline{\mathbf{x}}(\lambda_\delta + w).$$

It follows that the binary solution $\overline{\mathbf{x}}$ is exactly reconstructed in $\lambda_\delta + E$.
Moreover, being $\mathbf{x}^*(\lambda_\delta + w) = \overline{\mathbf{x}}(\lambda_\delta + w) + \alpha_w^*$, we can also explicitly compute the value of each α_w^*, that is,

$$\alpha_w^* = \mathbf{x}^*(\lambda_\delta + w) - \overline{\mathbf{x}}(\lambda_\delta + w) = \mathbf{x}^*(\lambda_\delta + w) - \text{round}(\mathbf{x}^*(\lambda_\delta + w)),$$

thus concluding the proof. □

As a consequence, the values α_w^* can be immediately computed from $\mathbf{x}^*$, and the binary solution can be retrieved from the central solution as $\overline{\mathbf{x}}(\xi,\eta) = \mathbf{x}^*(\xi,\eta) - \sum_{w\in E} \alpha_w^* g_w(\xi,\eta)$.

Corollary 2.3.12 ([2])**.** Let S be a simple cycle of uniqueness for a lattice grid $\mathcal{A}$. Then, the unique binary solution $\overline{\mathbf{x}}$ is uniquely and explicitly reconstructible from the central solution $\mathbf{x}^*$.

An important remark is the following:

Remark 2.3.13. If we choose S in a way such that in $\mathcal{F}_S$ all the pixels have disjoint enlarging regions, then $\overline{\mathbf{x}}(\xi,\eta) = \text{round}(\mathbf{x}^*(\xi,\eta))$ for all $(\xi,\eta) \in \mathcal{A}$. This is due to the fact that each pixel in $\overline{\mathbf{x}}$ differs from the corresponding pixel in $\mathbf{x}^*$ for one value α_w^* only and, since they all belong to the interval $(-\frac{1}{2}, \frac{1}{2})$, the rounding operation is sufficient for its computation. Even if trivial in the theoretical aspect, this is very important from a computational point of view, since such a characteristic in a simple cycle of uniqueness allows to define a reconstruction algorithm that is more efficient both in terms of running time and computational cost, as highlighted in our experimental results.

2.4 Reconstruction of binary and gray-scale images

We are now ready to describe the reconstruction algorithm e-BRA, implemented as a refinement of the algorithm BRA in [34], where the only admissible input was a set of uniqueness as in Theorem 2.2.1. We report its pseudo-code in Algorithm 1.
As we already noticed, the algorithm essentially computes the central solution of the tomographic problem and then, after suitable adjustments and roundings, the unique binary solution. Some comments:

Algorithm 1. e-BRA

Input: An unknown image $\mathcal{A}$ of size $m \times n$, a simple cycle of uniqueness S for $\mathcal{A}$

1 Collect the vector of projections along the directions in S, and set the tomographic problem $\mathbf{A}\mathbf{x} = \mathbf{p}_S$;

2 Using a suitable subroutine, compute an approximation of the solution with minimal Euclidean norm, $\tilde{\mathbf{x}}$;

3 Compute the double point $\lambda_\delta \in \mathcal{F}_S$ and the enlarging region E;

4 **for** *all* $w \in E$ **do**

5 | compute $\alpha_w = \tilde{\mathbf{x}}(\lambda_\delta + w) - \text{round}(\tilde{\mathbf{x}}(\lambda_\delta + w))$ and $g_w(\xi, \eta)$;

6 **end**

7 Compute $\overline{\mathbf{x}}(\xi, \eta) = \tilde{\mathbf{x}}(\xi, \eta) - \sum_{w \in E} \alpha_w g_w(\xi, \eta)$;

Output: $\overline{\mathbf{x}}$

Computational cost: e-BRA is divided in two main phases. We first have the computation of an approximation $\tilde{\mathbf{x}}$ of the central solution. The computational cost of this step depends on the choice of the algorithm that returns $\tilde{\mathbf{x}}$, that can be arbitrarily chosen. In our implementation, we exploit the Conjugate Gradient Least Square algorithm (CGLS), whose cost in the input size is $O(s\sqrt{mn})$, with s the size of the vector of projections $\mathbf{p}_S$. During the second phase, e-BRA computes the weakly bad configuration $\mathcal{F}_S$ and then the enlarging region of each point, thus resulting in a computational cost of $O((m-h)(n-k)(nm))$. As a conclusion, we have that e-BRA runs in $O(\max\{s\sqrt{mn}, (m-h)(n-k)(nm)\})$ time [34]. Actually, a proper choice of the set S can reduce this cost to the running time required for the calculation of $\tilde{\mathbf{x}}$ only. Notice that this cannot be further reduced.

Choice of the subroutine: instead of CGLS, many other subroutines can be chosen. In particular, the linear modeling of the tomographic problem is well suited for quadratic integer programming algorithms too. As previously remarked, this choice becomes essential for the improvement of e-BRA, since the computation of $\tilde{\mathbf{x}}$ cannot be avoided.

Cardinality of the set S: potentially, the set S can contain any even number of suitable directions. A higher cardinality results in a smaller size of E, and so in a reduction of the computational cost for the calculation of the values α_w. On the other hand, the size of collected data highly increases, as well as the required cost for computing $\tilde{\mathbf{x}}$. From our experiments, it seems that the best choice is $|S| = 6$, with the unique double point λ_δ determined by two sets of equal cardinality, $|I| = |J| = 3$. A further improvement consists in the choice of sets S leading to enlarging regions intersecting in a reduced number of points, or even, possibly, mutually disjointed.

Example 2.4.1. We describe the implementation of e-BRA, step-by-step, on a small image of 15×15 pixels, depicted in Figure 2.8. We use a simple cycle of uniqueness of six directions, $S = \{(2,-1), (2,-3), (3,2), (1,2), (2,1), (4,-5)\}$.
First of all, the algorithm collects the projections along S and sets the tomographic problem $\mathbf{A}\mathbf{x} = \mathbf{p}_S$. We do not report the data explicitly, since the dimensions are too large ($\mathbf{A} \in \{0,1\}^{382 \times 255}$ and $\mathbf{p}_S \in \mathbb{N}^{382}$).

Then, using the CGLS algorithm, an approximation $\tilde{\mathbf{x}}$ of the central solution is computed. We stop the subroutine after $\iota = 13$ iterations:

$$\tilde{\mathbf{x}} = \begin{pmatrix}
\begin{matrix} 0.9862 & -0.0028 & 0.9904 & 0.0098 & 0.0264 & 0.9407 & -0.0130 & 0.9638 & 0.7437 & 0.8720 \\ 0.8325 & 0.0410 & -0.0316 & -0.0080 & 1.0173 & & & & & \end{matrix} \\
\begin{matrix} -0.0024 & 0.9914 & -0.0055 & -0.0073 & 0.0239 & -0.0473 & 0.0803 & 0.0499 & 1.1179 & 1.0266 \\ 1.1064 & 1.1964 & 0.0882 & 0.9353 & 0.0069 & & & & & \end{matrix} \\
\begin{matrix} 0.0004 & -0.0181 & 1.0439 & 0.9892 & 0.0008 & 0.1684 & -0.0138 & 1.1518 & 0.9853 & 0.2706 \\ 0.8528 & 1.1044 & 1.0677 & -0.0072 & 0.0053 & & & & & \end{matrix} \\
\begin{matrix} 0.0078 & 0.9783 & 0.0288 & -0.0817 & 0.0321 & -0.0834 & -0.0126 & 0.8700 & -0.2064 & 0.9539 \\ 0.9756 & 0.0130 & 1.0319 & -0.0412 & 1.0760 & & & & & \end{matrix} \\
\begin{matrix} 0.0308 & 0.0326 & 0.0462 & 0.2594 & 1.0827 & 0.2275 & -0.1575 & 0.3360 & -0.1508 & 1.0140 \\ 0.9956 & 1.0907 & -0.0370 & 0.7510 & 0.9952 & & & & & \end{matrix} \\
\begin{matrix} 0.0398 & 0.8913 & 0.9733 & 0.8967 & -0.0263 & 0.8737 & 0.7553 & 0.0458 & -0.0001 & 0.9627 \\ -0.1126 & -0.1026 & 0.0492 & 0.8485 & 0.9482 & & & & & \end{matrix} \\
\begin{matrix} 0.8918 & 0.9873 & 0.7817 & 0.9773 & 0.7196 & 0.2573 & 0.8236 & 0.9757 & -0.0741 & 0.0672 \\ 0.1362 & 0.9200 & 0.3483 & 0.0662 & 0.1618 & & & & & \end{matrix} \\
\begin{matrix} 0.0747 & 1.0416 & 1.1361 & 1.1981 & 0.0297 & 0.1061 & -0.0251 & 0.0348 & 0.9670 & 0.8631 \\ 0.1246 & 0.9641 & -0.1041 & 0.8902 & -0.0749 & & & & & \end{matrix} \\
\begin{matrix} -0.0477 & 1.0824 & 0.7392 & 0.1516 & 0.9709 & 1.0280 & 0.9036 & 0.1022 & 1.2764 & -0.3104 \\ 0.0739 & 0.8967 & 0.1148 & 0.8216 & 0.0573 & & & & & \end{matrix} \\
\begin{matrix} 1.0144 & 0.0345 & 0.8415 & 1.0719 & 0.0864 & 0.1403 & 0.0463 & 0.9658 & 1.1908 & 0.0766 \\ 0.0565 & 0.1261 & 1.0815 & 0.1489 & 0.9600 & & & & & \end{matrix} \\
\begin{matrix} 0.9394 & 1.0721 & -0.0563 & 0.0829 & 0.9510 & 0.8715 & 0.0297 & 0.5774 & -0.0563 & 0.7465 \\ 1.1238 & -0.1512 & 0.9632 & -0.0377 & 0.0075 & & & & & \end{matrix} \\
\begin{matrix} 0.9438 & 0.0209 & -0.0673 & 0.0733 & 0.0331 & 0.8871 & 0.0373 & 1.0855 & 1.0711 & 0.0654 \\ -0.0177 & 0.0803 & -0.0560 & -0.0029 & 0.0230 & & & & & \end{matrix} \\
\begin{matrix} 0.9787 & 0.0327 & 0.9519 & 0.9652 & 0.2806 & 0.9389 & 0.1117 & -0.0789 & 0.2100 & -0.1226 \\ -0.0219 & 1.0163 & 0.0110 & -0.0013 & 1.0114 & & & & & \end{matrix} \\
\begin{matrix} 0.9780 & 0.0398 & 0.9824 & -0.0134 & 0.9777 & 0.8951 & 0.8634 & -0.1066 & 0.8953 & 1.0784 \\ 1.0031 & -0.0043 & 0.0325 & 1.0001 & 0.9893 & & & & & \end{matrix} \\
\begin{matrix} 0.9880 & 0.0094 & 0.0478 & 0.9997 & 0.1022 & 0.8954 & 0.1530 & 0.0670 & 0.0015 & 0.0020 \\ 0.0016 & -0.0140 & -0.0042 & 0.0083 & 0.0049 & & & & & \end{matrix}
\end{pmatrix}$$

We observe that $\tilde{\mathbf{x}}$ is a real-valued solution, with both positive and negative entries. Finally, e-BRA computes the double point $\lambda_\delta = (7, -2)$, the enlarging region $E = 1 \times 1$, and then provides as output the matrix

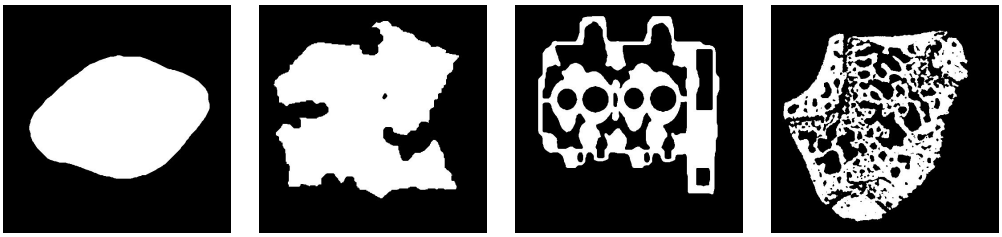
$$\overline{\mathbf{x}} = \begin{pmatrix}
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix},$$

that is exactly the binary representation of Fig. 2.8. We conclude that e-BRA performed a perfect reconstruction of the input image.

Figure 2.8: The binary image of size 15×15 on which we performed e-BRA, step-by-step.

2.4.1 Binary images

In order to give practical evidence to our results, we tested e-BRA for the reconstruction of four binary images already used in literature for similar experiments [34]. Each phantom has a size of 512×512 pixels, and they are all depicted in Figure 2.9.



(a) (b) (c) (d)

Figure 2.9: The four phantoms that we used for our tests.

The four images are quite different in shape and geometrical properties, so particularly suitable for our tests. We start with a very simple shape, representing a connected and slightly convex body, Fig. 2.9(a). We then move to an object with indented borders, but essentially connected (unless some missing pixels), Fig. 2.9(b). We finish with two non-connected bodies, increasing the irregularity of their shape and borders, Fig. 2.9(c) and Fig. 2.9(d).
Since we basically extended BRA allowing cycles of any size, we compare our results ([2]) with those ones published in [34]. Using the same iterative subroutine, CGLS, we show that our implementation requires a very lower number of iterations to achieve a perfect reconstruction of all four phantoms, and that the running time of the whole algorithm can be drastically reduced with a correct choice of the input set S. Since we deal with grids of big size (512×512 pixels), our strategy for the generation of cycles of uniqueness consists in the following: we randomly generated about 10^6 sets of six directions, and then checked for each of them if the hypothesis of Theorem 2.2.11 were satisfied. The value $d = 6$ has been chosen as the first generalization of $d = 4$, that is the size used for the experiments in [34]. Furthermore, we randomly generated sets of directions satisfying the condition $u_1 + u_2 + u_3 = u_4 + u_5 + u_6$. The reason of this choice stands in the fact that

having $|I| = |J| = 3$ helps to keep a sort of balance among the directions with respect to their length, in the sense of Euclidean norm, thus avoiding to pick the direction u_6 very skew. In general, the random generating algorithm can be slightly modified to get simple cycles of uniqueness of any size and with any admissible partition $|S| = |I| + |J|$.
Among all the suitable outputs, we selected two sets of directions:

$$S_1 = \{(92, -47), (91, -61), (71, 59), (44, -89), (98, 39), (112, 1)\},$$
$$S_2 = \{(98, -81), (99, 19), (58, -55), (65, 68), (1, 51), (189, -236)\}.$$

The choice was made according to some parameters that we evaluated to measure the goodness of a simple cycle of uniqueness: the maximum length among all the directions, n_{max}, the size of the enlarging region, E, and the parameter $R = \frac{n_{max}^2 \cdot |E|}{mn}$. For each of them, we selected the set of uniqueness achieving minimum value. The values related to S_1 and S_2 are reported in Table 2.1, where we also evaluate the same parameters for the set of directions $S_* = \{(80, 77), (81, 91), (80, 83), (241, 251)\}$, used in [34], whose reconstruction outputs are the starting point for our comparisons.

	$\mathbf{n_{max}^2}$	$\mathbf{\lvert E \rvert}$	**R**
S_*	121082	300	138.567
S_1	12545	864	41.347
S_2	91417	4	1.395

Table 2.1: The parameters related to the sets of directions that we choose for our tests.

The choice of these parameters has been done according to practical criteria: we want to collect as many projections as possible using few directions, so we need them to be short and not so skew (n_{max}); we want a fast algorithm, so a small enlarging region to avoid a big computational cost in the second phase of the algorithm ($|E|$). This cost becomes gradually lower when the number of intersections among different enlarging regions decreases, and it is null when they are all disjoint. The parameter R actually summarizes all the required properties for a good set of directions S.
After the generation and selection of cycles, we proceed with the reconstruction test. For each phantom, we run e-BRA gradually increasing the number of iterations required by the CGLS subroutine, that is the main parameter used to measure the improvement of e-BRA with respect to BRA. As showed in Tables 2.3, 2.4, 2.5 and 2.6, the use of cycles of uniqueness of size six instead of four allows to reach a perfect reconstruction in less iterations. The goodness of reconstruction is evaluated in terms of percentage of correctly reconstructed pixels (see Tables 2.3 and 2.4) and number of wrong pixels (see Tables 2.5 and 2.6). As a final criterion, we also report in Table 2.2 the average running time of BRA and e-BRA for the selected sets of directions. It is interesting to notice how this parameter drastically reduces on varying of the size of E: when using S_*, the algorithm runs in ~ 14 seconds; moving to six directions, we have a first growth to ~ 10 minutes when using the set S_1, due to the fact that the enlarging region has a bigger size. Finally, when using S_2, the running time of our algorithm collapses in few seconds, since we actually skip the second phase of the reconstruction process. Indeed, S_2 is selected in a way such that no intersections occur among the enlarging regions of the pixels in

$\mathcal{F}_{S_2}$. So, the last part of the algorithm consists in a rounding approximation only, that requires $O(1)$ time of execution, and e-BRA actually coincides with the rounding of the central solution given as output by the CGLS. For a better visualization, we depict in Figure 2.10 the size, position and intersections of the enlarging regions of the points in $\mathcal{F}_{S_1}$ and $\mathcal{F}_{S_2}$.

	Phantom 1	**Phantom 2**	**Phantom 3**	**Phantom 4**
S_*	14.09 s	14.16 s	13.98 s	14.05 s
S_1	597.64 s	597.09 s	585.42 s	597.09 s
S_2	10.67 s	10.66 s	10.69 s	10.75 s

Table 2.2: The average running time of e-BRA for the reconstruction of the four binary images depicted in Fig. 2.9 (in seconds).

Some final remarks about our experiments:

- ▷ The random generation of simple cycles of uniqueness is very expensive from a computational point of view, since the computation of the set D is fundamental to check if the hypothesis of Theorem 2.2.11 are satisfied, and consists of an exhaustive generation of all the subsets of S. An open research line in this direction is the possibility of developing smarter strategies for the selection of suitable sets of directions, possibly exploiting some iterative procedure, instead of a random one.
- ▷ The CGLS is an iterative algorithm that we used to compute an approximation of the central solution $\mathbf{x}^*$. Changing the subroutine could improve the general performance of the algorithm. As an example, the modeling of the tomographic problem as a linear system suggests the possibility of application of techniques in the field of integer linear programming.
- ▷ When moving from four to six directions, an impressive improvement in the performance of the algorithm arises. Such an improvement can be seen even when moving from six to eight directions, but it is less relevant both in terms of running time and number of required CGLS' iterations. These remarks suggest us that the best choice for the cardinality of a simple cycle of uniqueness is $|S| = 6$, at least from a practical point of view.

2.4.2 Generalization of e-BRA to gray-scale images

Since sets of uniqueness can be selected even for gray-scale images, the idea of a reconstruction algorithm for images with different gray-levels, similar to e-BRA, is very tempting. Unfortunately, at least for the moment, we have not been able to provide a theoretical result similar to Theorem 2.3.11, essential for the correctness of e-BRA. On the other hand, as discussed in Section 2.3.1, the gray-scale images that are the solution of some tomographic problem $\mathbf{Ax} = \mathbf{p}$ show very strong geometrical properties. Above all, they can be written as a linear combination of elements of the hypercube H_q^N, where we recall that N is the number of pixels of the image and $q = ||\overline{\mathbf{x}}||_2^2$, with $\overline{\mathbf{x}}$ a binary

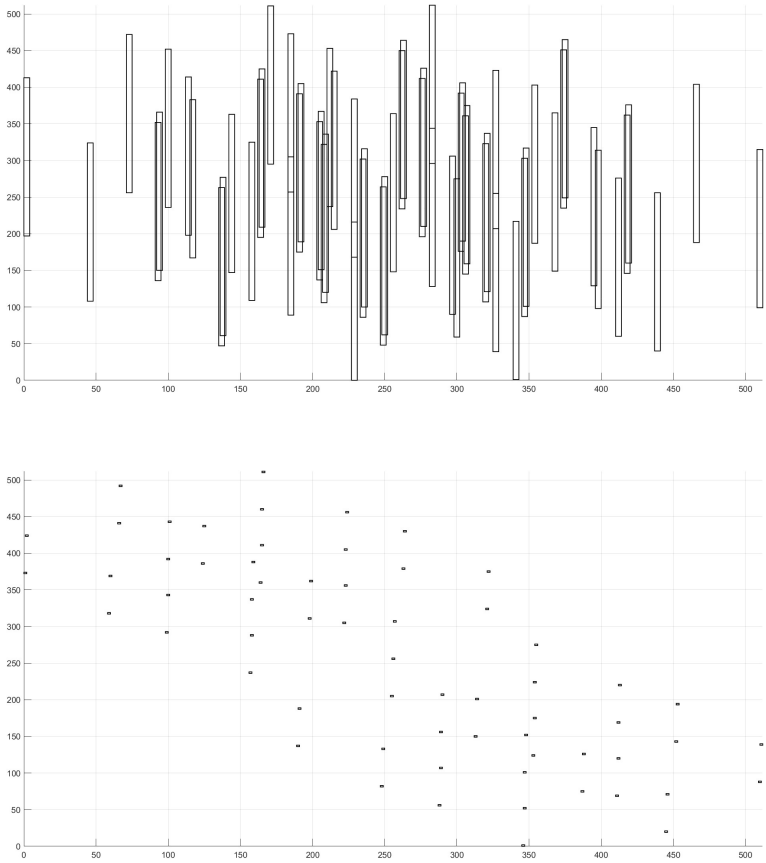


Figure 2.10: Top: The enlarging regions corresponding to each pixel of the weakly bad configuration $\mathcal{F}_{S_1}$, associated to the simple cycle S_1. Notice the various intersections among them. Bottom: The enlarging regions corresponding to each pixel of the weakly bad configuration $\mathcal{F}_{S_2}$, associated to the simple cycle S_2. In this case, the enlarging regions are pairwise disjoint sets.

	Phantom (a)		Phantom (b)		Phantom (c)		Phantom (d)	
	% reconstruction		% reconstruction		% reconstruction		% reconstruction	
♯ iterations	BRA	e-BRA	BRA	e-BRA	BRA	e-BRA	BRA	e-BRA
5	87,4092	98,1358	91,2003	96,8838	96,5935	99,1104	95,3346	98,4478
10	89,4630	99,6361	93,1129	98,3227	98,0015	99,6189	96,9589	99,4427
15	90,9096	99,8901	93,9625	98,2174	98,8770	99,7963	97,7921	99,6552
20	92,9287	99,9481	94,5007	98,9006	99,1268	99,9184	98,2510	99,8047
25	93,1965	99,9954	94,8883	99,4236	99,1817	99,9626	98,5619	99,9050
30	93,1419	100	94,9764	99,7509	99,2161	99,9821	98,7358	99,9374
35	93,1881	100	95,0813	99,9557	99,2851	99,9931	98,8308	99,9493
40	93,2270	100	95,1820	99,9916	99,3397	99,9966	98,8640	99,9538
45	93,2526	100	95,2492	99,9992	99,4194	99,9985	98,9540	99,9619
50	93,3460	100	95,4098	100	99,4907	99,9996	99,0292	99,9680
55	93,4811	100	95,6688	100	99,5422	100	99,1077	99,9725
65	93,9461	100	96,2460	100	99,6094	100	99,1680	99,9844
75	94,5168	100	96,7083	100	99,6449	100	99,2073	99,9950
85	95,1431	100	97,2759	100	99,6510	100	99,2123	99,9989
95	95,7157	100	97,7436	100	99,6544	100	99,1940	99,9992
105	96,2372	100	98,2449	100	99,6601	100	99,2180	100

Table 2.3: The table shows the percentage of pixels that were correctly reconstructed by e-BRA, w.r.t. the number of iterations of CGLS, by choosing the cycle of uniqueness S_1. For each phantom, the performance of the new algorithm is compared with the results reached with BRA.

	Phantom (a)		Phantom (b)		Phantom (c)		Phantom (d)	
	% reconstruction		% reconstruction		% reconstruction		% reconstruction	
♯ iterations	BRA	e-BRA	BRA	e-BRA	BRA	e-BRA	BRA	e-BRA
1	72,0032	99,8955	67,5571	97,2942	80,6828	87,7766	77,6054	83,5976
2	78,6480	99,9985	79,6726	98,7350	88,6410	97,7989	85,7597	96,5641
3	85,1456	100	87,1723	99,7883	93,3338	99,5522	91,3975	98,7865
4	87,4401	100	91,1255	99,9462	95,8172	99,9256	94,3275	99,4396
5	87,4092	100	91,2003	99,9905	96,5935	99,9920	95,3346	99,7246
6	87,9543	100	91,7496	100	96,9959	100	95,9316	99,8760
7	88,5426	100	92,1745	100	97,2382	100	96,2200	99,9241
8	89,0572	100	92,6483	100	97,5616	100	96,5168	99,9439
9	89,3127	100	92,9394	100	97,8333	100	96,7274	99,9638
10	89,4630	100	93,1129	100	98,0015	100	96,9589	99,9840
11	89,5687	100	93,2533	100	98,1678	100	97,1081	99,9924
12	89,8548	100	93,4521	100	98,3734	100	97,2919	99,9962
13	90,1440	100	93,6581	100	98,5573	100	97,4491	99,9985
14	90,5163	100	93,8484	100	98,7583	100	97,6395	100

Table 2.4: The table shows the percentage of pixels that were correctly reconstructed by e-BRA, w.r.t. the number of iterations of CGLS, by choosing the cycle of uniqueness S_2. For each phantom, the performance of the new algorithm is compared with the results reached with BRA.

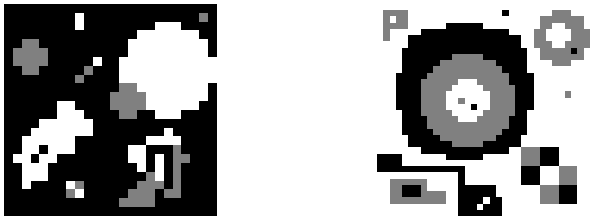
	Phantom (a)		Phantom (b)		Phantom (c)		Phantom (d)	
	♯ wrong pixel		♯ wrong pixel		♯ wrong pixel		♯ wrong pixel	
♯ iterations	BRA	e-BRA	BRA	e-BRA	BRA	e-BRA	BRA	e-BRA
5	33006	4887	23068	8169	8930	2332	12230	4069
10	27622	954	18054	4397	5239	999	7972	1461
15	23830	288	15827	4673	2944	534	5788	904
20	18537	136	14416	2882	2289	214	4585	512
25	17835	12	13400	1511	2145	98	3770	249
30	17978	0	13169	653	2055	47	3314	164
35	17857	0	12894	116	1874	18	3065	133
40	17755	0	12630	22	1731	9	2978	121
45	17688	0	12454	2	1522	4	2742	100
50	17443	0	12033	0	1335	1	2545	84
55	17089	0	11354	0	1200	0	2339	72
65	15870	0	9841	0	1024	0	2181	41
75	14374	0	8629	0	931	0	2078	13
85	12732	0	7141	0	915	0	2065	3
95	11231	0	5915	0	906	0	2113	2
105	9864	0	4601	0	891	0	2050	0

Table 2.5: The table shows the number of wrong pixels in the reconstruction of the binary image when choosing the cycle of uniqueness S_1, w.r.t. the number of iterations selected for CGLS. The performance is also compared with the algorithm BRA.

	Phantom (a)		Phantom (b)		Phantom (c)		Phantom (d)	
	♯ wrong pixel		♯ wrong pixel		♯ wrong pixel		♯ wrong pixel	
♯ iterations	BRA	e-BRA	BRA	e-BRA	BRA	e-BRA	BRA	e-BRA
1	73392	274	85047	7093	50639	32043	58706	42998
2	55973	4	53287	3316	29777	5770	37330	9007
3	38940	0	33627	555	17475	1174	22551	3181
4	32925	0	23264	141	10965	195	14870	1469
5	33006	0	23068	25	8930	21	12230	722
6	31577	0	21628	0	7875	0	10665	325
7	30035	0	20514	0	7240	0	9909	199
8	28686	0	19272	0	6392	0	9131	147
9	28016	0	18509	0	5680	0	8579	95
10	27622	0	18054	0	5239	0	7972	42
11	27345	0	17686	0	4803	0	7581	20
12	26595	0	17165	0	4264	0	7099	10
13	25837	0	16625	0	3782	0	6687	4
14	24861	0	16126	0	3255	0	6188	0

Table 2.6: The table shows the number of wrong pixels in the reconstruction of the binary image when choosing the cycle of uniqueness S_2, w.r.t. the number of iterations selected for CGLS. The performance is also compared with the algorithm BRA.

solution of the same tomographic problem. Starting from this remark, we can start again from the central solution $\mathbf{x}^*$, that can be seen as a linear combination of elements of H_q^N as well, and then slightly change the coefficients of such a linear combination to get the corresponding gray-scale solution. As already mentioned, we do not have a theoretical result supporting our strategy, but we tried some experiments for the reconstruction of some images on three gray levels, using a modified version of e-BRA. The algorithm we propose consists of the computation of the central solution $\mathbf{x}^*$, using again the CGLS, and then in its rounding to the closest integer.
We performed the described algorithm on the two phantoms depicted in Figure 2.11, both images on $l = 3$ gray levels. The first one, on the left, has size 24×24 pixels, while the second one, on the right, has size 34×34 pixels.



(a) (b)

Figure 2.11: The images with 3 gray levels used to test the sets of directions. The phantom (a) has size 24×24 pixels, while the phantom (b) has size 34×34 pixels.

We selected the following sets of directions, that are of uniqueness for images on $l = 3$ gray levels in the respective grids,

$$S_{24\times 24} = \{(1,2),(1,-1),(1,3),(1,-2),(1,8),(1,-7)\},$$
$$S_{34\times 34} = \{(1,1),(1,-2),(3,4),(5,3),(1,6),(1,7),(3,-10)\}.$$

In both cases, a perfect reconstruction is achieved, as showed in Tables 2.7 and 2.8 ([3]).

Some final remarks:

- ▷ In general, the rounding of the central solution could provide an image on l' gray levels, with l' different from the number of gray levels that we assume to know as prior information.
- ▷ If we choose sets of directions that are not of uniqueness, and perform the heuristic on the same phantoms, a perfect reconstruction is not achieved (see [3]). This enforces us to go on in this line of research for the development of a correct reconstruction strategy, supported by theoretical results.
- ▷ On the other hand, we performed experiments on phantoms of very small size, so the achievement of the perfect reconstruction could be due to the fact that few solutions of the analyzed tomographic problem exist (even only one).
- ▷ When increasing the number of gray levels, sets of uniqueness require a higher number of directions, and are not easy to compute. As pointed out for the binary

case, we need to find an efficient method for the construction of such sets, even when $l \geq 3$.

♯iter	% reconstruction	♯ wrong pixels
10	$84,03$	92
20	$87,67$	71
30	$89,93$	58
40	$91,15$	51
50	$91,67$	48
100	$93,18$	22
150	$99,31$	4
200	100	0

Table 2.7: Results of the test made on the 3 gray-scale phantom of size 24×24, in terms of percentage of good reconstruction and number of wrong pixels w.r.t. the iterations employed in the CGLS subroutine.

♯iter	% reconstruction	♯ wrong pixels
10	$85,73$	165
20	$89,27$	124
30	$90,40$	111
50	$92,82$	83
100	$96,45$	41
150	$98,36$	19
200	$99,57$	5
250	$99,91$	1
300	100	0

Table 2.8: Results of the test made on the 3 gray-scale phantom of size 34×34, in terms of percentage of good reconstruction and number of wrong pixels w.r.t. the iterations employed in the CGLS subroutine.

Conclusion and final remarks

As a conclusion of the chapter, we propose a short list of possible research lines that arise from our work.

- ▷ The notion of simple cycle can be extended to higher dimension, moving from $\mathbb{Z}^2$ to $\mathbb{Z}^n$, for any $n \geq 3$. Consequently, the uniqueness results could be easily generalized by replacing the notion of independence with the notion of simple cycle.

  Problem 1. Exploit the notion of simple cycle to find uniqueness results in higher dimensions.

- ▷ An algorithm for the generation of simple cycles of uniqueness, for a grid of given size, does not exist yet. A research line in this direction consists in the development

of some strategy for the construction of sets of directions in an iterative way. This would avoid the computation of the whole set of forbidden translations, that is exponential in the cardinality of the set of directions, and so impracticable.

Problem 2. Develop a quick algorithm for the construction of simple cycles of uniqueness.

▷ The complete characterization of a family of sets of uniqueness for gray-scale images is still missing. A research line in this direction would improve also the generation of this type of sets, in view of practical applications.

Problem 3. Extend and generalize the notion of simple cycle of uniqueness to gray-scale images.

▷ The computational cost of the algorithm e-BRA, as well as its general performance, can be improved using algorithms based on integer linear programming techniques.

Problem 4. Explore different subroutines for the computation of $\tilde{\mathbf{x}}$ in e-BRA.

▷ A theoretical result supporting a reconstruction strategy for gray-scale images is still missing. However, the experimental results we achieved through the heuristic that generalizes e-BRA are very promising, and encourage a research line in the direction that follows a generalization of Theorem 2.3.11.

Problem 5. Find a theoretical result for gray-scale images, like an analogous to the rounding theorem given for the binary case.

▷ In this thesis, we considered parallel X-rays only, but the achieved uniqueness results could be explored and extended to other cases, for example when the acquisition of data is made through *fan-beam* or *cone-beam*, especially in view of practical application.

Problem 6. Investigate uniqueness in case of different models and different techniques of acquisition of data.

Chapter 3

Reconstruction of uniform hypergraphs

In this chapter, we consider the problem of characterizing the integer sequences that are the degree sequences of some uniform hypergraphs, as well as the reconstruction of one of the related realizations in case of positive answer. Both problems are NP-complete in general, so we focus on specific subclasses of integer sequences, aiming at defining polynomial time algorithms for their characterization and reconstruction.
We first consider the so-called *pattern sequences*, associated to 3-uniform hypergraphs; then, we generalize this class of degree sequences introducing a link with Order Theory and a connection with the ideals of a specific poset on triplets of numbers; finally, we use randomized approaches for the study of the more general case of k-uniform hypergraphs. We conclude the chapter by considering a variant of the reconstruction problem on graphs, starting from the sequence of co-degrees instead of degrees, with a particular emphasis on trees.
The results described in this chapter are joint work with A. Frosini in [5], Section 3.2, A. Frosini, E. Pergola and S. Rinaldi in [9], Section 3.3, A. Frosini, E. Pergola, S. Rinaldi and L. Vuillon in [10], Section 3.3, M. Lienau, M. Schulte and A. Taraz in [12], Section 3.4.

3.1 Introduction and preliminaries

In this section we give the motivations of our work together with the notations and the mathematical tools we are going to use.

Graphs and hypergraphs

Graphs are a useful tool in discrete mathematics for modeling relationships between pairs of objects. We briefly recall their definition and main properties. For a complete overview, we address the reader to [62].
An (undirected) graph $G = (V, E)$ consists of a set of vertices $V = \{v_1, \ldots, v_n\}$, also called *nodes*, and a (multi)set of edges $E = \{e_1, \ldots, e_m\}$, where an edge is defined as a (multi)set of two elements from V. So, an edge connects a pair of vertices, that are said to

be *adjacent*. A *loop* is an edge of the form $e = \{v_i, v_i\}$, for some $i = 1, \ldots, n$, while *multiple* (or *parallel*) edges are edges consisting of the same pair of vertices. A graph is *simple* if neither loops nor parallel edges occur. Figure 3.1(a) shows a simple graph. Given a vertex $v \in V$, the set of its *neighbors* in the graph G is $N_G(v) = \{u \in V \text{ s.t. } \{u, v\} \in E\}$, i.e., the set of all the vertices that are adjacent to v. A *path* in a graph G is a sequence of edges that join a sequence of distinct vertices. A graph is *connected* if for any pair of vertices (u, v) there exists at least one path leading from u to v. Finally, a *cycle* is defined as a non-empty path in which only the first and last vertices are equal, and a graph is called *acyclic* if it does not contain any cycle. The graph in Figure 3.1(a) is connected and contains a cycle. We finally recall that two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic* if there exists a bijection $\phi : V_1 \to V_2$ between the sets of vertices such that $u, v \in V_1$ are adjacent in G_1 if and only if $\phi(u), \phi(v) \in V_2$ are adjacent in G_2. In fact, G_1 and G_2 have the same structure and properties.

As a matter of fact, more complex situations may require more complex relationships among elements. This leads to the definition of *hypergraphs*, structures that generalize the concept of graph. Formally, a hypergraph is a pair of sets $H = (V, E)$, where again $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$ are the sets of vertices and hyperedges, briefly edges, respectively, but where a hyperedge e_i is defined as a non-empty (multi)set of elements of V, and can have any cardinality grater than one. In general, a hyperedge may contain more copies of the same vertex (*loop*), and the set E may contain more copies of the same hyperedge (*multiple edges*). In our work, we require the hypergraph to be *simple*, meaning that neither loops nor multiple edges occur.

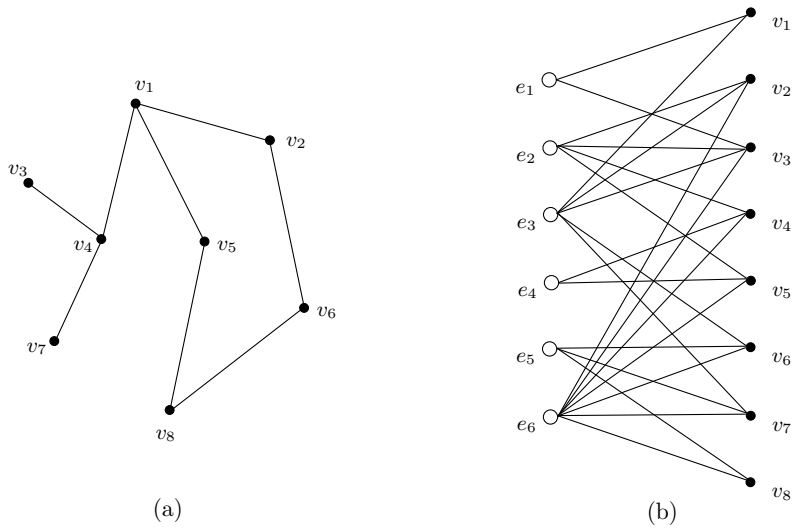


Figure 3.1: On the left, a simple graph. The graph is connected and contains a cycle of length five, $\{v_1, v_2, v_6, v_8, v_5, v_1\}$. On the right, a simple hypergraph represented as a bipartite graph: black nodes point out vertices, white nodes hyperedges. Each white node is connected to the black nodes of the vertices that constitute the corresponding edge.

Hypergraphs can be represented in different ways. In Fig. 3.1(b), the reader can find an example of a visual representation of a hypergraph as a bipartite graph, where black

points denote vertices, white points denote hyperedges, and vertices are connected to the hyperedges they belong to. However, in this thesis we make use of a different representation, describing a hypergraph through its incidence matrix. We recall here its definition.

Definition 3.1.1. Given a simple hypergraph $H = (V, E)$, with $V = \{v_1, \dots, v_n\}$ and $E = \{e_1, \dots, e_m\}$, its *incidence matrix*, also denoted by $H = (h_{ij})_{1 \le i \le m, 1 \le j \le n}$, is an $m \times n$ binary matrix defined as

$$h_{ij} = \begin{cases} 1 & \text{if } v_j \in e_i \\ 0 & \text{otherwise.} \end{cases}$$

Basically, each column of the incidence matrix represents a vertex of the hypergraph, while rows define the edges: the set of elements equal to 1 in the i-th row corresponds to the set of vertices that belong to the edge e_i. Notice that, since we require the hypergraph H to be simple, its incidence matrix has all distinct rows by definition (no repeated edges). We also underline the analogy between the matrix H and the grid $\mathcal{A}$ of size $m \times n$ described in Chapter 2. Following these similarities, we can consider the horizontal and vertical projections of H, i.e., the row and column sums. In case of rows, we get a vector of projections κ that collects the cardinalities of the edges. If this value is constant, for some $k \ge 2$, the hypergraph is said to be *k-uniform*. From now on, we will deal with k-uniform hypergraphs only. Notice that the case $k = 2$ corresponds to graphs. On the other hand, the i-th column sum collects the number of occurrences of the vertex v_i in the edges, for $i = 1, \dots, n$. This value is addressed as the *degree* of the vertex v_i, while the vector of all the degrees, $\pi = (d_1, \dots, d_n)$, is called the *degree sequence* of the hypergraph H. The hypergraph H is also called a *realization* of π. Without loss of generality, we consider the degree sequence arranged in weakly decreasing order, $d_1 \ge \dots \ge d_n > 0$. All the degrees are strictly greater than zero, since isolated vertices can be neglected. Finally, we introduce the notation $\sigma = \sum_{i=1}^{n} d_i$. Notice that in case of k-uniform hypergraph we have $m = \frac{\sigma}{k}$.

Example 3.1.2. The incidence matrix of the simple hypergraph depicted in Fig. 3.1(b) is

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Its vertical and horizontal projections are, respectively,

$$\pi = (4, 3, 3, 3, 3, 3, 2, 2),$$
$$\kappa = (7, 5, 4, 3, 2, 2).$$

The entries in κ are not constant, indeed H is not uniform.

We are now ready to define two problems, that clearly set a bridge between Discrete Tomography and Graph Theory.

Consistency: given an integer sequence π, does there exist a simple k-uniform hypergraph having π as its degree sequence? In case of positive answer, π is said to be *k-graphic*.

Reconstruction: given an integer sequence π that is k-graphic, find a simple hypergraph realizing π.

From the point of view of Discrete Tomography, the reconstruction problem is equivalent to the reconstruction of a binary image starting from its horizontal and vertical projections, $\kappa = (k, \ldots, k)$ and $\pi = (d_1, \ldots, d_n)$, respectively, where all the rows of the solution are required to be distinct.

The complexity of reconstructing hypergraphs

Both the consistency and reconstruction problems are well-known in literature. Their complexity drastically changes depending on the uniformity value k. In case of graphs, so $k = 2$, the two problems can be quickly solved using polynomial time strategies. The first characterization of graphic sequences dates back to 1960:

Theorem 3.1.3 (Erdős and Gallai [38]). Given a non-increasing sequence $\pi = (d_1, \ldots, d_n)$ of positive integers, there exists a simple graph realizing π if and only if

i) $\sum_{i=1}^{n} d_i$ is even, and

ii) $\sum_{i=1}^{r} d_i \leq r(r-1) + \sum_{i=r+1}^{n} \min\{r, d_i\}$, for all $r = 1, \ldots, n$.

In the following years, several other equivalent characterizations were provided (see [59] for a survey), as well as deterministic algorithms providing a realization for graphic sequences in polynomial time ([49, 50]).
Moving to hypergraphs, i.e. $k \geq 3$, few results have been achieved. Regarding the characterization of k-graphic sequences, a first non-constructive characterization is the following theorem, proven in 1975:

Theorem 3.1.4 (Dewdney [36]). A non-increasing sequence of positive integers $\pi = (d_1, \ldots, d_n)$ is k-graphic if and only if there exists a sequence $\pi' = (d'_2, \ldots, d'_n)$ such that

i) π' is $(k-1)$-graphic,

ii) $\sum_{i=2}^{n} d'_i = (k-1)d_1$,

iii) $\pi'' = (d_2 - d'_2, d_3 - d'_3, \ldots, d_n - d'_n)$ is k-graphic.

The key idea is to split the integer sequence π in two sequences, π' and π'', that are smaller in terms of entries, and then proceed recursively until a basic degree sequence that is known to be k-graphic is reached. The computationally hard part lies in the fact that the theorem does not give any hint about the form of π' and π'', so all the

possible splittings have to be considered to detect the graphicality of π, finally getting an algorithm that works in exponential time.
The second milestone result in the field is due to Deza et al., who in 2018 finally established the complexity of the characterization and reconstruction problems when moving from graphs to hypergraphs:

Theorem 3.1.5 (Deza et al. [33]). Given a non-increasing sequence $\pi = (d_1, \dots, d_n)$ of positive integers, it is NP-complete to decide if π is the degree sequence of a 3-uniform hypergraph.

Since the problem is NP-complete, even for the simple case $k = 3$, results concerning special classes of degree sequences that can be characterized or reconstructed in polynomial time acquire relevance. Among them, we recall regular or almost-regular sequences ([44]), results concerning sufficient ([16, 23]) or necessary ([17, 31]) conditions for a sequence to be k-graphic, as well as polynomial time reconstruction strategies for particular instances, mainly based on greedy techniques ([8, 9, 43]).

Hypergraphs and Order Theory

We generalize a construction described in the NP-completeness proof in [33], in order to introduce a class $\mathcal{D}$ of 3-uniform hypergraphs that are characterized by a strong structure in their incidence matrix. We briefly recall the definition of such a class, that will be studied in Section 3.2 together with the characterization and reconstruction of its subclass of pattern sequences. The following notions are borrowed from [8], a joint work with A. Frosini, W.L. Kocay and L. Tarsissi preceding the work in this thesis.

Definition 3.1.6. Given a non-increasing sequence $s = (s(1), \dots, s(n))$ of both positive and negative integers, the 3-uniform hypergraph H_s is defined through a bijection between the set of vertices $\{v_1, \dots, v_n\}$ and the entries of s. The construction of its incidence matrix is as follows: let us consider all the possible triplets (i, j, k) such that $1 \leq i < j < k \leq n$. Then, the vertices v_i, v_j and v_k form an edge of H_s if and only if $s(i)+s(j)+s(k) > 0$. We denote π_s the degree sequence of H_s, and define $\mathcal{D}$ the set of all the degree sequences that can be obtained through this procedure, for all $n \geq 3$.

Example 3.1.7. If we choose $s = (5, 2, 1, 0, -1, -6)$, we get the 3-uniform hypergraph

$$H_s = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Its degree sequence is $\pi = (7, 7, 5, 5, 5, 1)$. As an example, the 6-th row of the incidence matrix represents the edge (v_1, v_3, v_5), that is part of the hypergraph since $s(1) + s(3) + s(5) = 5 + 1 - 1 > 0$.

Three main results about the sequences in $\mathcal{D}$ are the following:

Property 3.1.8 ([8]). Given a sequence $\pi_s \in \mathcal{D}$ obtained from $s = (s(1), \ldots, s(n))$, and H_s the corresponding hypergraph, if (v_i, v_j, v_k) is an edge of H_s, then also $(v_i, v_j, v_{k'})$ is an edge of H_s, for all $j + 1 \leq k' \leq k$.

Proposition 3.1.9 ([8]). Given a degree sequence $\pi_s = (d_1, \ldots, d_n) \in \mathcal{D}$ obtained from an integer sequence s, if there exists an index $1 \leq i < n$ such that $d_i = d_{i+1}$, then there exists an integer sequence s' such that $\pi_s = \pi_{s'}$ and $s'(i) = s'(i + 1)$.

Theorem 3.1.10 ([8]). If $\pi \in \mathcal{D}$, then there exists a unique hypergraph (up to isomorphism) having π as its degree sequence.

Property 3.1.8 directly comes out from the construction of H_s starting from s, as well as Proposition 3.1.9, while Theorem 3.1.10 suggests the possibility of achieving a quick reconstruction for the sequences in $\mathcal{D}$, since no ambiguities arise in the set of possible solutions.

We now recall some basic notions of Order Theory (see [61] for a survey), and define a partially ordered set, briefly poset, that allows us to find a link between a special class of 3-uniform hypergraphs and its ideals. The obtained class includes $\mathcal{D}$, and will be studied in Section 3.3.
So, given a set Ω, a *partial order* on Ω is a binary relation $\preceq$ such that, for any $x, y, z \in \Omega$, we have

i) $x \preceq x$ (reflexivity),

ii) $x \preceq y$ and $y \preceq x$ imply $x = y$ (antisymmetry),

iii) $x \preceq y$ and $y \preceq z$ imply $x \preceq z$ (transitivity).

A *poset* is a set equipped with an order relation, $\mathcal{P} = (\Omega, \preceq)$. Notice that the relation $\preceq$ is a partial order in general, meaning that two elements in Ω could be *not comparable*. A set of all non-comparable elements is called an *antichain*. An *ideal* is a subset $I \subseteq \Omega$ such that, whenever $x \in I$, $y \in \Omega$ and $y \preceq x$, we have $y \in I$. For this reason, ideals are also called *down-sets*. Given $x \in \Omega$, the *principal ideal with generator* x is defined as

$$\downarrow\{x\} := \{y \in \Omega \text{ s.t. } y \preceq x\}.$$

Given a finite ideal I, it is always possible to write I as the union of finite principle ideals, $I = \downarrow\{x_1, \ldots, x_m\} = \downarrow\{x_1\} \cup \cdots \cup \downarrow\{x_m\}$ for some $x_1, \ldots, x_m \in \Omega$. In general, the antichain of its maximal elements uniquely identifies and generates an ideal I, where $m \in I$ is *maximal* if there is no $m' \in I$ such that $m \preceq m'$ and $m' \neq m$.

In the context of 3-uniform hypergraphs, we are interested in the definition of the poset of triplets $\mathcal{T}_n$. Given an integer $n \geq 3$, we define Ω_n as the set of all the triplets (a_1, a_2, a_3)

such that $a_i \in \{1, \ldots, n\}$ and $1 \leq a_1 < a_2 < a_3 \leq n$, and $\preceq$ the order relation that extends the natural order on triplets, that is,

$$(a_1, a_2, a_3) \preceq (b_1, b_2, b_3) \text{ if and only if } a_i \leq b_i \text{ for } i = 1, 2, 3.$$

The poset of triplets consists of $\mathcal{T}_n = (\Omega_n, \preceq)$.

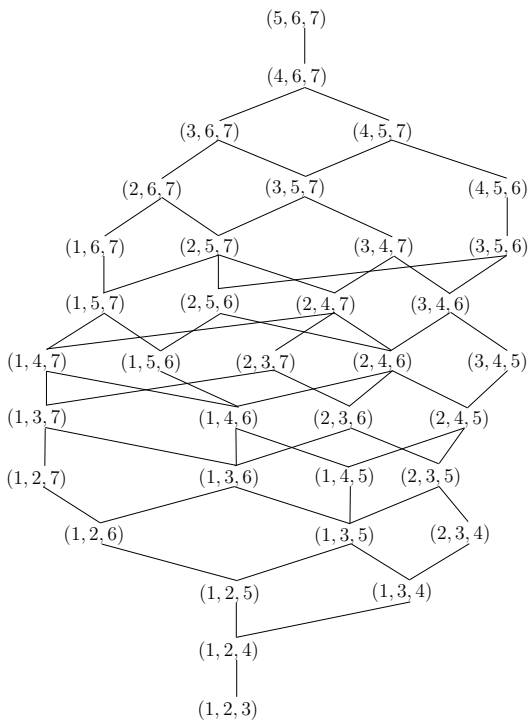


Figure 3.2: The complete poset of triplets $\mathcal{T}_7$.

It is clear that if we consider a set of n vertices, each element of Ω_n is in bijection with an edge of the complete, simple 3-uniform hypergraph. Indeed, the triplet (a_1, a_2, a_3) corresponds to the hyperedge $(v_{a_1}, v_{a_2}, v_{a_3})$. Now, let $\mathcal{I}_n$ be the set of the ideals of $\mathcal{T}_n$. By Property 3.1.8, each hypergraph in the class $\mathcal{D}$ corresponds to an ideal of $\mathcal{T}_n$, but the reverse is not true. Then, we are able to extend the class $\mathcal{D}$ by considering the set of degree sequences associated to the 3-uniform hypergraphs in $\mathcal{I}_n$. Section 3.3 is dedicated to their investigation, with particular emphasis on principal ideals and those ones realized by an antichain of two elements.

Example 3.1.11. The hypergraph H_s described in Example 3.1.7 corresponds in $\mathcal{I}_6$ to the ideal $I = \downarrow\{(1, 2, 6), (2, 4, 5)\}$, depicted in Figure 3.3.
On the other hand, the degree sequence $\pi = (25, 19, 17, 16, 12, 11, 9, 8, 6)$ in not in the class $\mathcal{D}$, even if it is associated to an element of $\mathcal{I}_9$. Indeed, there exist two non-isomorphic hypergraphs realizing π,

$$I_1 = \downarrow\{(1, 6, 9), (2, 3, 9), (2, 5, 7), (3, 4, 8)\},$$
$$I_2 = \downarrow\{(1, 5, 9), (1, 7, 8), (2, 4, 9), (3, 4, 7), (3, 5, 6)\},$$

in contradiction with Theorem 3.1.10.

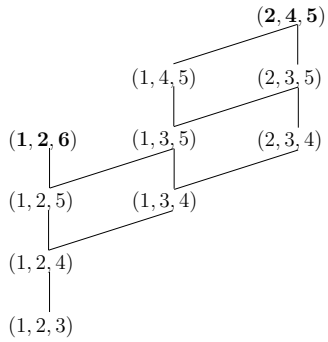


Figure 3.3: The ideal in $\mathcal{I}_6$ corresponding to the (unique) 3-uniform hypergraph having degree sequence $\pi = (7, 7, 5, 5, 5, 1)$. In this case, the down-set has two generators, in boldface.

Co-degrees and trees

The last part of this chapter is devoted to the study of a variation of the reconstruction problem we defined for hypergraphs, that is, the reconstruction of a simple graph starting from its co-degree sequence. We introduce here some preliminary notions and the notations.
Given a simple graph $G = (V, E)$ and two distinct vertices $u, v \in V$, the *co-degree* associated to the pair $\{u, v\}$ is the number of their common neighbors in the graph G. Formally, $c_{u,v} = |N_G(v) \cap N_G(u)|$. Given a graph on n vertices, the *co-degree sequence* γ is the list of all the co-degrees associated to all the possible pairs of distinct vertices, arranged in non-increasing order. Its length is $\binom{n}{2}$, and it can contain null entries, in general, if some vertices do not share any neighbor. See Figure 3.4 for an example.

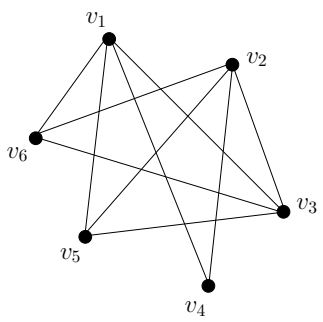


Figure 3.4: A graph with co-degree sequence $\gamma = (4, 3, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 0, 0)$. Its length is equal to $15 = \binom{6}{2}$, since the graph is defined on a set of six vertices. We remark that γ contains some null elements: as an example, $c_{2,4} = 0$, since v_2 and v_4 do not share any neighbor.

We define a variant of the consistency and reconstruction problems.

Problem: given an integer sequence γ of length $\binom{n}{2}$, for some $n \geq 2$, establish if there exists a simple graph on n vertices having γ as its co-degree sequence and, in case of positive answer, provide such a graph.

Up to our knowledge, the problem has not been investigated yet. We focus our attention on binary sequences, so we consider graphs in which any pair of vertices share at most one neighbor. An example of objects satisfying this property is the class of trees.
In Graph Theory, a *tree* T is a connected and acyclic graph. Among many properties of trees, we recall that any two vertices of T can be connected by a unique path, and that if T is defined on n vertices, then it has exactly $n-1$ edges. We denote by Υ_n the set of all the trees defined on n vertices.

Property 3.1.12. In a tree, every two vertices have at most one common neighbor.

The property trivially follows from the fact that trees are acyclic graphs.
Given a tree T, we designate a vertex $r \in V$ as the *root* of T. Given a vertex v, its *parent* is the vertex adjacent to v on the path to the root, while v is said to be its *child*. Any vertex has a unique parent, but can have any number of children, in general. The only exception is the root, that does not have any parent. The vertices of degree 1 and different from the root are called *leaves*; clearly, a leaf does not have any child. Finally, we call *siblings* two (or more) vertices sharing the same parent.
The *height* l of a tree is the length of the longest path connecting the root to a leaf. The *level* i is defined as the set of vertices at distance i from the root. By convention, the root is at level 0. We denote b_i the number of vertices that belong to the i-th level or, equivalently, the total number of children of all the vertices at level $i-1$. Figure 3.5 summarizes all the introduced notation.
Finally, we recall that both the path and the star on n vertices belong to Υ_n. The first, is a tree of height $l = n-1$, with exactly one node at each level; the latter, is a tree of height $l = 2$ in which the root has exactly $n-1$ children, that are also leaves.

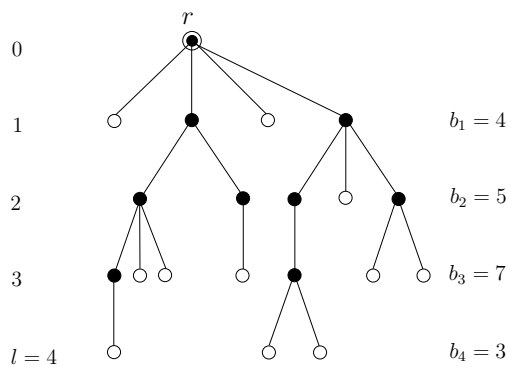


Figure 3.5: An example of tree of height $l = 4$. There are eleven leaves in total, represented by white nodes. The root is highlighted with a double circle.

We dedicate Section 3.5 of this chapter to the study of the co-degree sequences associated to trees.

3.2 Pattern sequences of 3-uniform hypergraphs

In this section, we approach the study of a special class of degree sequences of 3-uniform hypergraphs, addressed as *pattern sequences*. They constitute a subclass of $\mathcal{D}$, and are defined by choosing an integer sequence s satisfying certain peculiarities. We present a polynomial time algorithm to provide a quick solution both to the consistency and to the reconstruction problem.

3.2.1 Patterns with fixed step

Starting from the definition of the sequences in $\mathcal{D}$ given in Section 3.1, we now introduce the concept of pattern. Given a non-increasing integer sequence $s = (s(1), \dots, s(n))$, its *pattern* is an array $\mathbf{p} = (t_1, \dots, t_{n-1})$ that stores the differences between two consecutive values in s, that is, $t_i = s(i) - s(i+1)$ for all $i = 1, \dots, n-1$. Since s may contain both positive and negative elements, its pattern can be uniquely described as the concatenation of two arrays, $\mathbf{p} = (\mathbf{p}^+; \mathbf{p}^-)$, the first one storing the differences between non-negative elements in s, and the second one concerning the negative part of the integer sequence. We are interested in studying the properties of the degree sequences in $\mathcal{D}$ generated by sequences with a particular pattern, following the idea in [43]. First, we fix the negative part as $\mathbf{p}^- = (1, \dots, 1)$, and then investigate how the degree sequences in $\mathcal{D}$ vary in relation to $\mathbf{p}^+$. We call such elements *hypergraphic pattern sequences* (briefly, *pattern sequences*), and point out this class with $\mathcal{P} \subsetneq \mathcal{D}$.

Example 3.2.1. The integer sequence $s_1 = (5, 2, 1, 0, -1, -6)$ has pattern $\mathbf{p_1} = (3, 1, 1; 1, 5)$, and generates the degree sequence $\pi_{s_1} = (7, 7, 5, 5, 5, 1) \in \mathcal{D} \setminus \mathcal{P}$.
On the other hand, $s_2 = (4, 1, 0, -1, -2, -3)$ has pattern $\mathbf{p_2} = (3, 1; 1, 1, 1)$, and so generates a degree sequence in the class $\mathcal{P}$, $\pi_{s_2} = (8, 4, 4, 3, 3, 2)$.

Furthermore, as a preliminary analysis, we also fix a constant positive pattern as $\mathbf{p}^+ = (t, \dots, t)$, for some $t \geq 1$. We call t the *step* of $\mathbf{p}^+$, and $\mathcal{P}_t \subsetneq \mathcal{P}$ the set of pattern sequences with constant step equal to t.
The sequences in $\mathcal{P}_t$ can be iteratively constructed by increasing the length of their pattern. Formally, for a given step $t \geq 1$, we define

$$\begin{cases} s^{1,t} = (t, 0, -1, -2, \dots, -(t-1)), \\ s^{i,t} = (it, (i-1)t, (i-2)t, \dots, 0, -1, \dots, -(2(i-1)t - 1)) \quad \text{for all } i \geq 2, \end{cases}$$

and then $\pi^{i,t}$ the degree sequence obtained from $s^{i,t}$. We underline that the length of $\pi^{i,t}$ strictly depends on t and i, and, particularly, it increases together with the parameter i. Moreover, due to the regularity of $s^{i,t}$, it is possible to compute $\pi^{i,t}$ in linear time with respect to the length of the sequence.
Notice that if $t = 1$, the previous definition applies starting from $i = 2$, to avoid the construction of the empty hypergraph.

Remark 3.2.2. By definition, $s^{i,t}(1) + s^{i,t}(2) + s^{i,t}(n_{i,t}) = 1$ for all $i, t \geq 1$, with $n_{i,t}$ the length of the sequence. We fix this condition to avoid singleton vertices in the hypergraph $H_{s^{i,t}}$ or, equivalently, null elements in $\pi^{i,t}$.

Example 3.2.3. Choosing the step $t = 3$, the first integer sequences are

$$\begin{aligned}
s^{1,3} &= (3, 0, -1, -2)\\
s^{2,3} &= (6, 3, 0, -1, -2, -3, -4, -5, -6, -7, -8)\\
s^{3,3} &= (9, 6, 3, 0, -1, -2, -3, -4, -5, -6, -7, -8, -9, -10, -11, -12, -13, -14)\\
s^{4,3} &= (12, 9, 6, 3, 0, -1, -2, -3, -4, -5, -6, -7, -8, -9, -10, -11, -12, \dots, -20)\\
s^{5,3} &= (15, 12, 9, 6, 3, 0, -1, -2, -3, -4, -5, -6, -7, -8, -9, -10, -11, -12, \dots, -26).
\end{aligned}$$

The corresponding degree sequences are

$$\begin{aligned}
\pi^{1,3} &= (2, 2, 1, 1)\\
\pi^{2,3} &= (18, 11, 8, 6, 5, 4, 3, 2, 1, 1, 1)\\
\pi^{3,3} &= (48, 34, 24, 18, 15, 13, 11, 9, 8, 6, 5, 4, 2, 2, 2, 1, 1, 1)\\
\pi^{4,3} &= (93, 71, 54, 41, 32, 28, 25, 22, 19, 17, 15, 13, 11, 8, 7, 6, 4, 4, 4, 2, 2, 2, 1, 1, 1)\\
\pi^{5,3} &= (152, 123, 98, 78, 62, 50, 45, 41, 37, 33, 30, 27, 24, 22, 18, 16, 14, 11, 10, 9, 6, 6, 6, 4, 4, 4, 2, 2, 2, 1, 1, 1).
\end{aligned}$$

The following proposition summarizes the main properties of the sequences in $\mathcal{P}_t$:

Proposition 3.2.4 ([5]). Given a step $t \geq 1$ and the integer sequence $s^{i,t}$, for all $i \geq 1$ it holds:

i) $s^{i,t}(i+1) = 0$,

ii) $s^{i,t}(1) + s^{i,t}(j) + s^{i,t}(i + t(2i - j + 1)) = 1$, for all $j = 2, \dots, i+1$,

iii) $s^{i,t}(1) + s^{i,t}(j) + s^{i,t}(k) \leq 0$, for all $k \geq i + 1 + t(2i - j + 1)$,

iv) $n_{i,t} = (2t+1)i - t$.

All properties come after the definition of the sequences $s^{i,t}$.

Tails and their properties

Looking at the degree sequences generated in Example 3.2.3, we notice a regular behavior in their smaller entries: iteration by iteration, the values and positions, from right to left, of the last degrees are preserved in all the sequences; moreover, we can see that equal elements are repeated exactly 3 times, and that each time we increase the value i by one, a new value appears in position $\pi^{i,3}(n_{i,3} - 3(i-1))$. Surprisingly, as i increases, the different values of the last entries stabilize to the sequence $A002620$ in [63], whose first elements are $1, 2, 4, 6, 9, 12, 16, 20, 25$. We want to study such a behavior, so we formally introduce the notion of *tail* for the sequences in $\mathcal{P}_t$.
Given two integers $i, t \geq 1$, we define the tail $T^t(i)$ of the degree sequence $\pi^{i,t}$ as the array composed by its last $t(i-1)+1$ entries, namely,

$$T^t(i) = \left(\pi^{i,t}(n_{i,t} - t(i-1)), \dots, \pi^{i,t}(n_{i,t})\right).$$

Example 3.2.5. Going back to Example 3.2.3, the tails of the first pattern sequences in $\mathcal{P}_3$ are

$$\begin{aligned}
T^3(1) &= (1)\\
T^3(2) &= (2,1,1,1)\\
T^3(3) &= (4,2,2,2,1,1,1)\\
T^3(4) &= (6,4,4,4,2,2,2,1,1,1)\\
T^3(5) &= (9,6,6,6,4,4,4,2,2,2,1,1,1).
\end{aligned}$$

Notation. For all $i \geq 1$, we write e_{2i} as the sum of the first even numbers from 2 to $2i$ (included), and o_{2i-1} as the sum of the first odd numbers from 1 to $2i-1$ (included). We also introduce the power notation a^t to indicate the repetition of the element a for t times in an array.

In order to provide a fast reconstruction strategy, we prove that the regular behavior of tails holds for any fixed step $t \geq 1$, it is defined by a well-known numerical sequence ($A002620$ in [63]), and that these sequences tend to a fixed point as i tends to infinity.

Lemma 3.2.6 ([5])**.** For a fixed $t \geq 1$, let $T^t(i)$ be the tail of the i-th degree sequence $\pi^{i,t} \in \mathcal{P}_t$. Then, for all $i \geq 1$, it holds

$$\begin{aligned}
T^t(i) &= (o_i, e^t_{i-1}, o^t_{i-2}, e^t_{i-3}, \ldots, 2^t, 1^t) \quad \text{if } i \text{ is odd,}\\
T^t(i) &= (e_i, o^t_{i-1}, e^t_{i-2}, o^t_{i-3}, \ldots, 2^t, 1^t) \quad \text{otherwise.}
\end{aligned}$$

Proof. The proof is by induction on the index i.

Base case. We need a base case both for i even and odd.

i) If $i = 1$, the length of the tail is equal to 1, by definition. Moreover, by definition of $s^{1,t}$, the lowest degree of $\pi^{1,t}$ takes value 1, being $s^{1,t}(1) + s^{1,t}(2) + s^{1,t}(n_{1,t}) = 1$ the only positive sum of triplets involving $s^{1,t}(n_{1,t})$. Then, $T^t(1) = 1 = o_1$.

ii) If $i = 2$, the tail has length equal to $t+1$. By Proposition 3.2.4, *ii)* and *iii)*, we have

$$\begin{aligned}
s^{2,t}(1) + s^{2,t}(2) + s^{2,t}(j) &> 0 \quad \text{for all } 3 \leq j \leq n_{2,t},\\
s^{2,t}(1) + s^{2,t}(3) + s^{2,t}(j) &> 0 \quad \text{for all } 4 \leq j \leq n_{2,t} - t,\\
s^{2,t}(1) + s^{2,t}(4) + s^{2,t}(j) &\leq 0 \quad \text{for all } j \geq 5.
\end{aligned}$$

It follows that the last $t+1$ degrees of $\pi^{2,t}$ are $(2, 1 \ldots, 1)$, and consequently $T^t(2) = (e_2, o^t_1)$.

Induction step. We have to prove the statements for the generic iteration from i to $i+1$. Without loss of generality, we assume i to be even. In case of odd index, the proof proceeds similarly.

The tails $T^t(i)$ and $T^t(i+1)$ have lengths $t(i-1)+1$ and $ti+1$, respectively, and, by induction hypothesis, $T^t(i) = (e_i, o^t_{i-1}, e^t_{i-2}, \ldots, 2^t, 1^t)$. Then, to compute $T^t(i+1)$, we only need to compute which hyperedges are added to the hypergraph

$H_{s^{i,t}}$ to get $H_{s^{i+1,t}}$. By Proposition 3.2.4, $ii)$, the degrees of the last $it+1$ vertices of $\pi^{i+1,t}$ are

$$\delta = (i+1, i^t, (i-1)^t, \ldots, 2^t, 1^t).$$

Moreover, if we consider $\tilde{T}^t(i)$ the tail $T^t(i)$ up to its first t elements, we have by construction that

$$T^t(i+1) = \left(\tilde{T}^t(i), 0^{2t}\right) + \delta.$$

Then, by the induction hypothesis,

$$\begin{aligned} T^t(i+1) &= \left(o_{i-1} + (i+1), e^t_{i-2} + i^t, o^t_{i-3} + (i-1)^t, \ldots, e^t_2 + 4^t, o^t_1 + 3^t, 0^t + 2^t, 0^t + 1^t\right) \\ &= \left(o_{i+1}, e^t_i, o^t_{i-1}, \ldots, 2^t, 1^t\right), \end{aligned}$$

and so the thesis.

□

Example 3.2.7. For the sake of clarity, we show an example of the induction step presented in the proof of Lemma 3.2.6. We choose $t = 3$ and $i = 4$.
By induction hypothesis, we know the entries of the tail $T^3(4)$, i.e.,

$$T^3(4) = (e_4, o^3_3, e^3_2, o^3_1) = (6, 4, 4, 4, 2, 2, 2, 1, 1, 1).$$

If we remove the first $t = 3$ entries, we obtain the vector

$$\tilde{T}^3(4) = (4, 2, 2, 2, 1, 1, 1).$$

Now, the hyperedges added to $H_{s^{4,3}}$ to get $H_{s^{5,3}}$, and involving the last $it + 1 = 13$ vertices, $\{v_{20}, \ldots, v_{32}\}$, are the following

(v_1, v_2, v_{20})	(v_1, v_3, v_{20})	(v_1, v_4, v_{20})	(v_1, v_5, v_{20})	(v_1, v_6, v_{20})
(v_1, v_2, v_{21})	(v_1, v_3, v_{21})	(v_1, v_4, v_{21})	(v_1, v_5, v_{21})	
(v_1, v_2, v_{22})	(v_1, v_3, v_{22})	(v_1, v_4, v_{22})	(v_1, v_5, v_{22})	
(v_1, v_2, v_{23})	(v_1, v_3, v_{23})	(v_1, v_4, v_{23})	(v_1, v_5, v_{23})	
(v_1, v_2, v_{24})	(v_1, v_3, v_{24})	(v_1, v_4, v_{24})		
(v_1, v_2, v_{25})	(v_1, v_3, v_{25})	(v_1, v_4, v_{25})		
(v_1, v_2, v_{26})	(v_1, v_3, v_{26})	(v_1, v_4, v_{26})		
(v_1, v_2, v_{27})	(v_1, v_3, v_{27})			
(v_1, v_2, v_{28})	(v_1, v_3, v_{28})			
(v_1, v_2, v_{29})	(v_1, v_3, v_{29})			
(v_1, v_2, v_{30})				
(v_1, v_2, v_{31})				
(v_1, v_2, v_{32})				

and provide for the last entries of $\pi^{5,3}$ the degrees

$$\delta = (5, 4^3, 3^3, 2^3, 1^3) = (5, 4, 4, 4, 3, 3, 3, 2, 2, 2, 1, 1, 1).$$

To conclude the computation of the new tail, we need to sum $T^3(5) = (\tilde{T^3}(4), 0^6) + \delta$, that is,

$$\begin{aligned} T^3(5) &= (4,2,2,2,1,1,1,0,0,0,0,0,0) + (5,4,4,4,3,3,3,2,2,2,1,1,1) \\ &= (9,6,6,6,4,4,4,3,3,3,2,2,2,1,1,1) \\ &= (o_5, e_4^3, o_3^3, e_2^3, o_1^3). \end{aligned}$$

Remark 3.2.8. Notice that, in accordance to Examples 3.2.3 and 3.2.5, at each iteration a new value appears as first element of the tail.

Starting from Lemma 3.2.6, we are able to characterize the i-th tail $T^t(i)$ in terms of a unique numerical sequence $\{a_n\}_{n\geq 1}$, regardless the parity of the value i.

Theorem 3.2.9 ([5]). For a fixed step $t \geq 1$ and $i \geq 1$, the tail related to the i-th pattern sequence in $\mathcal{P}_t$ is

$$T^t(i) = \left(a_i, a_{i-1}^t, a_{i-2}^t \ldots, a_3^t, a_2^t, a_1^t\right),$$

with $\{a_n\}_{n\geq 1}$ the numerical sequence given by

$$a_n = \left\lceil \frac{n+1}{2} \right\rceil \cdot \left\lfloor \frac{n+1}{2} \right\rfloor \quad \text{for } n \geq 1.$$

Proof. If $n = 2j$ is even, then $a_n = j(j+1)$, i.e., it is the sum of the first j even numbers, e_{2j}. On the other hand, if $n = 2j+1$ is odd, then $a_n = (j+1)^2$, i.e., it is the sum of the first j odd numbers, o_{2j+1}. In both cases, the proof trivially follows from Lemma 3.2.6. □

Due to the very nice and regular behavior of tails, starting from $\{a_n\}_{n\geq 1}$ we can define, for a fixed $t \geq 1$, the degree sequence

$$\boldsymbol{\pi} = (\ldots, a_{i+1}^t, a_i^t, \ldots\ldots, a_3^t, a_2^t, a_1^t),$$

that can be seen as the *fixed point* of the class $\mathcal{P}_t$, namely, the limit of $\pi^{i,t}$ as i tends to infinity.
Being $\{a_n\}_{n\geq 1}$ the sequence $A002620$ in [63], our results confirm and generalize those ones provided by the authors in [43].

A reconstruction strategy for $\mathcal{P}_t$

Theorem 3.2.9 provides a complete characterization for the lower degrees of the sequences in $\mathcal{P}_t$, for all $t \geq 1$, as well as the definition of the fixed point $\boldsymbol{\pi}$. This result, together with the uniqueness property of the sequences in $\mathcal{D}$ (Theorem 3.1.10), allow to define a quick reconstruction strategy for the pattern sequences with fixed step. It consists in analyzing the lower entries of the input sequence, checking if they match with the elements of the numerical sequence $\{a_n\}_{n\geq 1}$ and with the tail $T^t(i)$, for some $i, t \geq 1$. In case of positive answer, the algorithm computes the integer sequence $s^{i,t}$ and then the corresponding degree sequence $\pi^{i,t}$. If this last coincides with the input, then the corresponding hypergraph $H_{s^{i,t}}$ can be reconstructed following the definition of the class

Algorithm 2. *PatternFix(π)*

Input: A non-increasing integer sequence π of length n

1 Compute t the number of 1s in π;
2 set $i = 2$ and $a_2 = 2$;
3 **while** $\pi(n-(i-1)t) = \cdots = \pi(n-(i-2)t+1) = a_i$ **do**
4 $\quad i = i + 1$;
5 $\quad a_i = \lceil \frac{i+1}{2} \rceil \lfloor \frac{i+1}{2} \rfloor$;
6 **end**
7 **if** $\pi(n-(i-1)t) = a_i$ **then**
8 $\quad$ compute the integer sequence $s^{i,t}$;
9 $\quad$ compute $\pi^{i,t}$ the degree sequence generated by $s^{i,t}$;
10 $\quad$ **if** $\pi^{i,t} = \pi$ **then**
11 $\quad\quad$ compute $H_{s^{i,t}}$;
12 $\quad\quad$ **return** success;
13 $\quad$ **else**
14 $\quad\quad$ **return** failure;
15 $\quad$ **end**
16 **else**
17 $\quad$ **return** failure;
18 **end**

Output: $H_{s^{i,t}}$

$\mathcal{D}$; otherwise, due to the uniqueness property, we can state that the input sequence is not in the class $\mathcal{P}_t$, for any $t \geq 1$, and so the algorithm returns *failure* as output. The described procedure, *PatternFix*, is detailed as a pseudo-code in Algorithm 2.
The algorithm clearly works in polynomial time. More precisely,

Theorem 3.2.10. *PatternFix* has a computational cost of $O(n^3)$ in the size of the input.

Proof. In the first phase, the while loop performs at most n iterations to check if there exists a tail $T^t(i)$, for some $i, t \geq 1$. In case of positive answer, the second phase of the algorithm performs the computation of the sequences $s^{i,t}$ and $\pi^{i,t}$, in linear time, and then the construction of the matrix $H_{s^{i,t}}$, that requires three nested loops, so a cost that is cubic in the length n of $s^{i,t}$. Summing up, the total computational cost is $O(n^3)$. □

Remark 3.2.11. Actually, the computational cost of *PatternFix* becomes linear in the size of the input if we only focus on the consistency problem, without providing the realization of the input sequence (in case of positive answer). It is sufficient to skip line 11 in Algorithm 2.

Example 3.2.12. We consider three examples of integer sequences as input of *PatternFix*.

$$\pi_1 = (73, 49, 33, 24, 21, 19, 16, 14, 12, 11, 9, 7, 6, 5, 4, 2, 2, 2, 2, 1, 1, 1, 1).$$

PatternFix immediately individuates the candidate step $t = 4$, and then the index $i = 3$ after the completion of the while loop. The condition $\pi_1(15) = a_3 = 4$, line 7, is also satisfied, so that the computation of $s^{3,4}$ and $\pi^{3,4}$ is carried on.

Since $\pi^{3,4} = \pi_1$, the algorithm successfully ends, providing the hypergraph $H_{s^{3,4}}$ as output.

$\pi_2 = (27, 20, 14, 11, 10, 8, 6, 5, 4, 2, 2, 1, 1)$.

PatternFix individuates the values $t = 2$ and $i = 3$, and then computes the sequences $s^{3,2}$ and $\pi^{3,2}$. Since $\pi^{3,2} \neq \pi_2$, the algorithm returns failure, so we can conclude that $\pi_2 \notin \mathcal{P}_t$ for any value $t \geq 1$.

$\pi_3 = (28, 28, 24, 21, 18, 15, 12, 11, 8, 5, 1)$.

PatternFix immediately fails, since no tail is present in the input sequence π_3.

3.2.2 Generic patterns

We now move to the more general case of the class $\mathcal{P}$, meaning that we allow any possible positive pattern $\mathbf{p}^+$ in our integer sequence s. To simplify our work, we add further hypothesis, resumed in the following formal definition.

Definition 3.2.13. A degree sequence $\pi \in \mathcal{D}$ is in $\mathcal{P}$ if and only if the integer sequence $s = (s(1), \ldots, s(n))$ realizing π satisfies the following conditions:

i) there exists an index $2 \leq i \leq n-1$ such that $s(i) = 0$,

ii) $s(i) - s(i+1) = 1$ for all $z \leq i \leq n-1$, with $z = \max\{1 < z < n \text{ s.t. } s(z) = 0\}$,

iii) $s(1) + s(2) + s(n) = 1$.

Basically, we still require the negative pattern to be $\mathbf{p}^- = (1, \ldots, 1)$; moreover, we assume that there exists at least one entry in s equal to 0, pointing out the rightmost one with the index z, and we also require $\mathbf{p}^-$ to be as long as possible, avoiding vertices with degree zero (condition *iii*)).

Example 3.2.14. Choosing $s = (9, 4, 2, 1, 0, 0, -1, -2, -3, \ldots, -12)$, we get the pattern sequence $\pi = (69, 38, 29, 25, 22, 22, 19, 16, 13, 10, 9, 7, 6, 5, 3, 2, 1, 1)$. The positive pattern is $\mathbf{p}^+ = (5, 2, 1, 1, 0)$, while the rightmost null element of s lies in position $z = 6$.

In case of sequences in $\mathcal{P} \setminus \mathcal{P}_t$, we lose the notion of tail (see Example 3.2.14), so that the strategy provided by *PatternFix* becomes useless for their detection. So, we need to define a different algorithm, trying to take advantage directly from the regularity of s instead of the degree sequence.

A reconstruction algorithm for pattern sequences

We propose the algorithm *PatternRec* that, given an integer sequence π as input, decides if it is a pattern sequence and, if so, reconstructs the related hypergraph H_π in polynomial time. The pseudo-code of *PatternRec* is given in Algorithm 3. Here we sketch the main steps of the procedure, whose correctness is guaranteed by Theorem 3.2.16.
As a prior information, we assume to know the value of the parameters z and $t_1 = s(1) - s(2)$. The latter may vary between 0 and $-\left\lfloor \frac{s(n)}{2} \right\rfloor$, by the structure of s, with

$s(n)$ that assumes value $-n+2$ at most, being n the length of the sequence. In practice, many parallel computations can be started, one for each value of the parameters, however keeping polynomial the complexity of the whole algorithm.
The main procedure consists in the following steps, that are iteratively repeated:

Step 1. Compute the number q of entries equal to 1 in π, and set $q = s(2) - s(3)$. The fact that q coincides with t_2 easily follows from the construction of the sequences in $\mathcal{D}$. Indeed, if a vertex v_k has degree 1, it means that there exists a unique pair of indices (i, j) such that $s(i) + s(j) + s(k) > 0$, with $1 \leq i < j < k \leq n$. Since s is non-increasing, the only possible choice is $(i, j) = (1, 2)$. All the other vertices $v_{k'}$ having degree strictly greater than one are such that $s(2) + s(3) + s(k') > 0$ also holds, since $s(2) + s(3)$ is the second greatest sum between elements of s. Then, $q = t_2$.

Step 2. By the knowledge of t_1, t_2, z and n, and the equation $s(1)+s(2)+s(n) = 1$ (see Definition 3.2.13), compute univocally the values $s(1), s(2), s(3)$ and $s(z), \ldots, s(n)$.

Step 3. For all the known values in s, compute $s(i)+s(j)+s(k)$ for $i < j < k$, and, if it is strictly greater than zero, insert the corresponding hyperedge in H. Then, update the input sequence π by subtracting the degree sequence of the added hyperedges.

Step 4. Compute q the number of vertices that reached degree equal to 0 after the previous update, and set $t_3 = s(3)-s(4) = q$. Then, compute $s(4)$ by the knowledge of $s(3)$ and t_3. The fact that $q = t_3$ follows from the same reasoning described in point 1.

Step 5. Repeat Step 3 and Step 4 until all the entries of s have been detected.

If the final update of the input sequence is the null vector, then H realizes π and is provided as output. Otherwise the algorithm fails, meaning that $\pi \notin \mathcal{P}$.

Example 3.2.15. We detail, as an example, the reconstruction of the sequence

$$\pi = (69, 38, 29, 25, 22, 22, 19, 16, 13, 10, 9, 7, 6, 5, 3, 2, \mathbf{1}, \mathbf{1})$$

performed by *PatternRec*. We assume to know $t_1 = 5$ and $z = 6$.
First, an 18-length vector s is allocated, and its last thirteen elements are initialized by the knowledge of $n = 18$ and $z = 6$:

$$s = (s(1), s(2), s(3), s(4), s(5), 0, -1, -2, -3, \ldots, -12).$$

Then, the value $t_2 = 2$ is retrieved by counting the number of vertices of degree one in π, and the computation of the values $s(1), s(2)$ and $s(3)$ is achieved through the resolution of the linear system

$$\begin{cases} s(1) - s(2) = 5 \\ s(2) - s(3) = 2 \\ s(1) + s(2) - 12 = 1 \end{cases}$$

getting

$$s = (9, 4, 2, s(4), s(5), 0, -1, -2, -3, \ldots, -12).$$

Now, the insertion of the following hyperedges is performed:

$$\begin{array}{ll} (v_1, v_2, v_k) & \text{for } 3 \leq k \leq 18, \\ (v_1, v_3, v_k) & \text{for } 4 \leq k \leq 16, \\ (v_2, v_3, v_k) & \text{for } 4 \leq k \leq 11, \\ (v_i, v_j, v_k) & \text{such that } i = 1, 2, 3, j = 6, \ldots, 17, k = j+1, \ldots, 18, \text{ and } s(i) + s(j) + s(k) > 0. \end{array}$$

After the update, the integer sequence is

$$\pi' = (28, 10, 6, 22, 19, 7, 11, 5, 4, 3, 2, 2, 2, 2, 1, \mathbf{0}, 0, 0),$$

with a new null entry (in boldface). Then, $t_3 = 1$ and so $s(4) = 1$.
The algorithm performs now the insertion of all the edges involving v_4 and the known values of s. We get

$$\pi'' = (17, 3, 2, 1, 16, 4, 8, 2, 2, 1, 1, 1, 1, 1, \mathbf{0}, 0, 0, 0),$$

again with one new null entry (in boldface). Then, $t_4 = 1$ and so $s(5) = 0$, and the final computation of the sequence s is reached:

$$s = (9, 4, 2, 1, 0, 0, -1, -2, -3, \ldots, -12).$$

A final check reveals that the hypergraph H_s actually realizes the input sequence π.

Theorem 3.2.16 ([5]). Given a non-increasing integer sequence π of length n, if $\pi \in \mathcal{P}$, then *PatternRec* provides its unique realization with a computational cost of $O(n^3)$.

Proof. We start from the pseudo-code described in Algorithm 3. Since *PatternRec* follows an iterative procedure, the proof will be by induction. More specifically, we have to show that at each step a new element of the integer sequence s is correctly detected in the *for* loop of line 13. So, the induction will be on the index l of such a loop.
Before starting the analysis of the loop, we observe that at the beginning of the procedure the values $s(1), s(2), s(3)$ and $s(z), \ldots, s(n)$ are correctly detected. These last obviously, by the knowledge of the values z and n. Moreover, the required condition $s(1) + s(2) + s(n) = 1$, together with the information $t_1 = s(1) - s(2)$, immediately provide the values $s(1)$ and $s(2)$. We finally analyze $s(3)$. At line 3 of the pseudo-code, the algorithm computes q, the number of 1s in the sequence π. If π is a sequence in $\mathcal{D}$ then, by construction, a vertex v_k takes degree equal to 1 if and only if it appears in the hyperedge (v_1, v_2, v_k) only. So, for all the vertices v_k of degree 1, we have that $s(1) + s(2) + s(k) > 0$ and $s(1) + s(3) + s(k) \leq 0$ hold at the same time, since $s(1) + s(2)$ and $s(1) + s(3)$ are the greatest sum we can have between elements in s. It follows that the value q must provide the difference between the number of vertices with degree strictly greater than one and those ones with degree exactly equal to 1, i.e., the difference between $s(2)$ and $s(3)$. Then, $s(3)$ can be correctly computed as $s(2) - q$. We are now ready for our induction proof.

Algorithm 3. *PatternRec(π,z,t_1)*

Input: A non-increasing integer sequence π of length n, an index $2 \leq z \leq n-1$, a positive integer value $0 \leq t_1 \leq \left\lfloor \frac{n-2}{2} \right\rfloor$

```
Initialize s a null array of length n;
set (s(z), ..., s(n)) = (0, -1, ..., -(n - z));
compute q the number of elements in π that are equal to 1;
set s(2) = (1 - s(n) - t_1)/2, s(1) = 1 - s(n) - s(2) and s(3) = s(2) - q;
for i, j, k ∈ {1, 2, 3, z, z + 1, ..., n} and i < j < k do
    if s(i) + s(j) + s(k) > 0 then
        π(i) = π(i) - 1;
        π(j) = π(j) - 1;
        π(k) = π(k) - 1;
    end
end
π^(4) = π;
for l = 4 : z - 1 do
    compute w the number of elements in π^(l) that are equal to 0;
    set r = w - q; s(l) = s(l - 1) - r; q = q + r; π^(l+1) = π^(l);
    for i, j, k ∈ {1, ..., l, z, z + 1, ..., n} and i ≠ j ≠ k do
        if s(i) + s(j) + s(k) > 0 and was not previously computed then
            π^(l+1)(i) = π^(l+1)(i) - 1;
            π^(l+1)(j) = π^(l+1)(j) - 1;
            π^(l+1)(k) = π^(l+1)(k) - 1;
        end
    end
end
compute the degree sequence π* generated by s;
if π* = π then
    compute H_s;
    return success;
else
    return failure;
end
```

Output: H_s

Base case, $l = 4$**.** The algorithm inserts all the hyperedges of type (v_i, v_j, v_k) that belong to the hypergraph, for $i < j < k$ chosen in the set of indexes for which we know the values of s, namely, $\{1, 2, 3, z, z+1, \ldots, n\}$. Then, the input sequence is updated at $\pi^{(4)} = (*, \ldots, *, 0^{r_4}, 0^q)$, where stars stay for non-null values. Notice that all the previous vertices of degree 1 now lowered their degree to 0, as well as other r_4 degrees reached such a value, for some $r_4 \geq 0$. By construction, we have:

i) $\pi^{(4)}(n - q - r_4) \neq 0$.

ii) The edge (v_1, v_4, v_{n-q-r_4}) belongs to the hypergraph since, considering all the insertions already performed, now $s(1) + s(4)$ is the greatest sum we can have between elements in s, and v_{n-q-r_4} belongs (at least) to another edge of the hypergraph, see point *i)*. Then, $s(1) + s(4) + s(n - q - r_4) > 0$ must hold.

iii) The last q degrees of the sequence became null due to the insertion of edges of type (v_1, v_2, v_k), for $k = n - q + 1, \ldots, n$.

iv) The vertices from position $n - q - r_4 + 1$ to $n - q$ had degree strictly greater than one, and so became null due to the insertion of edges of type (v_1, v_2, v_k) and (v_1, v_3, v_k), for $k = n - q - r_4 + 1, \ldots, n - q$.

From *iii)* and *iv)*, it follows that r_4 is the number of vertices v_k such that $s(1) + s(3) + s(k) > 0$ and $s(1) + s(4) + s(k) \leq 0$ hold at the same time or, equivalently, $r_4 = s(3) - s(4)$. Then, the value $s(4)$ is correctly detected.

Induction step. We assume that the first l values of the sequence s have been correctly computed, and show that *PatternRec* detects $s(l + 1)$ without errors.

After the insertion of all the edges involving the indices in $\{1, 2, \ldots, l, z, z+1, \ldots, n\}$, the input sequence has been updated at $\pi^{(l+1)} = (*, \ldots, *, 0^{r_{l+1}}, 0^{q_{l+1}})$, where the null degrees corresponding to the entries $0^{r_{l+1}}$ were strictly greater than zero in the previous step. Indeed, by induction hypothesis, $q_{l+1} = \sum_{i=2}^{l-1} s(i) - s(i+1)$. As detailed before, this implies that the insertion of the edges (v_1, v_l, v_k) has been performed for all $k = l + 1, \ldots, n - r_{l+1} - q_{l+1}$, and also that $(v_1, v_{l+1}, v_{n-r_{l+1}-q_{l+1}})$ is part of the hypergraph since $v_{n-r_{l+1}-q_{l+1}}$ has still a positive degree. In other words, the value r_{l+1} corresponds to the number of indices k for which $s(1)+s(l)+s(k) > 0$ and $s(1) + s(l+1) + s(k) \leq 0$ hold at the same time, that is, $r_{l+1} = s(l) - s(l+1)$. Then, the value $s(l + 1)$ is correctly computed, and the proof is given.

The computational cost is clearly polynomial in the size of the input, of order $O(n^3)$. Indeed, the detection of the sequence s goes hand by hand with the insertion of edges and the construction of the hypergraph, cubic in the size of the input. □

3.3 Order Theory and ideals, an extension of the class $\mathcal{D}$

In the previous section, we carried on the study of a subclass of 3-graphic sequences, $\mathcal{P} \subset \mathcal{D}$. We now define an extension of $\mathcal{D}$, by considering the poset of triplets $\mathcal{T}_n = (\Omega_n, \preceq)$ introduced in Section 3.1. $\mathcal{D}_n^{ext}$ is defined as the set of all the non-increasing integer

sequences π of length n that are realized by a hypergraph H associated to an ideal of the poset $\mathcal{T}_n$. As already mentioned, the realization of a sequence in $\mathcal{D}_n^{ext}$ is not unique, in general (see Example 3.1.11). We dedicate this section to the study of the properties of the sequences realized by ideals with one or two generators, providing their complete characterization and, mostly, an efficient reconstruction strategy.
From now on, we consider hypergraphs with n non-isolated vertices.

3.3.1 Principal ideals

We remind that an ideal $I \in \mathcal{I}_n$ is *principal* if the antichain of its maximal elements has cardinality one or, equivalently, if it can be generated by one element, i.e., $I_g = \downarrow\{g\}$ for some $g = (a, b, c) \in \Omega_n$. We denote by $\pi_g = (d_1, \ldots, d_n) \in \mathcal{D}_n^{ext}$ the degree sequence of the hypergraph in bijection with I_g.

Remark 3.3.1. From the assumption that no isolated vertices are present in the hypergraph, it follows that $g = (a, b, n)$.

The correspondence between hypergraphs and ideals allows to compute the degree sequence $\pi_g = (d_1, \ldots, d_n)$ looking at the triplets of $\downarrow\{g\}$, by counting the occurrences of the value i, for $1 \leq i \leq n$, that appear in first, second, or third position in the triplets. Starting from this observation, we express the degree of each vertex in terms of the values a and b only, where a and b define the generator $g = (a, b, n)$.

Lemma 3.3.2 ([10])**.** Let $g = (a, b, n) \in \Omega_n$. For all $1 \leq i \leq n$, the number of occurrences of the element i as first entry of a triplet in I_g is

$$O_1^i = \begin{cases} \frac{(i-b)(i+b-2n+1)}{2} & \text{if } 1 \leq i \leq a, \\ 0 & \text{if } a+1 \leq i \leq n. \end{cases}$$

Proof. By definition of principal ideal, $I_g = \downarrow\{(a, b, n)\}$, any number strictly greater than a cannot appear as first element of a triplet. This leads to $O_1^i = 0$ for all $a + 1 \leq i \leq n$. On the other hand, if $1 \leq i \leq a$, then O_1^i is given by the number of triplets of type (i, x, y), with $i + 1 \leq x \leq b$ and $x + 1 \leq y \leq n$, that is,

$$O_1^i = \sum_{x=i+1}^{b} \sum_{y=x+1}^{n} 1.$$

Solving the sum gives the thesis. □

Lemma 3.3.3 ([10])**.** Let $g = (a, b, n) \in \Omega_n$. For all $1 \leq i \leq n$, the number of occurrences of the element i as second entry of a triplet in I_g is

$$O_2^i = \begin{cases} (i-1)(n-i) & \text{if } 1 \leq i \leq a, \\ a(n-i) & \text{if } a+1 \leq i \leq b, \\ 0 & \text{if } b+1 \leq i \leq n. \end{cases}$$

Proof. By definition of principal ideal, $I_g = \downarrow\{(a,b,n)\}$, any number strictly greater than b cannot appear as second element of a triplet. This leads to $O_2^i = 0$ for all $b+1 \leq i \leq n$. On the other hand, if $1 \leq i \leq b$, then O_2^i is given by the number of triplets of type (x,i,y), with $1 \leq x \leq \min\{i-1,a\}$ and $i+1 \leq y \leq n$, that is,

$$O_2^i = \sum_{x=1}^{\min\{i-1,a\}} \sum_{y=i+1}^{n} 1.$$

Two cases arise:

i) If $i \leq a$, then $O_2^i = \sum_{x=1}^{i-1} \sum_{y=i+1}^{n} 1$, and so $O_2^i = (i-1)(n-i)$.

ii) Otherwise, $O_2^i = \sum_{x=1}^{a} \sum_{y=i+1}^{n} 1$, and so $O_2^i = a(n-i)$.

□

Lemma 3.3.4 ([10]). Let $g = (a,b,n) \in \Omega_n$. For all $1 \leq i \leq n$, the number of occurrences of the element i as third entry of a triplet in I_g is

$$O_3^i = \begin{cases} \frac{1}{2}i^2 - \frac{3}{2}i + 1 & \text{if } 1 \leq i \leq a, \\ \frac{a(2i-a-3)}{2} & \text{if } a+1 \leq i \leq b, \\ \frac{a(2b-a-1)}{2} & \text{if } b+1 \leq i \leq n. \end{cases}$$

Proof. Since no isolated vertices occur in a realization of π_g, and being I_g a down-set, any element $1 \leq i \leq n$ can appear as third entry of the triplets of the ideal. Precisely, O_3^i is given by the number of triplets of type (x,y,i), with $1 \leq x \leq \min\{i-2,a\}$ and $x+1 \leq y \leq \min\{i-1,b\}$, that is,

$$O_3^i = \sum_{x=1}^{\min\{i-2,a\}} \sum_{y=x+1}^{\min\{i-1,b\}} 1.$$

Three cases arise:

i) If $i \leq a$, then $O_3^i = \sum_{x=1}^{i-2} \sum_{y=x+1}^{i-1} 1$, and so $O_3^i = \frac{1}{2}i^2 - \frac{3}{2}i + 1$.

ii) If $a+1 \leq i \leq b$, then $O_3^i = \sum_{x=1}^{a} \sum_{y=x+1}^{i-1} 1$, and so $O_3^i = \frac{a(2i-a-3)}{2}$.

iii) Otherwise, $O_3^i = \sum_{x=1}^{a} \sum_{y=x+1}^{b} 1$, and so $O_3^i = \frac{a(2b-a-1)}{2}$.

□

Putting together the results of Lemmas 3.3.2, 3.3.3 and 3.3.4, the degree sequence π_g of a principal ideal can be computed:

Theorem 3.3.5 ([10]). Let $g = (a, b, n) \in \Omega_n$, and $\pi_g = (d_1, \ldots, d_n)$ be the degree sequence of the hypergraph defined by I_g. Then,

$$d_i = \begin{cases} \frac{(b-1)(2n-b-2)}{2} & \text{if } 1 \leq i \leq a, \\ \frac{a(2n-a-3)}{2} & \text{if } a+1 \leq i \leq b, \\ \frac{a(2b-a-1)}{2} & \text{if } b+1 \leq i \leq n. \end{cases} \tag{3.1}$$

Proof. The values d_i, for $i = 1, \ldots, n$, can be immediately computed starting from the results given in Lemmas 3.3.2, 3.3.3 and 3.3.4 as $d_i = O_1^i + O_2^i + O_3^i$. We explicitly write the calculations that lead to the thesis:

$1 \leq i \leq a$. In this case, O_1^i, O_2^i and O_3^i are all different from zero, so that

$$d_i = \frac{(i-b)(i+b-2n+1)}{2} + (i-1)(n-i) + \frac{1}{2}i^2 - \frac{3}{2}i + 1 = \frac{(b-1)(2n-b-2)}{2}.$$

$a+1 \leq i \leq b$. In this case, O_1^i is equal to zero, so that

$$d_i = a(n-i) + \frac{a(2i-a-3)}{2} = \frac{a(2n-a-3)}{2}.$$

$b+1 \leq i \leq n$. This is the final case, where the only non-null summand is O_3^i, that is,

$$d_i = \frac{a(2b-a-1)}{2}.$$

□

Remark 3.3.6. We underline that the value d_i does not depend on the index i, for $1 \leq i \leq n$, but is determined by a, b and n only, i.e., by the generator $g = (a, b, n)$ of the principal ideal.

It is clear from Theorem 3.3.5 that the degrees of the vertices associated to a principal ideal can assume at most three distinct values; moreover, this number can decrease to two or even one, depending on the entries a, b of the triplet that generates the hypergraph. We characterize all the possible cases in the following result.

Corollary 3.3.7 ([10]). The degrees of the vertices associated to a principal ideal can assume at most three distinct values.

Proof. Starting from Eq. (3.1) in Theorem 3.3.5, we denote the values of the degrees as $p_1 = \frac{(b-1)(2n-b-2)}{2}$, $p_2 = \frac{a(2n-a-3)}{2}$ and $p_3 = \frac{a(2b-a-1)}{2}$. We analyze the cases in which two or three of these values coincide, as well as the case in which they are all distinct:

i) $p_1 = p_2 = p_3$, if and only if $a = n-2$ and $b = n-1$. Here, $I_g = \downarrow\{(n-2, n-1, n)\}$ is the complete 3-hypergraph on n vertices, indeed its degree sequence is constant and equal to $\pi_g = (p_1^n)$, with $p_1 = \frac{(n-1)(n-2)}{2}$.

ii) $p_1 \neq p_2$ and $p_2 = p_3$, if and only if $a < n-2$ and $b = n-1$. In this case, the degree sequence has two distinct entries, $p_1 = \frac{(n-1)(n-2)}{2}$ and $p_2 = \frac{a(2n-a-3)}{2}$, and it is equal to $\pi_g = (p_1^a, p_2^{n-a})$.

iii) $p_1 = p_2$ and $p_2 \neq p_3$, if and only if $b = a+1$ and $b < n-1$. In this case, the degree sequence has two distinct entries, $p_1 = \frac{a(2n-a-3)}{2}$ and $p_3 = \frac{a(a+1)}{2}$, and it is equal to $\pi_g = (p_1^b, p_3^{n-b})$.

iv) $p_1 \neq p_2 \neq p_3$, if and only if no consecutive elements are present in the generator $g = (a, b, n)$. In this case, the degree sequence has three distinct entries and it is equal to $\pi_g = (p_1^a, p_2^{b-a}, p_3^{n-b})$, with p_1, p_2 and p_3 given in Eq. (3.1).

□

Remark 3.3.8. The number of distinct degrees in π_g depends on how much *close* are the entries a, b and n in the triplet g. In particular, each time two consecutive numbers are present in the generator, the number of distinct degrees in π_g decreases by one.

Example 3.3.9. Let us consider three principal ideals in $\mathcal{I}_{10}$ and the related degree sequences:

$$\begin{aligned} g_1 &= (2, 7, 10) & \pi_1 &= (33, 33, 15, 15, 15, 15, 15, 11, 11, 11), \\ g_2 &= (5, 6, 10) & \pi_2 &= (30, 30, 30, 30, 30, 30, 15, 15, 15, 15), \\ g_3 &= (8, 9, 10) & \pi_3 &= (36, 36, 36, 36, 36, 36, 36, 36, 36, 36). \end{aligned}$$

According to Corollary 3.3.7, the presence of consecutive numbers in the generator decreases the number of distinct values in the degree sequence.

When considering a non-increasing integer sequence π of length n where three distinct entries are present, at most, one can ask if π is the degree sequence associated to a principal ideal in $\mathcal{T}_n$ and, in case of positive answer, find its realization through the detection of its unique generator. We already observed that such a triplet depends on two distinct values only, that can be computed solving the following system of equations:

$$\begin{cases} p_1 = \frac{(b-1)(2n-b-2)}{2} \\ p_2 = \frac{a(2n-a-3)}{2} \\ 1 \leq a < b \leq n-1 \text{ integers,} \end{cases} \tag{3.2}$$

where p_1 and p_2 are the two greatest distinct values that occur in π. The structure of (3.2) is particularly easy: indeed, it is composed by two independent quadratic equations in a and b, and so can be quickly solved. Moreover, if it admits a solution, then such a solution is unique.

Theorem 3.3.10 ([10]). If it exists, the solution of (3.2) is unique.

Proof. The system described in (3.2) consists of two quadratic equations in a and b that can be solved independently. We show that each of them admits a unique solution satisfying the required constraints on the respective variable, i.e., $1 \leq a < b \leq n-1$. We consider the first equation. The uniqueness of the solution for the second one can be proven following the same argument.
The value b is obtained by solving

$$b^2 + (1-2n)b + 2n + 2p_1 - 2 = 0,$$

whose solutions are $b_{1,2} = \frac{2n-1\pm\sqrt{\Delta_1}}{2}$, where $\Delta_1 = (2n-3)^2 - 8p_1$. Since $b \leq n-1$, the inequality $\pm\sqrt{\Delta_1} \leq -1$ must hold, that is, the only admissible solution is $b = \frac{2n-1-\sqrt{\Delta_1}}{2}$. Similarly, $a = \frac{2n-3-\sqrt{\Delta_2}}{2}$, where $\Delta_2 = (2n-3)^2 - 8p_2$, is the unique admissible solution for the equation in the variable a. □

Example 3.3.11. Let us consider again the degree sequence π_1 in Example 3.3.9,

$$\pi_1 = (33, 33, 15, 15, 15, 15, 15, 11, 11, 11).$$

The generator $g_1 = (a, b, 10)$ of the principal ideal that may realize π_1 can be computed solving the system of equations

$$\begin{cases} 33 = \frac{(b-1)(20-b-2)}{2} \\ 15 = \frac{a(20-a-3)}{2} \\ 1 \leq a < b \leq 9 \text{ integer numbers,} \end{cases}$$

or, equivalently, the quadratic equations

$$\begin{aligned} b^2 - 19b + 84 = 0, \\ a^2 - 17a + 30 = 0. \end{aligned}$$

The only solutions that are in accordance with the condition $1 \leq a < b \leq 9$ are $a = 2$ and $b = 7$, that in fact give the correct generator $g_1 = (2, 7, 10)$ (cf. Example 3.3.9).

Theorem 3.3.10 provides a sufficient condition for the resolution of the consistency problem for an integer sequence of type $\pi = (d_x^\alpha, d_y^\beta, d_z^\gamma)$, but does not solve it in general. Indeed, it is sufficient to solve the system of equations (3.2) to establish if π can be realized by a principal ideal in $\mathcal{T}_n$, but Theorem 3.3.10 does not allow to solve the consistency problem in case of negative answer. Moreover, we underline that the uniqueness of a and b as solutions of (3.2) guarantees that, if it exists, the principal ideal realizing π is unique, but does not guarantee that the realization of π is unique in general. As a matter of fact, another hypergraph having π as degree sequence could exist, both in $\mathcal{I}_n$ (in this case, an ideal with at least two generators) or outside the class of ideals of $\mathcal{T}_n$ (in this case, $\pi \notin \mathcal{D}_n^{ext}$).
Actually, we can solve the uniqueness problem and show that the degree sequence of a principal ideal I_g admits a unique realization.

Theorem 3.3.12 ([10]). Given an integer sequence $\pi = (d_1, \ldots, d_n)$, if there exists a principal ideal $I_g \in \mathcal{I}_n$ realizing π, then no other realization of π exists, up to isomorphism.

Proof. By Theorem 3.1.10, if we show that $\pi \in \mathcal{D}$, then uniqueness directly follows. We proceed in the research of an integer sequence s such that $\pi = \pi_s$. We distinguish three cases, depending on the number of distinct entries in π, that are at most three by Corollary 3.3.7.

i) All vertices have the same degree, namely, $\pi = (p^n)$ for some p. Since by hypothesis $\pi = \pi_g$ for some g, from Corollary 3.3.7 it is $p = \frac{(n-1)(n-2)}{2}$, and so I_g is the complete 3-hypergraph on n vertices. The same hypergraph can be constructed starting from the integer sequence $s = (1^n)$, and so $\pi \in \mathcal{D}$.

ii) The degrees of the vertices assume two distinct values, namely, $\pi = (p^x, q^{n-x})$ for some p, q and x. According to Corollary 3.3.7, two further cases must be analyzed:

a) $q = \frac{x(2n-x-3)}{2}$, and so $x = a$ and $p = \frac{(n-1)(n-2)}{2}$ (these values are unique from Theorem 3.3.10). We show that the integer sequence $s = (1^a, 0^{n-a})$ is such that $\pi = \pi_s = (d_1^s, \ldots, d_n^s)$. From Proposition 3.1.9, it is sufficient to show that $d_1^s = p$ and $d_n^s = q$.

The first vertex always appears in the first position of the triplets of Ω_n, and those ones that constitute the hypergraph H_s are such that $s(1)+s(j)+s(k) > 0$. As $s(1) = 1$, the inequality holds if and only if $(s(j), s(k)) = (1,1)$, $(1,0)$ or $(0,0)$. The number of possible choices for $(s(j), s(k))$ is $\binom{a-1}{2} + (a-1)(n-a) + \binom{n-a}{2}$, that is, $d_1^s = \frac{(n-1)(n-2)}{2} = p$.

Similarly, the last vertex v_n appears in the triplets of Ω_n in third position only, and an edge (i, j, n) is part of the hypergraph H_s if and only if $s(i) + s(j) + s(n) > 0$, with $s(n) = 0$. It follows that $(s(i), s(j)) = (1,1)$ or $(1,0)$, with $\binom{a}{2} + a(n-a-1)$ possible choices. As a consequence, $d_n^s = \frac{a(2n-a-3)}{2} = q$, as claimed.

b) $p = \frac{(x-1)(2n-x-2)}{2}$, and so $x = b$ and $q = \frac{b(b-1)}{2}$ (these values are unique from Theorem 3.3.10). Following the same argument used in a), $\pi = \pi_s$ holds choosing $s = (2^b, -1^{n-b})$.

iii) The degrees of the vertices assume three distinct values, namely, $\pi = (p^x, q^y, r^{n-x-y})$ for some p, q, r and x, y. Again from Corollary 3.3.7 and Theorem 3.3.10, we know that $p = \frac{(b-1)(2n-b-2)}{2}$, $q = \frac{a(2n-a-3)}{2}$ and $r = \frac{a(2b-a-1)}{2}$, and $x = a$, $y = b - a$.

We consider the integer sequence $s = (2^a, 0^{b-a}, -1^{n-b})$ and the associated degree sequence $\pi_s = (d_1^s, \ldots, d_n^s) \in \mathcal{D}$. As shown before, it is easy to verify that $d_1^s = p$ and $d_n^s = r$, as required. We finally need to check if $d_{a+1}^s = q$. Since $s(a+1) = 0$, the vertex v_{a+1} can appear both as second and third element of the triplets of Ω_n. In the first case, we have to count the number of pairs of type $(s(i), s(k))$ such that $s(i) + 0 + s(k) > 0$, in the latter the number of pairs $(s(i), s(j))$ such that

$s(i) + s(j) + 0 > 0$. In total, we need the number of pairs of type $(2, 0)$, $(2, -1)$ or $(2, 2)$, that is, $a(b - a - 1) + a(n - b) + \binom{a}{2} = q$, as claimed.

We have shown that, in all cases, if $\pi = \pi_g$ for some $g \in \Omega_n$, then $\pi \in \mathcal{D}$ and, as a consequence, the only hypergraph realizing it is the principal ideal $I_g \in \mathcal{I}_n$. This concludes the proof. □

Remark 3.3.13. Theorem 3.3.12 guarantees the uniqueness of reconstruction in case of a degree sequence associated to a principal ideal in $\mathcal{I}_n$, but does not characterize all the 3-graphic sequences in which only three (or less) distinct degrees are present. As an example, the integer sequence $\pi = (1, 1, 1, 1, 1, 1)$ is 3-graphic, but (one of) its realization $H = \{(1, 2, 3), (4, 5, 6)\}$ is not in $\mathcal{I}_6$.

We conclude the study of principal ideals with the description of a trivial reconstruction strategy for this type of hypergraphs, whose pseudo-code is described in Algorithm 4.

Algorithm 4. *Generator(π)*

Input: $\pi = (p^x, q^y, r^z)$ a non-increasing integer sequence with (at most) three distinct entries

```
Compute x, y and z the number of occurrences of each distinct entry in π;
if y ≠ 0 and z = 0 then
    set y = 1 and z = n − x − 1;
end
if y = 0 and z = 0 then
    set x = n − 2, y = 1 and z = 1;
end
Set g = (x, x + y, x + y + z);
Compute the principal ideal I_g and its degree sequence π_g;
if π_g = π then
    H = I_g;
else
    return failure;
end
```

Output: H

Theorem 3.3.14. Given a non-increasing integer sequence $\pi = (d_1, \ldots, d_n)$ with at most three distinct entries, it is possible to establish in polynomial time if π is realized by a principal ideal in $\mathcal{I}_n$. In case of positive answer, the realization is unique and can be provided in $O(n^3)$ time.

The result is a straightforward consequence of Theorem 3.3.5, that characterizes the degrees of the sequences realized by principal ideals, and Theorem 3.3.12, that guarantees the uniqueness property. The generator $g = (a, b, n)$ can be found simply counting the number of entries for each value in π, see Theorem 3.3.5, with a computational cost that is linear in the size of the input. The generation of the corresponding hypergraph requires a computational cost of $O(n^3)$, again polynomial in the size of the input.

Example 3.3.15. Let us consider the following integer sequences of length $n = 12$,

$$\pi_1 = (34, 34, 34, 27, 27, 9, 9, 9, 9, 9, 9, 9),$$
$$\pi_2 = (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2),$$

and the performance of the algorithm *Generator*.
In the first case, the algorithm computes $x = 3$, $y = 2$ and $z = 7$, and then the candidate generator $g_1 = (3, 5, 12)$. Since the degree sequence of the principal ideal $\downarrow\{(3, 5, 12)\}$ is equal to the input, the algorithm successfully ends providing the hypergraph I_{g_1} as output.
In the second case, the candidate generator is computed as $g_2 = (10, 11, 12)$, and the algorithm fails since π_2 does not coincide with the degree sequence of the complete 3-hypergraph on twelve vertices.

3.3.2 Ideals with two generators

We now move to the case of ideals with two generators, $g_1 = (a, b, c)$ and $g_2 = (d, e, f)$. By definition, the two triplets must be non-comparable w.r.t. the order $\preceq$ defined on $\mathcal{T}_n$. Moreover, at least one between c and f must be equal to n, to avoid isolated vertices in the hypergraph $I_{g_1,g_2} = \downarrow\{g_1, g_2\}$. Without loss of generality, we assume $f = n$.
As we did for principal ideals, we proceed in characterizing the degree sequences associated to such hypergraphs. Even in this case, it is possible to compute the degrees of the vertices starting from the values a, b, c, d and e only, that uniquely identify the generators. Differently from principal ideals, several cases arise, according to the condition of non-comparability of g_1 and g_2.
To avoid tedious repetitions, in this section we present our results omitting detailed computations in the proofs. For a more complete and precise version, we address the reader to our paper [10].

Lemma 3.3.16 ([10])**.** Let $g_1 = (a, b, c), g_2 = (d, e, n) \in \Omega_n$ be non-comparable, $I_{g_1,g_2} = \downarrow\{g_1, g_2\}$, and π_{g_1,g_2} the degree sequence of the hypergraph H_{g_1,g_2} associated to I_{g_1,g_2}. Then,

$$\pi_{g_1,g_2} = \pi_{g_1} + \pi_{g_2} - \pi_{\min\{g_1,g_2\}},$$

where $\min\{g_1, g_2\} = (\min\{a, d\}, \min\{b, e\}, \min\{c, n\})$.

Proof. The proof directly follows from the inclusion-exclusion principle, being $I_{\min\{g_1,g_2\}} = I_{g_1} \cap I_{g_2}$ by definition and, as a consequence, $I_{g_1,g_2} = I_{g_1} \cup (I_{g_2} \setminus I_{\min\{g_1,g_2\}})$. □

General case

Since we know how to compute the degree sequence π_g for a given $g \in \Omega_n$ (see Theorem 3.3.5), we can easily obtain the degree sequences of the form π_{g_1,g_2} by exploiting Lemma 3.3.16. For the moment, we assume the values a, b, c, d and e to be all distinct and different from n. The cases in which repetitions occur will be treated later.

Theorem 3.3.17 ([10])**.** Given two non-comparable triplets $g_1 = (a, b, c), g_2 = (d, e, n) \in \Omega_n$, the degree sequence $\pi_{g_1,g_2} = (d_1, \ldots, d_n)$ of the hypergraph associated to $I_{g_1,g_2} = \downarrow\{g_1, g_2\}$ is:

i) If $d < a < b < e < c < n$, then

$$d_i = \begin{cases} \frac{(e-1)(2n-e-2)}{2} & \text{if } 1 \leq i \leq d \\ \frac{(b-1)(2c-b-2)+2d(n-c)}{2} & \text{if } d+1 \leq i \leq a \\ \frac{a(2c-a-3)+2d(n-c)}{2} & \text{if } a+1 \leq i \leq b \\ \frac{a(2b-a-1)+2d(n-b-1)}{2} & \text{if } b+1 \leq i \leq e \\ \frac{a(2b-a-1)+2d(e-b)}{2} & \text{if } e+1 \leq i \leq c \\ \frac{d(2e-d-1)}{2} & \text{if } c+1 \leq i \leq n. \end{cases}$$

ii) If $d < a < b < c < e < n$, then

$$d_i = \begin{cases} \frac{(e-1)(2n-e-2)}{2} & \text{if } 1 \leq i \leq d \\ \frac{(b-1)(2c-b-2)+2d(n-c)}{2} & \text{if } d+1 \leq i \leq a \\ \frac{a(2c-a-3)+2d(n-c)}{2} & \text{if } a+1 \leq i \leq b \\ \frac{a(2b-a-1)+2d(n-b-1)}{2} & \text{if } b+1 \leq i \leq c \\ \frac{d(2n-d-3)}{2} & \text{if } c+1 \leq i \leq e \\ \frac{d(2e-d-1)}{2} & \text{if } e+1 \leq i \leq n. \end{cases}$$

iii) If $d < a < e < b < c < n$, then

$$d_i = \begin{cases} \frac{(b-1)(2c-b-2)+2(e-1)(n-c)}{2} & \text{if } 1 \leq i \leq d \\ \frac{(b-1)(2c-b-2)+2d(n-c)}{2} & \text{if } d+1 \leq i \leq a \\ \frac{a(2c-a-3)+2d(n-c)}{2} & \text{if } a+1 \leq i \leq e \\ \frac{a(2c-a-3)}{2} & \text{if } e+1 \leq i \leq b \\ \frac{a(2b-a-1)}{2} & \text{if } b+1 \leq i \leq c \\ \frac{d(2e-d-1)}{2} & \text{if } c+1 \leq i \leq n. \end{cases}$$

iv) If $d < e < a < b < c < n$, then

$$d_i = \begin{cases} \frac{(b-1)(2c-b-2)+2(e-1)(n-c)}{2} & \text{if } 1 \leq i \leq d \\ \frac{(b-1)(2c-b-2)+2d(n-c)}{2} & \text{if } d+1 \leq i \leq e \\ \frac{(b-1)(2c-b-2)}{2} & \text{if } e+1 \leq i \leq a \\ \frac{a(2c-a-3)}{2} & \text{if } a+1 \leq i \leq b \\ \frac{a(2b-a-1)}{2} & \text{if } b+1 \leq i \leq c \\ \frac{d(2e-d-1)}{2} & \text{if } c+1 \leq i \leq n. \end{cases}$$

v) If $a < d < e < b < c < n$, then

$$d_i = \begin{cases} \frac{(b-1)(2c-b-2)+2(e-1)(n-c)}{2} & \text{if } 1 \leq i \leq a \\ \frac{(e-1)(2n-e-2)}{2} & \text{if } a+1 \leq i \leq d \\ \frac{d(2n-d-3)}{2} & \text{if } d+1 \leq i \leq e \\ \frac{2a(c-e-1)+d(2e-d-1)}{2} & \text{if } e+1 \leq i \leq b \\ \frac{2a(b-e)+d(2e-d-1)}{2} & \text{if } b+1 \leq i \leq c \\ \frac{d(2e-d-1)}{2} & \text{if } c+1 \leq i \leq n. \end{cases}$$

Proof. The computation directly follows from Theorem 3.3.5 and Lemma 3.3.16 (see [10] for the complete calculations). □

Theorem 3.3.17 states that the degree sequences associated to ideals with two generators have at most six distinct entries. As in case of principal ideals, this number can decrease in case of consecutive numbers in the triplets of generators or in case of repetitions, as an example, if $a = d$.

Remark 3.3.18. Each repetition/pair of consecutive numbers in g_1 and g_2 makes the number of distinct degrees in π_{g_1,g_2} decrease by one, as shown in the sequel.

Degree sequences containing three distinct values

By an exhaustive check of the equations provided in Theorem 3.3.17, we characterize all the degree sequences associated to ideals with two generators having exactly three distinct entries. Moreover, the uniqueness property holds for this class.

Theorem 3.3.19 ([10])**.** The degree sequence π_{g_1,g_2} has exactly three distinct entries if and only if g_1 and g_2 are chosen as follows:

i) $g_1 = (a, n-1, n)$ and $g_2 = (d, d+1, n)$, with $a < d$, or

ii) $g_1 = (a, a+1, a+2)$ and $g_2 = (d, n-1, n)$, with $d < a$, or

iii) $g_1 = (a, a+1, a+2)$ and $g_2 = (d, d+1, n)$, with $d < a$.

Moreover, the hypergraph associated to the ideal I_{g_1,g_2} is the unique realization of π_{g_1,g_2} (up to isomorphism).

Proof. The choice of generators that allows to reduce the number of distinct degrees, still keeping them non-comparable, comes from an exhaustive check of all the possible cases. The uniqueness property follows by the fact that $\pi_{g_1,g_2} \in \mathcal{D}$. We provide here the sequence s that realizes the degree sequence in each case:

i) $s = (3^a, 2^{d+1-a}, -1^{n-d-1})$,

ii) $s = (5^d, 1^{a+2-d}, -2^{n-a-2})$,

iii) $s = (2^{d+1}, 1^{a-d+1}, -3^{n-a-2})$.

□

Remark 3.3.20. If we try to reduce the number of distinct degrees to two, we get $g_1 \preceq g_2$ or vice versa, going back to the definition of principal ideal. Then, three is the minimum number of distinct entries in a degree sequence associated to an ideal with two generators.

Degree sequences containing four distinct values

Analogously, we characterize all the degree sequences associated to ideals with two generators having exactly four distinct entries. Even in this case, the uniqueness property is maintained.

Theorem 3.3.21 ([10]). The degree sequence π_{g_1,g_2} has exactly four distinct entries if and only if g_1 and g_2 are chosen as follows:

i) $g_1 = (a, n-1, n)$ and $g_2 = (d, e, n)$, with $a < d < e-1$, or

ii) $g_1 = (a, b, n)$ and $g_2 = (d, d+1, n)$, with $a < d < b$, or

iii) $g_1 = (a, a+1, a+2)$ and $g_2 = (d, e, n)$, with $d < a < e$, or

iv) $g_1 = (a, b, b+1)$ and $g_2 = (d, n-1, n)$, with $d < a < b$, or

v) $g_1 = (a, b, b+1)$ and $g_2 = (d, d+1, n)$, with $d < a$, or

vi) $g_1 = (a, b, b+1)$ and $g_2 = (d, d+1, n)$, with $a < d$, or

vii) $g_1 = (a, a+1, c)$ and $g_2 = (d, d+1, n)$, with $d < a$, or

viii) $g_1 = (a, a+1, c)$ and $g_2 = (d, d+1, n)$, with $d < a$, or

ix) $g_1 = (a, b, b+1)$ and $g_2 = (a, a+1, n)$, with $a+1 < b$.

Moreover, the hypergraph associated to the ideal I_{g_1,g_2} is the unique realization of π_{g_1,g_2} (up to isomorphism).

Proof. The choice of generators that allows to reduce the number of distinct degrees, still keeping them non-comparable, comes from an exhaustive check of all the possible cases. The uniqueness property follows by the fact that $\pi_{g_1,g_2} \in \mathcal{D}$. We provide here the sequence s that realizes the degree sequence in each case:

i) $s = (3^a, 2^{d-a}, 0^{e-d}, -1^{n-e})$,

ii) $s = (4^a, 2^{d+1-a}, -1^{b-d-1}, -2^{n-b})$,

iii) $s = (6^d, 1^{a+2-d}, -2^{e-a-2}, -3^{n-e})$,

iv) $s = (7^d, 1^{a-d}, 0^{b+1-a}, -3^{n-b-1})$,

v) $s = (3^{d+1}, 1^{a-d-1}, 0^{b+1-a}, -4^{n-b-1})$,

vi) $s = (3^a, 2^{d+1-a}, -1^{b-d}, -2^{n-b-1})$,

vii) $s = (5^d, 1^{a+1-d}, -1^{c-a-1}, -2^{n-c})$,

viii) $s = (4^{d+1}, 2^{a-d}, -2^{c-a-1}, -6^{n-c})$,

ix) $s = (4^a, 0, -1^{b-a}, -3^{n-b-1})$.

□

Remark 3.3.22. Differently from the previous case, now the triplets of generators can have common elements. In particular, $a = d$ in case *ix*).

Degree sequences containing five distinct values

We continue our analysis with the case of five distinct entries in a degree sequence of type π_{g_1,g_2}. For the first time, we are not able to prove the uniqueness property, since in some cases such sequences belong to the class $\mathcal{D}^{ext} \setminus \mathcal{D}$, where uniqueness is not guaranteed (see Example 3.1.11 in the introduction).

Theorem 3.3.23 ([10]). The degree sequence π_{g_1,g_2} has exactly five distinct entries if and only if g_1 and g_2 are chosen as follows:

i) $g_1 = (a, b, n)$ and $g_2 = (d, e, n)$, with $a < d < e < b$, or

ii) $g_1 = (a, b, b+1)$ and $g_2 = (d, e, n)$, with $d < a < b+1 < e$, or

iii) $g_1 = (a, b, b+1)$ and $g_2 = (d, e, n)$, with $d < a < e < b+1$, or

iv) $g_1 = (a, b, b+1)$ and $g_2 = (d, e, n)$, with $d < e < a$, or

v) $g_1 = (a, b, b+1)$ and $g_2 = (d, e, n)$, with $a < d < e < b$, or

vi) $g_1 = (a, a+1, c)$ and $g_2 = (d, e, n)$, with $d < e < a$, or

vii) $g_1 = (a, a+1, c)$ and $g_2 = (d, e, n)$, with $d < a < e < c$, or

viii) $g_1 = (a, a+1, c)$ and $g_2 = (d, e, n)$, with $d < a < c < e$, or

ix) $g_1 = (a, b, c)$ and $g_2 = (d, n-1, n)$, with $d < a$, or

x) $g_1 = (a, b, c)$ and $g_2 = (d, d+1, n)$, with $d < a < b$, or

xi) $g_1 = (a, b, c)$ and $g_2 = (d, d+1, n)$, with $a < d < b-1$, or

xii) $g_1 = (a, b, c)$ and $g_2 = (d, b, n)$, with $d < a$ and at most two consecutive elements in g_1, or

xiii) $g_1 = (a, b, n)$ and $g_2 = (d, e, n)$, with $d < a < b < e$ and no consecutive elements in g_1 and g_2, or

xiv) $g_1 = (a, b, c)$ and $g_2 = (a, e, n)$, with $e < b$ and no consecutive elements in g_1 and g_2.

Moreover, if $b \neq e$, then the hypergraph associated to the ideal I_{g_1,g_2} is the unique realization of π_{g_1,g_2} (up to isomorphism).

Proof. The choice of generators that allows to reduce the number of distinct degrees, still keeping them non-comparable, comes from an exhaustive check of all the possible cases. The uniqueness property, apart from case *xii*), follows by the fact that $\pi_{g_1,g_2} \in \mathcal{D}$. We provide here the sequence s that realizes the degree sequence in each case:

i) $s = (6^a, 4^{d-a}, 0^{e-d}, -2^{b-e}, -3^{n-b})$,

ii) $s = (6^d, 1^{a-d}, 0^{b+1-a}, -2^{e-b-1}, -3^{n-e})$,

iii) $s = (7^d, 3^{a-d}, 0^{e-a}, -1^{b+1-e}, -6^{n-b-1})$,

iv) $s = (5^d, 3^{e-d}, 1^{a-e}, 0^{b+1-a}, -6^{n-b-1})$,

v) $s = (7^a, 6^{d-a}, -1^{e-d}, -3^{b+1-e}, -4^{n-b-1})$,

vi) $s = (4^d, 3^{e-d}, 2^{a+1-e}, -2^{c-a-1}, -6^{n-c})$,

vii) $s = (5^d, 3^{e-d}, 2^{a+1-e}, -3^{c-a-1}, -7^{n-c})$,

viii) $s = (6^d, 1^{a+1-d}, -1^{c-a-1}, -2^{e-c}, -3^{n-e})$,

ix) $s = (9^d, 2^{a-d}, 0^{b-a}, -1^{c-b}, -4^{n-c})$,

x) $s = (4^{d+1}, 3^{a-d-1}, 0^{b-a}, -2^{c-b}, -7^{n-c})$,

xi) $s = (4^a, 2^{d+1-a}, -1^{b-d-1}, -2^{c-b}, -3^{n-c})$,

xiii) $s = (6^d, 4^{a-d}, 0^{b-a}, -2^{e-b}, -3^{n-e})$,

xiv) $s = (4^a, 0^{e-a}, -1^{b-e}, -2^{c-b}, -3^{n-c})$.

□

Remark 3.3.24 ([10]). Actually, it is possible to show that π_{g_1,g_2}, with g_1, g_2 as in case *xii*) of Theorem 3.3.23, admits a unique realization in the following two cases:

$b < a + 3$**:** $\pi_{g_1,g_2} = \pi_s$ with $s = (5^d, 3^{a-d}, 2^{b-a}, -4^{c-b}, -6^{n-c})$,

$b \geq a + 3$ **and** $d = a + 1$**:** $\pi_{g_1,g_2} = \pi_s$ with $s = (9^d, 6, 0^{b-a}, -5^{c-b}, -7^{n-c})$.

We conclude this section with the following theorem, that summarizes all the obtained results in case of five distinct values:

Theorem 3.3.25 ([10]). Let π_{g_1,g_2} be a five distinct values degree sequence realized by an ideal with two generators. Then, $\pi_{g_1,g_2} \in \mathcal{D}^{ext} \setminus \mathcal{D}$ if and only if $g_1 = (a, b, c)$ and $g_2 = (d, b, n)$, with $b \geq a + 3$ and $a \geq d + 2$.

We remind that we do not have any uniqueness result concerning the class $\mathcal{D}^{ext} \setminus \mathcal{D}$. Indeed, a unique realization could hold even in the case described in Theorem 3.3.25, although a rigorous proof is still missing.

Degree sequences containing six distinct values

We conclude mentioning two results from [10] for the general case of six values degree sequences, that characterize which of them belong or not to the class $\mathcal{D}$.

Theorem 3.3.26 ([10]). Let π_{g_1,g_2} be a six distinct values degree sequence with generators $g_1 = (a, b, c)$ and $g_2 = (d, e, n)$. The uniqueness of π_{g_1,g_2} is guaranteed only if one of the following conditions holds:

i) $d < a < b < c < e$, or

ii) $d < e < a < b < c$, or

iii) $a < d < e < b < c$.

Theorem 3.3.27 ([10]). Let π_{g_1,g_2} be the degree sequence associated to $I_{g_1,g_2} = \downarrow\{g_1, g_2\}$, with $g_1 = (a, b, c)$ and $g_2 = (d, e, n)$. Then, $\pi_{g_1,g_2} \in \mathcal{D}^{ext} \setminus \mathcal{D}$ if one of the following cases occur:

i) $d < a < b < e < c$ and $c \geq e + 2$, $a \geq d + 2$, $b \geq a + 2$, or

ii) $d < a < e < b < c$ and $c \geq b + 2$, $a \geq d + 2$, $b \geq e + 2$.

3.4 A randomized approach for the reconstruction of uniform hypergraphs

We dedicate this section to the study of the more general case of k-uniform hypergraphs, approaching the consistency and reconstruction problems introducing randomized strategies. We make use of the *probabilistic method*. This method is used to prove the existence of a mathematical object with prescribed characteristics, and consists in showing that choosing at random an element from a specified class, the probability of getting the desired structure is strictly greater than zero. Although this procedure is not constructive in general, the existence of the object is guaranteed by the fact that there exists at least one with the required characteristics in the specified class (see e.g. [1]).
An example of application of the probabilistic method in the field of Graph Theory is the *configuration model*. Such a model generates random graphs with a prescribed degree sequence ([22, 51]), and consists in the following: given $\pi = (d_1, \ldots, d_n)$ an integer sequence, d_i *half-edges* are assigned to the vertex v_i, for $i = 1, \ldots, n$. Then, two half-edges are picked uniformly at random to form an edge of the graph, and then the procedure is repeated until all the half-edges are gone. The obtained graph is clearly a realization of π, although it is not simple in general (both loops and repeated edges can be created during the process). As a consequence, it is interesting to study the probability of getting a simple graph realizing π, as done in [52], looking for hypothesis on the sequence π that guarantee such a realization.
In [29] and [30], the configuration model has been generalized to generate k-uniform hypergraphs with prescribed degree sequence, and recently similar techniques have been applied for the random generation of k-hypergraphs with prescribed degree sequence,

using a bijection between them and bipartite graphs [37]. Both models allow to prove that a k-uniform realization of $\pi = (d_1, \ldots, d_n)$ exists, under certain conditions. In particular, the configuration model works under the assumption that $d_1 = O(\log n)$, while the second approach is less restrictive, requiring $d_1 = o(\min\{\sigma^{1/2}, \sigma^{1-2/k}\})$ only. We improve the latter result through the definition of a different randomized algorithm, reaching the condition $d_1 = o(\sigma^{1-2/k})$.

3.4.1 Balls and boxes

Let us consider $k \geq 3$ and an integer sequence $\pi = (d_1, \ldots, d_n)$ for which we want to find a k-realization H. Starting from the idea that lies under the configuration model, one can think to equip each vertex of H with a number of $1/k$-edges equal to the desired degree, and then pick k of them uniformly at random to construct a hyperedge. Intuitively, when k grows, following this strategy it becomes more likely to produce loops instead of repeated edges. For this reason, we develop an algorithm that constructs a k-realization of π avoiding loops, and then we analyze the probability of the presence of multiple edges. We show that the probability of constructing a hypergraph realizing π that is also *simple* is strictly greater than zero, under certain assumptions on the sequence π. According to the probabilistic method, this is a proof that π is k-graphic.
To simplify the notation, in this section sometimes we will denote the vertices by $V = \{1, \ldots, n\}$. Moreover, given $\pi = (d_1, \ldots, d_n)$ and $k \geq 3$, we work under the hypothesis that k divides $\sigma = \sum_{i=1}^{n} d_i$ and $d_i \leq \frac{\sigma}{k}$ for all $i = 1, \ldots, n$, two trivial but necessary conditions for a sequence π to be k-graphic.

General idea

We set our algorithm by modeling the problem with balls and boxes. More precisely, following the philosophy of the configuration model, for each vertex v_i we consider d_i balls, labeled with i, for $i = 1, \ldots, n$. They can be seen as the equivalent of the $1/k$-edges used in the other approach. Then, we model the random picking of edges through the distribution of the balls in a suitable number of boxes, from which we will draw the hyperedges to construct the k-uniform hypergraph. Summing up, the algorithm consists of the following three steps:

Step 1. Given an integer sequence $\pi = (d_1, \ldots, d_n)$ and an integer $k \geq 3$, consider d_i balls with label i, for $i = 1, \ldots, n$, and $k + 1$ boxes, with labels $1, \ldots, k + 1$. Each box has volume equal to $\frac{\sigma}{k}$, meaning that can contain $\frac{\sigma}{k}$ balls at most.

Step 2. Distribute the balls into the boxes, keeping all the balls with the same label in the same box, and without overshooting their volume.

Step 3. Consider the k boxes that contain the highest number of balls (in case of equal cardinality, choose the one with highest label). Draw one ball from each box, uniformly at random. The labels of the chosen balls correspond to the vertices that constitute the first edge of the hypergraph we are constructing. Repeat this process until all the boxes are empty.

If the process correctly finishes, the output will be a k-uniform hypergraph realizing the input sequence π, and where no loops occur. The avoidance of loops is due to the distribution that has been made in Step 2, where we require to have all the balls with equal label in the same box, so that they cannot be drawn at the same time.

Example 3.4.1. Let us consider the integer sequence $\pi = (5, 3, 3, 2, 2, 2, 1)$ and the value $k = 3$. We describe how the algorithm we gave works on input (π, k).

Step 1. We set five balls having label equal to 1, three balls having labels 2 and 3, two balls having labels 4, 5 and 6, and one ball having label 7. Furthermore, we consider four boxes, B_1, B_2, B_3 and B_4.

Step 2. According to the greedy strategy described in Algorithm 5, we distribute the balls in the boxes, getting

$$\begin{aligned} B_1 &= \{1, 1, 1, 1, 1\}, \\ B_2 &= \{2, 2, 2, 6, 6\}, \\ B_3 &= \{3, 3, 3, 7\}, \\ B_4 &= \{4, 4, 5, 5\}. \end{aligned}$$

Step 3. We start drawing edges. At the beginning, the three boxes having highest cardinality, giving priority to highest labels in case of same value, are B_1, B_2 and B_4. We draw one ball from each of them, uniformly at random, to construct the first hyperedge of the hypergraph. Let us suppose to obtain $e_1 = (1, 2, 5)$.

At the second round, the boxes with highest cardinality are B_1, B_2 and B_3. We pick at random one element from each of them to get the second hyperedge, say $e_2 = (1, 3, 6)$.

We iterate the procedure until all the edges of the hypergraph have been sampled.

As an example, a possible output on input (π, k) is the hypergraph H having edges, in order of sampling, equal to

$$\begin{aligned} e_1 &= (1, 2, 5) \\ e_2 &= (1, 3, 6) \\ e_3 &= (3, 5, 6) \\ e_4 &= (1, 3, 4) \\ e_5 &= (1, 2, 4) \\ e_6 &= (1, 2, 7). \end{aligned}$$

Notice that:

- ▷ No loops occur in H. This is due to the fact that we put all the balls having same label in the same box, see Step 2. This condition immediately ensures that the same vertex cannot be drawn twice when sampling an edge.

- ▷ The output H is a 3-uniform hypergraph having degree sequence equal to π. Even in this case, such characteristics are guaranteed by construction. Indeed, the hypergraph is uniform since we always sample edges having same cardinality, in this case equal to $k = 3$ (Step 3). Moreover, the degree sequence of H is π, since in

Step 1 we set exactly one ball for each occurrence of a vertex that we want in the output, for all seven vertices.

- ▷ H is a simple hypergraph, so a realization of π where no loops and multiple edges occur. The avoidance of loops is guaranteed by construction, while multiple edges could be sampled, in general.

Some doubts immediately arise:

i) Why do we set $k+1$ boxes instead of k, that seems to be a more natural choice?

ii) Is it possible to distribute the balls in the boxes satisfying all the conditions required in Step 2?

iii) Does the process always terminate, meaning that we do not get stuck with two (or more) boxes that become empty at the same time?

iv) How likely is to draw the same edge more than once?

The answer to *i*) lies in the fact that setting k boxes would result, in Step 2, in filling each box with exactly $\frac{\sigma}{k}$ balls. Since we want to put all the balls with equal label in the same box, we are actually asking to solve an instance of *multi-way numbering partition*, a problem that is known to be NP-hard [45].

In Theorem 3.4.6, we positively answer to point *ii*), while we can fulfill the request of point *iii*) by adding some hypothesis on the entries of π.

We also introduce the following formal definition to describe a distribution of balls satisfying the required conditions:

Definition 3.4.2. Given $n \in \mathbb{N}$ and $\pi = (d_1, \dots, d_n)$, we define $\text{Allocation}_{k+1}(\pi)$ as the set of all $(k+1)$-tuples $(B_1, \dots, B_{k+1})$ of pairwise disjoint multisets $B_1, \dots, B_{k+1}$ such that each label $i = 1, \dots, n$ is contained exactly d_i times in one of the multisets, and $\sigma/k \geq |B_1| \geq \dots \geq |B_{k+1}|$.

Finally, we can investigate the probability of getting multiple edges and then answer to *iv*).

Implementation

We provide the pseudo-code of our random strategy in Algorithm 7, that is actually a concatenation of Step 2, implemented in Algorithm 5, *Allocation*, and Step 3, implemented in Algorithm 6, *Sampling*.

Before showing the correctness of *SampleHypergraph*, we analyze its computational cost:

Theorem 3.4.3 ([12])**.** The algorithm *SampleHypergraph* has a computational cost of $O(kn + \sigma)$, so polynomial in the size of the input.

Proof. The algorithm *SampleHypergraph* is actually the concatenation of *Allocation* and *Sampling*. As arises from Algorithms 5 and 6, respectively, the first one has a computational cost of $O(\ell n)$, the latter of $O(k\frac{\sigma}{k}) = O(\sigma)$. As a result, the total cost of our strategy is $O(kn + \sigma)$. □

Algorithm 5. *Allocation*(π, ℓ)

Input: $\pi = (d_1, \ldots, d_n)$ a non-increasing sequence of natural numbers, $\ell \in \mathbb{N}$

```
Set B_1, ..., B_ℓ = ∅;
for i = 1, ..., n do
    Let J ⊆ {1, ..., ℓ} be such that |B_j| is minimal for all j ∈ J;
    j_min = min(J);
    Add d_i copies of the vertex i to B_{j_min};
end
```

Output: $(B_1, \ldots, B_\ell)$

Algorithm 6. *Sampling*$(B_1, \ldots, B_{k+1})$

Input: $(B_1, \ldots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$

```
Set E = ∅;
for i = 1, ..., σ/k do
    Let J ⊆ {1, ..., k+1} be such that |B_j| is minimal for all j ∈ J;
    j_min = min(J); edge = ∅;
    for ℓ = 1, ..., k+1, ℓ ≠ j_min do
        choose b_ℓ uniformly at random from B_ℓ;
        edge = edge ∪ {b_ℓ};
        B_ℓ = B_ℓ \ {b_ℓ};
    end
    E = E ∪ {edge};
end
```

Output: E

Algorithm 7. *SampleHypergraph*(π, k)

Input: $\pi = (d_1, \ldots, d_n)$ a non-increasing sequence of natural numbers, $k \geq 3$

```
(B_1, ..., B_{k+1}) = Allocation(π, k+1);
Relabel B_1, ..., B_{k+1} such that they are decreasing in size;
if (B_1, ..., B_{k+1}) ∉ Allocation_{k+1}(π) then
    return (Error)
end
E=Sampling(B_1, ..., B_{k+1});
```

Output: E

3.4.2 Correctness and integer sequences that are k-graphic

To show the correctness of *SampleHypergraph*, we first need to show that *Allocation* returns a suitable input for *Sampling*, so a set $(B_1, \ldots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$, and then that the algorithm *Sampling* actually terminates without errors.
We start with a lemma that provides a bound for the cardinalities of the boxes obtained as output of *Allocation*.

Lemma 3.4.4 ([12]). For $\pi = (d_1, \ldots, d_n)$ and $\ell \in \mathbb{N}$, the algorithm *Allocation*(π, ℓ) yields

$$|B_i| \leq \max\left\{d_1, \frac{\sigma}{\ell} + d_{\ell+1}\right\}$$

for all $i = 1, \dots, \ell$.

Proof. First of all, we observe that *Allocation* is a greedy algorithm: the boxes are filled in order, starting from the emptiest one and giving priority to lower indices. So, in the first ℓ steps, the boxes $B_1, \dots, B_\ell$ are filled with the balls with labels $1, \dots, \ell$, respectively. Now, if all the vertices are distributed equally, we would get exactly $\frac{\sigma}{\ell}$ balls in each box. In case $d_1 \geq \frac{\sigma}{\ell}$, we get d_1 as an upper bound on the cardinalities $|B_i|$, after the first ℓ steps. Since now $d_{\ell+1}$ is the highest degree left, and all boxes still contain less than $\frac{\sigma}{\ell}$ elements, their cardinality can overshoot $\frac{\sigma}{\ell}$ by at most $d_{\ell+1}$, giving the thesis. □

We proceed with the analysis of the algorithm *Sampling*. For the moment, we suppose to have in input a suitable set $(B_1, \dots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$. We furthermore fix the notation $\text{Supp}(A) = \{a : a \in A\}$ to point out the distinct elements that constitute a multiset A, up to their multiplicity.

Theorem 3.4.5 ([12]). Consider $n \in \mathbb{N}$, $\pi = (d_1, \dots, d_n)$ and $k \geq 3$. For $(B_1, \dots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$, we have that

i) the algorithm *Sampling* on input $(\pi, B_1, \dots, B_{k+1})$ terminates,

ii) provides a k-hypergraph H without loops and having degree sequence equal to π,

iii) and

$$\begin{aligned}&\mathbf{P}(H \text{ has no parallel edges})\\ &\geq 1 - \sum_{\ell=1}^{k+1} \min_{j \in \{1,\dots,k+1\} \setminus \{\ell\}} \frac{|B_j|(|B_j| - 1)}{2} \prod_{i \in \{1,\dots,k+1\} \setminus \{\ell\}} \frac{\max_{u \in \text{Supp}(B_i)} d_u}{|B_i|}.\end{aligned}$$

Proof. We proceed claim by claim.

i) To show that the algorithm terminates, we have to prove that we do not get stuck reaching two empty boxes at the same time, meaning that we can continue to draw edges until all the balls are gone. First of all we notice, by Definition 3.4.2 and line 4 in Algorithm 6, that B_{k+1} is always among the boxes containing the fewest elements. We now show that, as soon as B_{k+1} runs empty, all the other boxes contain exactly one ball, so that *Sampling* terminates with no errors.

By Definition 3.4.2, $\sigma/k \geq |B_1| \geq \dots \geq |B_{k+1}|$ holds, so there exist $r_1, \dots, r_k \geq 0$ such that

$$|B_i| = \sigma/k - r_i \text{ for } i = 1, \dots, k \qquad \text{and} \qquad |B_{k+1}| = \textstyle\sum_{i=1}^k r_i.$$

We also define

$$\Delta = \sum_{i=2}^{k} |B_1| - |B_i| = \sum_{i=2}^{k} (r_i - r_1) \leq \sum_{i=1}^{k} r_i = |B_{k+1}|,$$

that takes into account the number of balls missing to fill the first k boxes to the height of B_1 (always the fullest one by construction).

We consider how Δ changes each time we draw balls. Two cases arise:

- *We do not draw from* B_1. If we do not draw from the first box, then this is among the boxes containing the fewest balls. But since B_1 is always the fullest one, we have already reached $\Delta = 0$.
- *We draw from* B_1. In this case, Δ reduces its value by one, since the difference between B_1 and B_j gets reduced by one, being B_j the only box from which we did not draw, while all the other differences keep the same value.

Having $\Delta \leq |B_{k+1}|$, we conclude that at a certain point we reach $\Delta = 0$, that is, we continue drawing balls from the first k boxes until $|B_1| = |B_2| = \cdots = |B_{k+1}|$.

From now on, the difference between the fullest and emptiest box is at most one and, since k divides σ, when the last box runs empty each of the remaining ones contains exactly one ball, and so *i*) is proven.

ii) The hypergraph created by *Sampling* is k-uniform and realizes π by construction. The absence of loops is guaranteed by the pairwise disjointedness of the multisets B_i, for $i = 1, \ldots, k+1$.

iii) Finally, we analyze the probability of drawing the same edge twice.

For $\ell = 1, \ldots, k+1$, we denote $E_\ell = \{e_i : i = 1, \ldots, |E_\ell|\}$ the list of all the edges that do not contain a vertex from B_ℓ, ordered according to their sampling. From the pairwise disjointedness of $B_1, \ldots, B_{k+1}$, it follows that two lists E_i and E_j, with $i \neq j$, do not share any edge. If A_ℓ is the event that some edge occurs twice in E_ℓ, for $\ell = 1, \ldots, k+1$, then

$$\mathbf{P}(H \text{ has no parallel edges}) \geq 1 - \sum_{\ell=1}^{k+1} \mathbf{P}(A_\ell). \tag{3.3}$$

We analyze $\mathbf{P}(A_\ell)$ for a fixed index $\ell = 1, \ldots, k+1$, assuming $A_\ell \neq \emptyset$. Then,

$$\mathbf{P}(A_\ell) \leq \sum_{1 \leq i < j \leq |E_\ell|} \mathbf{P}(e_i = e_j). \tag{3.4}$$

At first, we consider all the elements in B_i as distinguishable, and denote a pair of edges listing the vertices that constitute them,

$$\begin{aligned} e_i &= (f_1, \ldots, f_{\ell-1}, f_{\ell+1}, \ldots, f_{k+1}), \\ e_j &= (g_1, \ldots, g_{\ell-1}, g_{\ell+1}, \ldots, g_{k+1}), \end{aligned}$$

with $f_s, g_s \in B_s$ and $f_s \neq g_s$, for all $s = 1, \ldots, \ell-1, \ell+1, \ldots, k+1$. Since we sample balls uniformly at random, we have

$$\mathbf{P}(e_i = f, e_j = g) = \prod_{s \in \{1,\ldots,k+1\} \setminus \{\ell\}} \frac{1}{|B_s|(|B_s| - 1)}.$$

We now have to take into account that balls with same label are actually indistinguishable, and so consider the number of their copies, given by the degrees d_i. Then, the probability of drawing the same edge h is

$$\mathbf{P}(e_i = e_j = h) = \prod_{s \in \{1,\dots,k+1\} \setminus \{\ell\}} \frac{d_{h_s}(d_{h_s} - 1)}{|B_s|(|B_s| - 1)}.$$

Replacing in (3.4) and taking into account symmetry, we obtain

$$\begin{aligned} \mathbf{P}(A_\ell) &\leq \frac{|E_\ell|(|E_\ell| - 1)}{2} \mathbf{P}(e_1 = e_2) \\ &= \frac{|E_\ell|(|E_\ell| - 1)}{2} \prod_{i \in \{1,\dots,k+1\} \setminus \{\ell\}} \sum_{j \in \mathrm{Supp}(B_i)} \frac{d_j(d_j - 1)}{|B_i|(|B_i| - 1)}. \end{aligned}$$

Finally, we observe that $|E_\ell| \leq \min_{j \in \{1,\dots,k+1\} \setminus \{\ell\}} |B_j|$, since all the edges in E_ℓ must contain vertices from all B_i, for $i \neq \ell$. Moreover, for all $i = 1, \dots, k+1$, it holds

$$\begin{aligned} \sum_{j \in \mathrm{Supp}(B_i)} d_j(d_j - 1) &\leq \max_{u \in \mathrm{Supp}(B_i)} d_u \sum_{j \in \mathrm{Supp}(B_i)} (d_j - 1) \\ &\leq \max_{u \in \mathrm{Supp}(B_i)} d_u(|B_i| - 1). \end{aligned}$$

Replacing all the inequalities in (3.3) yields *iii*). □

We are now ready to characterize the integer sequences for which the algorithm *Sample-Hypergraph* returns a realization, with no multiple edges, with probability strictly greater than zero. So, according to the probabilistic method, we actually characterize a class of k-graphic sequences, thus solving for them the consistency problem in polynomial time.

Theorem 3.4.6 ([12]). For $n \in \mathbb{N}$ and $k \geq 3$, let $\pi = (d_1, \dots, d_n)$ be an integer sequence such that $k(k+1)d_{k+2} \leq \sigma$. Then, there exists a polynomial time randomized algorithm that always returns a k-hypergraph H with degree sequence equal to π, and satisfies

$$\mathbf{P}(H \text{ is simple}) \geq 1 - \frac{k+1}{2} \left(\frac{3k}{2}\right)^{k-2} \frac{d_1^k}{\sigma^{k-2}}.$$

Proof. We consider *SampleHypergraph* as the candidate randomized algorithm.
From Theorem 3.4.5, we can prove that *SampleHypergraph* returns a suitable output if we are able to show that $(B_1, \dots, B_{k+1}) \in \mathrm{Allocation}_{k+1}(\pi)$. In fact, we only need to prove that $|B_1| \leq \sigma/k$. According to Lemma 3.4.4, choosing $\ell = k+1$, we have

$$|B_1| \leq \max\left\{d_1, \frac{\sigma}{k+1} + d_{k+2}\right\}.$$

By our hypothesis on π, we know that $d_1 \leq \sigma/k$ (trivial), and

$$\frac{\sigma}{k+1} + d_{k+2} \leq \frac{\sigma}{k+1} + \frac{\sigma}{k(k+1)} = \frac{\sigma}{k}.$$

Then, $|B_1| \leq \sigma/k$ and $(B_1, \dots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$, as required.

We now compute the lower bound for $\mathbf{P}(H \text{ is simple})$. From the lower bound provided in Theorem 3.4.5, and being $d_1 \geq \cdots \geq d_n$ and $|B_1| \geq \cdots \geq |B_{k+1}|$, we have

$$\begin{aligned}
&\mathbf{P}(H \text{ has no parallel edges}) \\
&\geq 1 - \sum_{\ell=1}^{k+1} \min_{j \in \{1,\dots,k+1\} \setminus \{\ell\}} \frac{|B_j|(|B_j|-1)}{2} \prod_{i \in \{1,\dots,k+1\} \setminus \{\ell\}} \frac{\max_{u \in \text{Supp}(B_i)} d_u}{|B_i|} \\
&\geq 1 - \frac{k+1}{2} \frac{d_1^k}{|B_{k-1}|^{k-2}}.
\end{aligned}$$

Finally, since each box contains σ/k elements at most, we have at least $\sigma - (k-2)\frac{\sigma}{k}$ vertices to be distributed in the last three boxes. Being B_{k-1} the fullest one among these, it holds

$$|B_{k-1}| \geq \frac{1}{3}\left(\sigma - (k-2)\frac{\sigma}{k}\right) = \frac{2\sigma}{3k}. \tag{3.5}$$

Replacing in the previous formula, we finally get

$$\mathbf{P}(H \text{ is simple}) \geq 1 - \frac{k+1}{2}\left(\frac{3k}{2}\right)^{k-2} \frac{d_1^k}{\sigma^{k-2}},$$

and so the lower bound is proven.

Finally, from Theorem 3.4.3, we have that *SampleHypergraph* runs in polynomial time in the size of the input, and this concludes the proof. □

Remark 3.4.7. Notice that if $d_1^k = o(\sigma^{k-2})$, then $\mathbf{P}(H \text{ is simple}) \to 1$ as $n \to \infty$.

From Remark 3.4.7, two corollaries immediately follow, that definitively individuate two classes of k-graphic degree sequences:

Corollary 3.4.8 ([12])**.** Let $k \geq 3$ and $\pi = (d_1, \dots, d_n)$. If $d_1 \leq Cn^\alpha$ for $C > 0$ and $\alpha < 1 - \frac{2}{k}$, then $\mathbf{P}(H \text{ is simple}) \to 1$ as $n \to \infty$.

Corollary 3.4.9 ([12])**.** Let $k \geq 3$ and $\pi = (d_1, \dots, d_n)$, and define $\rho := d_1/d_n$. If

$$d_1^2 \left(\frac{\rho}{n}\right)^{k-2} \to 0,$$

then $\mathbf{P}(H \text{ is simple}) \to 1$ as $n \to \infty$.

From the above results, we argue that an integer sequence π is k-graphic whether its degrees are sufficiently small (Corollary 3.4.8) or sufficiently close to each other (Corollary 3.4.9). We also underline that *SampleHypergraph* is a constructive algorithm. Indeed, Theorem 3.4.6 and Corollaries 3.4.8 and 3.4.9 provide theoretical results for the characterization of some classes of k-graphic sequences: it is sufficient to check if the entries of π satisfy the hypothesis of Theorem 3.4.6, and this can be done immediately (precisely, in linear time). On the other hand, it is possible to run *SampleHypergraph* on π and get, again in polynomial time, a realization of the degree sequence, that is a simple hypergraph with a positive probability tending to 1 as the length of π tends to infinity.

3.4.3 A variation that ignores higher degrees

We can summarize the results reached in the previous section stating that

Result: Given a non-increasing integer sequence $\pi = (d_1, \dots, d_n)$ and $k \geq 3$, if $d_1 = o(\sigma^{1-2/k})$, then π is k-graphic.

So, the value of the highest entry in π is fundamental to check, using Theorem 3.4.6, if sufficient conditions for a sequence to be k-graphic are satisfied.
In some cases, such hypothesis are too restrictive. We exhibit an example borrowed from [12]:

Example 3.4.10. We consider the integer (after proper roundings) sequence

$$\pi := \Bigg(\underbrace{\frac{n}{\log^3(n)}, \dots, \frac{n}{\log^3(n)}}_{\log(n)}, \underbrace{\frac{\sqrt{n}}{\log(n)}, \dots, \frac{\sqrt{n}}{\log(n)}}_{n-\log(n)} \Bigg), \tag{3.6}$$

and $k = 4$. We check if the condition $d_1 = o(\sigma^{1-2/k})$, see Remark 3.4.7, is satisfied.
We have $\sigma \approx \frac{n^{3/2}}{\log(n)}$, and so $d_1 \neq o\left(\sigma^{1-2/k}\right) = o\left(\frac{n^{3/4}}{\sqrt{\log(n)}}\right)$. Hence, Theorem 3.4.6 cannot be applied to establish if π is 4-graphic.

So, we provide a slight variation of the algorithm *SampleHypergraph* that allows to ignore the higher entries of π, still yielding sufficient conditions for a sequence to be k-graphic.

Algorithm 8 describes the pseudo-code of *SampleHypergraph2*, used in scenarios with $k \geq 4$. It consists in allocating the smaller entries of the input sequence π in four boxes, and then use the remaining $k-3$ for the allocation of the higher degrees. Then, boxes are reordered w.r.t their cardinality, and edges are drawn using *Sampling*.
It seems plausible that having many small degrees helps avoiding the construction of parallel edges. We reserve four boxes for them, so that it is less likely to draw many small degrees in the same hyperedge.

Algorithm 8. *SampleHypergraph2*(π, k, m)

Input: $\pi = (d_1, \dots, d_n)$ a non-increasing sequence of natural numbers,
$k \geq 4, m \in \{1, \dots, n\}$

```
(B_1, ..., B_4)=Allocation((d_m, ..., d_n), 4);
(B_5, ..., B_{k+1})=Allocation((d_1, ..., d_{m-1}), k-3);
Relabel B_1, ..., B_{k+1} such that they are decreasing in size;
if (B_1, ..., B_{k+1}) ∉ Allocation_{k+1}(π) then
    return (Error)
end
E=Sampling(B_1, ..., B_{k+1});
```

Output: E

The following theorem provides the correctness of *SampleHypergraph2*, as well as sufficient conditions on π that guarantee to reach a suitable realization with probability strictly greater than zero.

Theorem 3.4.11 ([12]). For $k \geq 4$, $n \in \mathbb{N}$ and $\pi = (d_1, \ldots, d_n)$, let $m \in \{1, \ldots, n\}$ be maximal with

$$\frac{4\sigma}{k+1} \leq \sum_{i=m}^{n} d_i.$$

If $k(k+1)d_{k-2} \leq \sigma$ and $5k(k+1)d_m \leq 4\sigma$, then there exists a polynomial time randomized algorithm that always returns a k-hypergraph H with degree sequence π, and such that

$$\mathbf{P}(H \text{ is simple}) \geq 1 - \frac{3k(k+1)}{4} \frac{d_m^3}{\sigma}.$$

Proof. The proof follows the same idea carried on in the proof of Theorem 3.4.6, and we consider *SampleHypergraph2* as the candidate randomized algorithm.
We start showing that the hypothesis on π are sufficient to get a suitable output from lines 1-3 of Algorithm 8, meaning $(B_1, \ldots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$, and so that *SampleHypergraph2* correctly terminates providing a hypergraph H with degree sequence π (by Theorem 3.4.5). We only need to prove that $|B_1| \leq \sigma/k$. Two cases arise:

- *B_1 is generated in line 1 of Algorithm 8.* By Lemma 3.4.4,

$$|B_1| \leq \max\left\{d_m, \frac{\sum_{i=m}^{n} d_i}{4} + d_{m+4}\right\}.$$

  Moreover, by hypothesis, we have $d_m \leq d_1 \leq \sigma/k$, with m the maximum index s.t. $\sum_{i=m}^{n} d_i \geq \frac{4\sigma}{k+1}$, and so

$$\sum_{i=m+1}^{n} d_i < \frac{4\sigma}{k+1}.$$

  We finally get

$$\frac{\sum_{i=m}^{n} d_i}{4} + d_{m+4} \overset{d_{m+4} \leq d_m}{<} \frac{\sigma}{k+1} + \frac{5d_m}{4} \leq \frac{\sigma}{k+1} + \frac{\sigma}{k(k+1)} = \frac{\sigma}{k}.$$

- *B_1 is generated in line 2 of Algorithm 8.* By Lemma 3.4.4,

$$|B_1| \leq \max\left\{d_1, \frac{\sum_{i=1}^{m-1} d_i}{k-3} + d_{k-2}\right\}.$$

  Using again the definition of m, together with the hypothesis on d_{k-2}, we obtain

$$\frac{\sum_{i=1}^{m-1} d_i}{k-3} + d_{k-2} = \frac{\sigma - \sum_{i=m}^{n} d_i}{k-3} + d_{k-2} \leq \frac{\sigma - \frac{4\sigma}{k+1}}{k-3} + \frac{\sigma}{k(k+1)} = \frac{\sigma}{k},$$

  and so the thesis.

We now compute the lower bound for $\mathbf{P}(H \text{ is simple})$, starting again from Theorem 3.4.5, point *iii*):

$$\begin{aligned}&\mathbf{P}(H \text{ is simple})\\ &\geq 1 - \sum_{\ell=1}^{k+1} \min_{j\in\{1,\dots,k+1\}\setminus\{\ell\}} \frac{|B_j|(|B_j|-1)}{2} \prod_{i\in\{1,\dots,k+1\}\setminus\{\ell\}} \frac{\max_{u\in\mathrm{Supp}(B_i)} d_u}{|B_i|}.\end{aligned}$$

Let a, b, c, d be the indices corresponding to the boxes generated in line 1 of Algorithm 8, so containing the balls with labels $m, m+1, \dots, n$.
In the product of the above equation, at least three indices among a, b, c, d appear, all different from ℓ. Let us denote them as x_ℓ, y_ℓ and z_ℓ. Since $\max_{u\in\mathrm{Supp}(B_i)} d_u \leq |B_i|$ for all $i = 1, \dots, k+1$, we have

$$\mathbf{P}(H \text{ is simple}) \geq 1 - \sum_{\ell=1}^{k+1} \min_{j\in\{1,\dots,k+1\}\setminus\{\ell\}} \frac{|B_j|(|B_j|-1)}{2} \frac{d_m^3}{|B_{x_\ell}||B_{y_\ell}||B_{z_\ell}|}.$$

Being $x_\ell, y_\ell, z_\ell \neq \ell$, $|B_{x_\ell}|, |B_{y_\ell}|, |B_{z_\ell}| \geq \min_{j\in\{1,\dots,k+1\}\setminus\{\ell\}} |B_j|$ must hold. Moreover, the sets are pairwise disjoint, and their maximum is larger than or equal to $|B_{k-1}|$ (by Definition 3.4.2). So, we finally obtain the desired bound,

$$\mathbf{P}(H \text{ is simple}) \geq 1 - \sum_{\ell=1}^{k+1} \frac{1}{2} \frac{d_m^3}{|B_{k-1}|} \geq 1 - \frac{3k(k+1)}{4} \frac{d_m^3}{\sigma},$$

being $|B_{k-1}| \geq \dfrac{2\sigma}{3k}$ from (3.5).
Finally, Theorem 3.4.3 guarantees a computational cost that is polynomial in the size of the input, and this concludes the proof. □

3.5 Co-degree sequences and trees

In this final section, we consider the problem of reconstructing a simple graph from its co-degree sequence. Up to our knowledge, the problem has not been investigated yet.
At first, we consider only graphs with the property that two vertices have at most one common neighbor, so whose co-degree sequence has entries equal to 0 or 1 only, i.e., $\gamma = \left(1^c, 0^{\binom{n}{2}-c}\right)$ for some $c \in [0, \binom{n}{2}]$.
From here on, the number of vertices in the graph is fixed and equal to n.

3.5.1 Trivial instances and ambiguity

We start analyzing the instances $c = 0$ and $c = \binom{n}{2}$, that may appear easy at first glance. The solution in case of $c = \binom{n}{2}$ is given by the following theorem:

Theorem 3.5.1 (Erdős et al. [39]). If G is a graph on n vertices in which any two vertices have exactly one common neighbor, then $n = 2i+1$ and G consists of i triangles which have one common vertex.

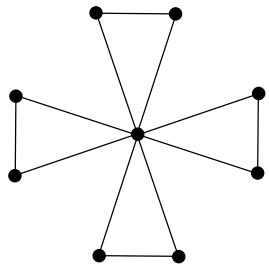


Figure 3.6: The friendship graph on 9 vertices. Each pair of vertices has exactly one common neighbor, so its co-degree sequence is $\gamma = (1^{36})$.

So, in this case the solution is unique and consists of the so called *friendship graph*, Fig. 3.6.
Moving to $c = 0$, ambiguities arise. Indeed, any graph in which the length of each path is at most equal to 1 is a solution, see Fig. 3.7.

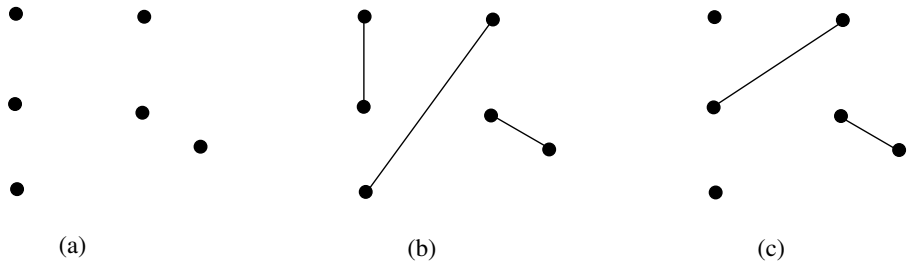


Figure 3.7: Three examples of graphs on six vertices having null co-degree sequence, $\gamma = (0^{15})$.

We conclude that the solution to our problem is not unique in general, meaning that two (or more) non-isomorphic graphs may be the solution of the same instance. A further example of such a case is shown in Fig. 3.8.

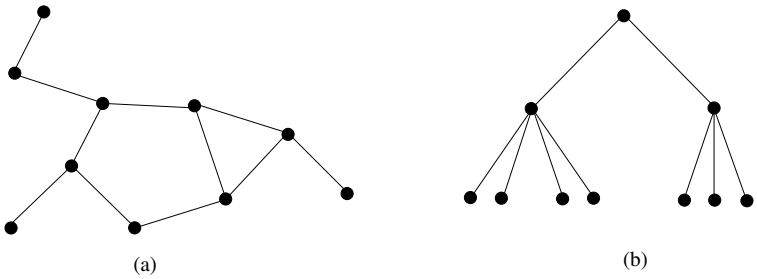


Figure 3.8: Two non-isomorphic graphs with same co-degree sequence, $\gamma = (1^{17}, 0^{28})$.

As a consequence, we decide to simplify the problem by imposing a further constraint: we ask the solution to be a tree, so a graph having a specific structure.
We denote by $T(n, c)$ an instance of the problem of finding a tree on n vertices having co-degree sequence equal to $\gamma = (1^c, 0^{\binom{n}{2}-c})$.

3.5.2 Trees

When considering trees, the parameter c may vary from $n-2$ to $\binom{n-1}{2}$ only. Indeed, we recall that trees are connected graphs having exactly $n-1$ edges, so, intuitively, the minimum of c cannot be zero, but increases with the value of n. Similarly, the upper bound changes, and reduces, again due to the fact that exactly $n-1$ edges are present in the graph. We can easily compute the new interval in which c varies, and also provide the (unique) solution to the instance $T(n,c)$ for the extreme cases.

Theorem 3.5.2. If $\gamma = (1^c, 0^{\binom{n}{2}-c})$ is the co-degree sequence of a tree on n vertices, then $n-2 \leq c \leq \binom{n-1}{2}$. Moreover, when choosing $c = n-2$ or $c = \binom{n-1}{2}$, the solution of $T(n,c)$ is unique, and consists in the first case of the path, in the latter of the star graph.

Proof. Since in trees a path connecting two vertices is unique, counting the number of vertices sharing exactly one neighbor is equivalent to count the number of pairs at distance 2. A tree on n vertices has exactly $n-1$ edges, so there are exactly $n-1$ pairs of vertices at distance 1. As a consequence,

$$|\{\text{pairs at distance} \geq 2\}| = \binom{n}{2} - (n-1) = \binom{n-1}{2},$$

and so

$$c = |\{\text{pairs at distance } 2\}| = \binom{n-1}{2} - |\{\text{pairs at distance} \geq 3\}|.$$

It follows that the maximum admitted value for c is reached when no pairs of vertices are at distance greater than or equal to 3, that is, $c_{max} = \binom{n-1}{2}$. The only tree realizing such a situation is the star, Fig. 3.9(b).
On the other hand, we show by induction that the tree on $n \geq 3$ vertices for which c reaches the minimum admitted value is the path.

Base case. The only tree (up to isomorphism) on $n = 3$ vertices is the path, where there is only one pair of vertices at distance 2. So, $c = 1 = n-2$, and it is also the minimum. Moving to $n = 4$, we add an edge to get a new tree, and this can be done in two different ways:

i) We connect the new edge to the leaf (or, equivalently, to the root). In this case, we get again the path, and we create a new pair of vertices at distance 2. So, $c' = 2$.

ii) We connect the new edge to the vertex of the path at level one. In this case, we create two new pairs of vertices at distance two, and so $c' = 3$.

It follows that the minimum for $n = 4$ is realized again by the path, and it is $c'_{min} = 2 = n-2$.

Induction step. Let us consider the path on n vertices, where there are exactly $c = n-2$ pairs of vertices at distance two, and let us suppose that this is the minimum possible value we can reach. We move to a tree on $n+1$ vertices by adding a new edge, and count the number c' of non-null co-degrees.

i) If we add the new edge to the leaf of the path (or, equivalently, to the root), we create a new pair of vertices at distance two, so $c' = c + 1$. Then, we construct again the path, on $n + 1$ vertices, and $c' = (n - 2) + 1 = (n + 1) - 2$.

ii) If we add the new edge to a vertex different from the root or the leaf, we create a tree on $n + 1$ vertices that is different from the path, and having two new pairs of vertices at distance two. So, $c' = (n - 2) + 2 = n$.

It follows that c'_{min} is realized by the path on $n + 1$ vertices, and takes value $c'_{min} = n - 1$.

From the iterative construction we presented by induction, we also deduce that the path is the unique realization of the minimum value of c, and this concludes the proof. □

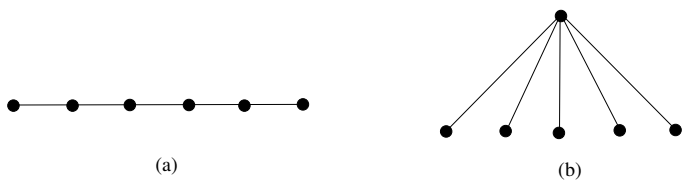


Figure 3.9: The path (a) and the star (b) on six vertices.

In a more general framework, ambiguity finds space even in case of trees.

Example 3.5.3. Choosing $n = 12$ and $c = 16$, there exist (at least) two non-isomorphic solutions to the instance $T(12, 16)$, depicted in Fig. 3.10. T_1 and T_2 are clearly non-isomorphic: as an example, they have different degree sequences.

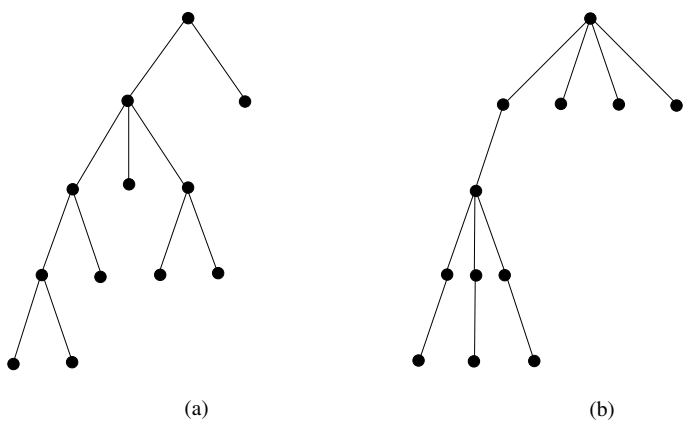


Figure 3.10: Two non-isomorphic trees in Υ_{12} having co-degree sequence $\gamma = (1^{16}, 0^{50})$.

Canonical trees

Starting from our observations about the ambiguity of reconstruction, we define a canonical structure for trees, from which we can deduce an easy formula for the computation

of their co-degree sequence, and then, starting from it, a reconstruction algorithm. To do that, we define the operation *left-shift*.

Definition 3.5.4. Given a tree T, let v_y be the leftmost leaf in T, and let us consider a node v_x at level $x \geq 1$ and not on the path from v_y to the root. Let $L(x)$ denote the set of descendants of v_x. The *left-shift operation applied on* v_x consists in moving $L(x)$ such that v_y becomes its root.

Figure 3.11 shows the application of the operation left-shift.

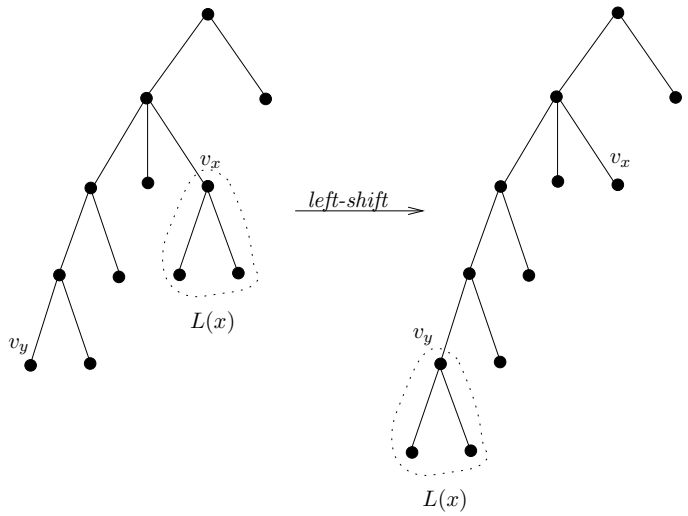


Figure 3.11: The left-shift operation applied on a node v_x at level 2. The obtained tree, on the right, is such that no further vertex satisfies the requirements needed to apply a left-shift.

Theorem 3.5.5. Given a tree T with co-degree sequence γ, let T' be obtained from T after the application of the left-shift operation. Then, T' has co-degree sequence γ, i.e., the left-shift does not change the co-degree sequence of a tree.

Proof. Let $\gamma = (1^c, 0^{\binom{n}{2}-c})$ be the co-degree sequence of T, and v_x the node on which we apply the left-shift operation to obtain T' from T. Let us consider $T_x = L(x) \cup \{v_x\}$ the sub-tree composed by v_x, its root, and all its descendants, and let us denote c_x the number of pairs of vertices in T_x sharing one neighbor. Moreover, let d be the number of children of v_x. Since v_x is at level $x \geq 1$ in T, by hypothesis, d is also the number of vertices in T_x that share the neighbor v_x with some vertex in $T \setminus T_x$, precisely with the parent of v_x. As a consequence, $c = c_x + d + d'$, where d' is the number of pairs of vertices in $T \setminus T_x$ sharing one neighbor.
Let us now consider the left-shift operation applied on v_x, consisting in moving $L(x)$ such that the leftmost leaf of T, say v_y, becomes its root. We underline that $v_y \notin L(x)$, since v_x is not on the path from v_y to the root by hypothesis. We denote T' the resulting tree. Then, being $\gamma' = (1^{c'}, 0^{\binom{n}{2}-c'})$ its co-degree sequence, the value c' can be computed following the same argument used for c, just replacing v_x with v_y. Notice that v_y is at

level $y \geq 1$ as well, since it was the leftmost leaf in T, and that the number of pairs counted by d and d', previous defined, does not change. As a consequence, $c = c'$, and the thesis follows. $\square$

We say that a tree has a *canonical structure*, or briefly is *canonical*, if it is such that the left-shift operation cannot be further applied (no suitable node is left), see Fig. 3.11.

Property 3.5.6. If T is canonical, then all its levels are composed by siblings, i.e., all the vertices on the same level have the same parent.

Property 3.5.6 is clear after observing that otherwise there would be at least one node on which the left-shift operation could be applied, see Fig. 3.11 for an example.
From Theorem 3.5.5, we deduce that if a solution to $T(n, c)$ exists, then a solution to the same instance, and having a canonical structure, exists too. So, from now on, we consider canonical trees only.

Remark 3.5.7. Given a co-degree sequence γ, two (or more) non-isomorphic canonical trees realizing it may exist, in general. An example is shown in Fig. 3.12.

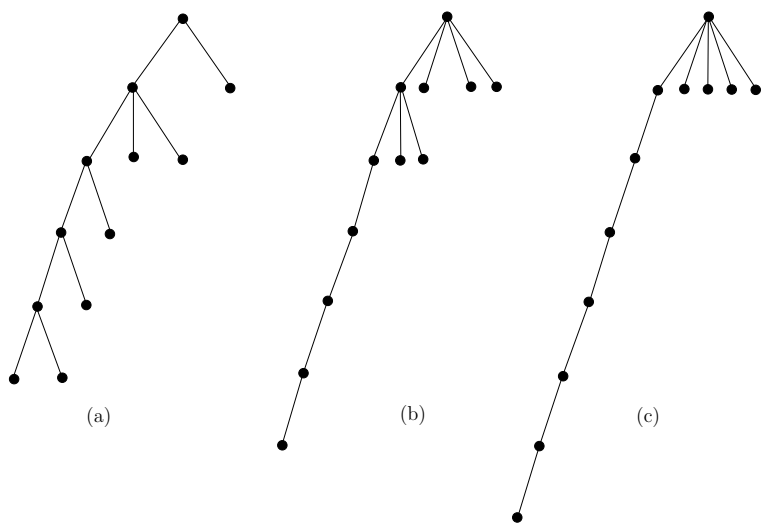


Figure 3.12: Three canonical trees with co-degree sequence $\gamma = (1^{16}, 0^{50})$. By computing the respective degree sequences, it is easy to check that they are all non-isomorphic.

Structure and co-degree sequence of canonical trees

As a consequence of the fact that if the instance $T(n, c)$ admits a solution, then a solution having canonical structure exists too, we can reformulate the co-degree reconstruction problem as follows.

Problem: given a binary integer sequence γ of length $\binom{n}{2}$, for some $n \geq 2$, determine whether there exists a canonical tree on n vertices having γ as its co-degree sequence.

Due to the structure of canonical trees, it is possible to find a closed formula to compute the value c that defines the binary sequence γ. We introduce such a formula in the next paragraph.
By Property 3.5.6, a canonical tree is characterized by a path, on the left, whose length coincides with the height l of the tree. We denote the nodes on this path, from the root to the leaf, as $r_0, \dots, r_l$. As a direct consequence of Property 3.5.6, each r_i, for $i = 0, \dots, l-1$, is exactly the parent of b_{i+1} nodes, corresponding to the siblings that constitute the $(i+1)$-th level of the tree.

Theorem 3.5.8. Let T be a canonical tree on n vertices of height l, with $b_0, b_1, \dots, b_l$ the cardinalities of its levels, and let us denote $\gamma = (1^c, 0^{\binom{n}{2}-c})$ its co-degree sequence. Then

$$c = \sum_{i=1}^{l} \binom{b_i}{2} + n - 1 - b_1.$$

Proof. The number of nodes in a tree is given by the sum of the cardinalities of its levels, $n = \sum_{i=0}^{l} b_i$. By convention, the level of the root has cardinality $b_0 = 1$, so that

$$\sum_{i=2}^{l} b_i = n - 1 - b_1. \tag{3.7}$$

We count the number of pairs of vertices sharing one neighbor. We start considering siblings: since T is canonical, each level is composed by exactly b_i siblings, for $i = 1, \dots, l$. By definition, any pair of siblings at level i has the parent r_{i-1} as common neighbor, for $i = 1, \dots, l$. Then, when considering siblings, each level contributes to c for a value

$$c_i = \binom{b_i}{2}, \text{ for } i = 1, \dots, l.\text{[1]} \tag{3.8}$$

Furthermore, on each level a node shares one neighbor, specifically its parent r_{i-1}, with one node lying on the longest path, specifically r_{i-2} (that is, the parent of its parent). Notice that level 1 is excluded from this computation, since the root, at level zero, has no parent by definition. The contribution to c given by this type of pairs is expressed by the number of the involved vertices, that is,

$$c^* = \sum_{i=2}^{l} b_i. \tag{3.9}$$

To complete our analysis, we have to consider a last type of pair: two vertices at different levels and not lying on the longest path, say $u \in b_i$ and $v \in b_j$, with $1 \le i < j \le l$. It is clear from the structure of canonical trees that the distance between u and v is $j - i + 2$, so greater than or equal to 3 for each $i \neq j$. So, this type of pair is not involved in the computation of c.

[1] With abuse of notation, we set $\binom{1}{2} = 0$.

We conclude that c is obtained as the sum of Eq. (3.8), for $i = 1, \dots, l$, and Eq. (3.9),

$$c = \sum_{i=1}^{l} \binom{b_i}{2} + \sum_{i=2}^{l} b_i.$$

Replacing (3.7) in the previous equation gives

$$c = \sum_{i=1}^{l} \binom{b_i}{2} + n - 1 - b_1,$$

and completes the proof. □

Notice that the number of children of the root, b_1, plays a special role in the computation of c. This is due to the fact that the root does not have a parent by definition.

From Theorem 3.5.8, we reformulate the reconstruction problem in an arithmetical flavor. Indeed, given two integers n and c, we have to find l suitable integers $b_1, \dots, b_l$ such that

$$c = \sum_{i=1}^{l} \binom{b_i}{2} + n - 1 - b_1.$$

We underline again that many solutions may exist, see Remark 3.5.7. This is reflected in the possibility of different choices for the parameter l, as well as for the parameter b_1, in general. Actually, in the next section we show that the parameter b_1 can always be fixed as $b_1 = 1$, while there is still room to choose l.
The reconstruction algorithm we are going to define gives priority to smaller values for l, but can be slightly modified according to other preferences.

Levels of canonical trees and their cardinalities

Let us consider a canonical tree T of height l, and its co-degree sequence $\gamma = (1^c, 0^{\binom{n}{2}-c})$. We focus our attention on the cardinalities of its levels, $b_1, \dots, b_l$, that lead to a quick computation of the parameter c. Indeed, we recall that

$$c = \sum_{i=1}^{l} \binom{b_i}{2} + n - 1 - b_1 \tag{3.10}$$

from Theorem 3.5.8. It is clear from Eq. (3.10) that b_1 plays a special role in the computation of c, while $b_2, \dots, b_l$ are equally significant. From this observation, we choose an order on the cardinalities of the levels of a canonical tree, in order to reach a well-defined, canonical structure in view of the development of a reconstruction algorithm.
Let us consider two indices $2 \leq i < j \leq l$. Given a canonical tree T of height l and with co-degree sequence γ, if we exchange the nodes at level i with the nodes at level j, we get a tree T' that is still canonical, and with same co-degree sequence. This last property directly follows from Eq. (3.10).

As a consequence, we define a further operation on canonical trees, that consists of the following: if $b_1 = 1$, we do not change the tree. If $b_1 \geq 2$, we designate one of the children of the root, different from r_1, as the new root of the tree. Actually, this operation does not change the structure of the graph, since it consists in the renaming of the root only. On the other hand, from a practical point of view, it allows to get $b_1 = 1$ for any canonical tree. We also point out that the canonical tree thus obtained has height increased by one w.r.t. the previous one. Example 3.5.9 clarifies the described operation.
Finally, we re-order the levels of the tree, except from the first one, in decreasing order with respect to their cardinalities, that is, $b_2 \geq b_3 \geq \cdots \geq b_l$. Eq. (3.10) ensures that such a re-ordering does not affect the co-degree sequence γ.

Example 3.5.9. Let us consider the tree T in Fig. 3.13, whose co-degree sequence is $\gamma = (1^{21}, 0^{84})$. Fig. 3.14(a) shows a second tree, T', whose height is $l' = 7$ and with same co-degree sequence, obtained from T after the iterative application of the left-shift operation. It is canonical.
In Fig. 3.14(b), the operation that leads to $b_1 = 1$ is shown. Notice that T'' has height increased by one w.r.t. T', $l'' = 8$.
Finally, Fig. 3.14(c) represents the canonical tree T''' that is obtained from T'' after the re-ordering of its levels, with respect to the cardinalities.

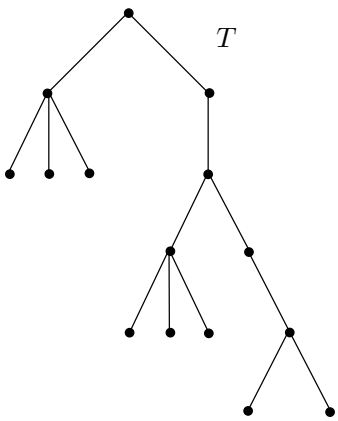


Figure 3.13: A tree on 15 vertices with co-degree sequence $\gamma = (1^{21}, 0^{84})$.

Property 3.5.10. Given a binary sequence $\gamma = (1^c, 0^{\binom{n}{2}-c})$, if there exists a tree having γ as its co-degree sequence, then there exists a canonical tree realizing γ whose levels satisfy:

i) $b_1 = 1$, and

ii) $b_2 \geq b_3 \geq \cdots \geq b_l$, with l the height of the tree.

The property is a direct consequence of the operations we described above.

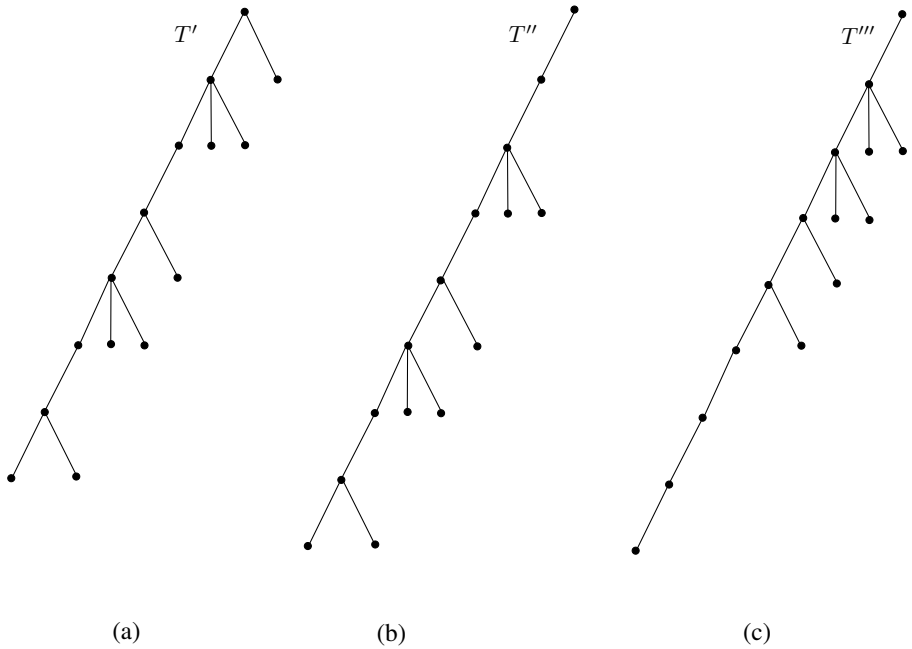


Figure 3.14: Three canonical trees on 15 vertices with co-degree sequence $\gamma = (1^{21}, 0^{84})$. In (b) and (c), the first level has cardinality equal to $b_1 = 1$. In (c), the levels are in decreasing order with respect to their cardinalities.

From Eq. (3.10) and Property 3.5.10, we conclude that the reconstruction problem consists in finding $b_2 \geq \cdots \geq b_l$ such that

$$c = \sum_{i=2}^{l} \binom{b_i}{2} + n - 2.$$

Notice that l can still vary, and more solutions may exist, in general.

3.5.3 Reconstruction of trees from their co-degree sequence

We formalize the reconstruction problem in its arithmetical formulation, highlighting all the required constraints.

Problem: given two integers n and c, with $n-1 \leq c \leq \binom{n-1}{2}$, find $l, b_2, \ldots, b_l \in \mathbb{N}$ such that

$$\begin{cases} c = \displaystyle\sum_{i=2}^{l} \binom{b_i}{2} + n - 2 \\ \displaystyle\sum_{i=2}^{l} b_i = n - 2 \\ 1 \leq l \leq n-1 \\ b_2 \geq b_3 \geq \cdots \geq b_l \geq 1. \end{cases} \tag{3.11}$$

As already mentioned, the solution is not unique in general, since canonical trees of different height l and with same co-degree sequence may exist. Since l can vary from 1 to $n-1$, we suppose to have its value as prior information. Indeed, in a practical implementation, we can run parallel lines of computation augmenting the computational cost of a factor of $O(n)$ only. So, in case of polynomial time implementation, the polinomiality of the whole procedure is not lost.
We manipulate the first equation in (3.11): replacing $\binom{b_i}{2} = \frac{b_i(b_i-1)}{2}$, for all $i = 2, \ldots, l$, we obtain

$$\sum_{i=2}^{l} b_i^2 - b_i = 2c - 2n - 4$$

and then, using that $\sum_{i=2}^{l} b_i = n - 2$,

$$\sum_{i=2}^{l} b_i^2 = 2c + 2 - n.$$

So, the reconstruction problem reaches the following final formulation:

Given n, c and l, find $b_2, \ldots, b_l \in \mathbb{N}$ such that

$$\begin{cases} \sum_{i=1}^{l-1} b_{i+1}^2 = 2c + 2 - n \\ \sum_{i=1}^{l-1} b_{i+1} = n - 2. \end{cases} \tag{3.12}$$

Starting from (3.12), we are now ready to define a reconstruction algorithm based on dynamic programming. To shorten the notation, we set $x := 2c + 2 - n$ and $y := n - 2$.

The algorithm *FindTree*

The equations given in (3.12) suggest the definition of an algorithm that uses dynamic programming, to avoid a brute-force generation of all the possible partitions of y into $l-1$ parts. Indeed, dynamic programming is a powerful tool that implicitly explores the (exponential) space of solutions, but without computing all of them, taking advantage of the recursive nature of the problem (a survey about algorithm design's techniques can be found in [54]). The key idea of this approach is to divide the starting problem in smaller sub-problems, and recursively solve them. The final solution is obtained by exploiting the solutions of the sub-problems previously solved (*top-down* approach).
We construct a *boolean table*, where we keep track of the existence of a solution for each sub-problem that we need to solve to get a solution of the input instance. After the computation of the table, we know whether a solution exists or not. In case of positive answer, such a solution can be deduced from the table itself.
More in detail, in our case the sub-problems we have to consider are the partial sums of the values b_i, that finally lead to the sums defined in Eq. (3.12). Indeed, we recursively

add a summand b_i until we find, if they exist, $l-1$ integers whose sum is y and whose squares sum to x.

We build a three-dimensional boolean table, Q, since we need to take into account three parameters: an index ℓ that counts how many summands b_i we have already considered, a variable χ that stores the partial sum of the squares of the summands $b_2^2 + \cdots + b_{\ell+1}^2$, and a variable υ that stores the partial sum $b_2 + \cdots + b_{\ell+1}$.
We define the entries of the boolean table as follows: $Q(\ell, \chi, \upsilon) = 1$ if and only if there exist $b_2, \ldots, b_{\ell+1}$ such that their sum is υ and the sum of their squares is χ. Namely, the entry $Q(\ell, \chi, \upsilon)$ points out whether the sub-problem

$$\begin{cases} \sum_{i=1}^{\ell-1} b_{i+1}^2 = \chi \\ \sum_{i=1}^{\ell-1} b_{i+1} = \upsilon \end{cases}$$

admits a solution or not.

Since we want to solve (3.12), we fill the boolean table until the entry $Q(l-1, x, y)$. Then, a solution to our instance exists if and only if $Q(l-1, x, y) = 1$.
The recursive relation used to fill the table is

$$Q(\ell, \chi, \upsilon) = 1 \quad \text{if and only if there exists} \quad 1 \leq t \leq \upsilon \text{ s.t. } Q(\ell-1, \chi - t^2, \upsilon - t) = 1, \tag{3.13}$$

starting with the base case

$$Q(1, \chi, \upsilon) = 1 \text{ if and only if } \upsilon^2 = \chi.$$

Indeed, the problem

$$\begin{cases} \sum_{i=1}^{\ell-1} b_{i+1}^2 = \chi \\ \sum_{i=1}^{\ell-1} b_{i+1} = \upsilon, \end{cases}$$

corresponding to $Q(\ell, \chi, \upsilon)$, admits a solution if and only if the sub-problem

$$\begin{cases} \sum_{i=1}^{\ell-2} b_{i+1}^2 = \chi - t^2 \\ \sum_{i=1}^{\ell-2} b_{i+1} = \upsilon - t, \end{cases}$$

corresponding to $Q(\ell-1, \chi - t^2, \upsilon - t)$, admits a solution too. In fact, to find a solution to the first instance, it is sufficient to consider the solution of the sub-problem and then add $b_{\ell-1} = t$ as the next summand.

Once the table is completed, it is possible to retrieve the values $b_2, \ldots, b_l$ that constitute the solution looking at the values t that satisfy Eq. (3.13), i.e., at the solutions of the sub-problems, that can be deduced from the non-null entries of Q. See Example 3.5.11 for a practical implementation.

The pseudo-code of the algorithm we described, *FindTree*, is given in Algorithm 10, together with the subroutine *QTable*, Algorithm 9, used for the construction of the boolean table Q. Example 3.5.11 describes the execution of the algorithm *FindTree* on input $l = 4$, $x = 17$ and $y = 7$.

Example 3.5.11. Let us consider the instance $T(9, 12)$, and fix the value $l = 4$ for the height of the solution. The formulation of the problem as in (3.12) is

$$\begin{cases} \sum_{i=1}^{3} b_{i+1}^2 = 17 \\ \sum_{i=1}^{3} b_{i+1} = 7, \end{cases}$$

and so we need to compute the boolean entry $Q(3, 17, 7)$ to establish if a solution exists. We start filling the first layer (base case) of the table Q, corresponding to the sub-problems of type

$$\begin{cases} \sum_{i=1}^{1} b_{i+1}^2 = \chi \\ \sum_{i=1}^{1} b_{i+1} = \upsilon. \end{cases}$$

We get the matrix

$$Q(1, \chi, \upsilon) = \begin{pmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Since the sub-problems corresponding to the entries of the first layer of Q require a solution having only one summand, b_2, the only entries different from zero correspond to the case $\upsilon^2 = \chi$, so to $Q(1, 1, 1)$, $Q(1, 4, 2)$, $Q(1, 9, 3)$ and $Q(1, 16, 4)$.

We now proceed with the second layer, corresponding to the solutions of the sub-problems of type

$$\begin{cases} \sum_{i=1}^{2} b_{i+1}^2 = \chi \\ \sum_{i=1}^{2} b_{i+1} = \upsilon, \end{cases}$$

and such that b_2 corresponds to a solution of $Q(1, i, j)$ for some indices i, j.
We get the boolean matrix

$$Q(2, \chi, \upsilon) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{pmatrix}$$

As an example, $Q(2, 8, 4) = 1$, since there exists a value t such that $Q(1, 8-t^2, 4-t) = 1$. Such a value is $t = 2$.

We proceed with the completion of the last layer of the boolean table, that turns out to be

$$Q(3, \chi, \upsilon) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \end{pmatrix}$$

Since $Q(3,17,7) = 1$, we conclude that a solution exists, i.e., a canonical tree on $n = 9$ vertices whose height is $l = 4$, such that $b_1 = 1$, and having co-degree sequence equal to $\gamma = (1^{12}, 0^{24})$.

We now compute the values b_4, b_3 and b_2 that constitute our solution. Starting from the third layer, we consider its non-null entries and the corresponding values t:

$Q(3,3,3)$,	$t = 3$	admissible if and only if	$Q(2,8,4) = 1$	true
$Q(3,6,4)$,	$t = 4$	admissible if and only if	$Q(2,1,3) = 1$	false
$Q(3,9,5)$,	$t = 5$	admissible if and only if	$Q(2,-8,2) = 1$	false
$Q(3,11,5)$,	$t = 5$	admissible if and only if	$Q(2,-8,2) = 1$	false
$Q(3,12,6)$,	$t = 6$	admissible if and only if	$Q(2,-19,1) = 1$	false
$Q(3,14,6)$,	$t = 6$	admissible if and only if	$Q(2,-19,1) = 1$	false
$Q(3,17,7)$,	$t = 7$	admissible if and only if	$Q(2,-32,0) = 1$	false

Then, we have a unique choice for b_4, corresponding to $t = 3$. We now move to the second layer of Q and consider the sub-problem we got after the choice $b_4 = 3$, corresponding to the entry $Q(2,8,4)$.
We have that

$t = 1$	is admissible if and only if	$Q(1,7,3) = 1$	false
$t = 2$	is admissible if and only if	$Q(1,4,2) = 1$	true
$t = 3$	is admissible if and only if	$Q(1,-1,1) = 1$	false
$t = 4$	is admissible if and only if	$Q(1,-8,0) = 1$	false

So, we conclude that $b_3 = 2$.
Finally, we move to the first layer and deduce $b_2 = 2$ from the entry $Q(1,4,2) = 1$.

After a re-ordering of the levels according to their cardinalities, we get the tree in Fig. 3.15 as a solution of the instance $T(9,12)$.

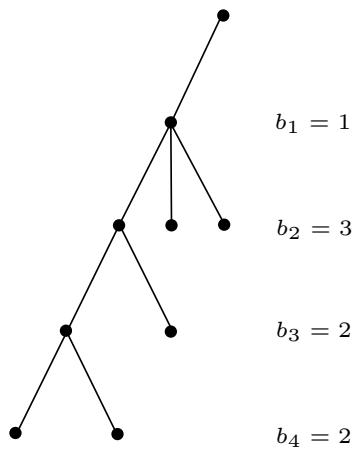


Figure 3.15: The canonical tree of height $l = 4$ that is a solution of $T(9,12)$.

Algorithm 9. *QTable*(l, x, y)

Input: l, x, y positive integers, with $x, y \leq l - 1$

```
Initialize Q an (l − 1) × x × y null matrix;
for i = 1, ..., x do
    for j = 1, ..., y do
        if i = j^2 then
            Q(1, i, j) = 1;
        end
    end
end
for ℓ = 2, ..., l − 1 do
    for i = 1, ..., x do
        for j = 1, ..., y do
            for t = 1, ..., j do
                if Q(ℓ − 1, i − t^2, j − t) = 1 then
                    Q(ℓ, i, j) = 1;
                end
            end
        end
    end
end
```

Output: Q

Algorithm 10. *FindTree*(Q)

Input: Q an output of *QTable(l,x,y)* such that $Q(l-1, x, y) = 1$

```
1  Initialize b an array of length l;
2  ℓ = l − 1;
3  while ℓ ≥ 1 do
4      for t = 1, ..., y do
5          if Q(ℓ − 1, x − t^2, y − t) = 1 then
6              b(ℓ + 1) = t;
7              x = x − t^2;
8              y = y − t;
9              ℓ = ℓ − 1;
10             go to line 3;
11         end
12     end
13 end
14 Rearrange the entries of b in decreasing order;
```

Output: b

Lemma 3.5.12. The computational cost of the algorithms *QTable* and *FindTree* is $O(n^5)$ and $O(n^2)$, respectively, with n the number of vertices of the tree we want to reconstruct.

Proof. From Algorithm 9, we immediately obtain the computational cost of *QTable*, that is given by the four nested *for* loops, running on the indices ℓ, i, j and t from 1 to $l-1, x, y$

and j, respectively. Since l and y are of the order of n, while x is of the order of n^2, the total computational cost results in $O(n^5)$.
On the other hand, from Algorithm 10 we observe that the computational cost of *FindTree* is given by the nested *while* and *for* loops, both running on indices of size of order of n. It follows that the algorithm runs in $O(n^2)$ time. □

We conclude that a solution to an instance $T(n, c)$ can always be found in polynomial time.

Theorem 3.5.13. Given two positive integers n and c, with $n - 2 \leq c \leq \binom{n-1}{2}$, it is possible to find a solution to the instance $T(n, c)$ in polynomial time, if such a solution exists.

Proof. From Property 3.5.10, we can look for canonical trees realizing the co-degree sequence $\gamma = (1^c, 0^{\binom{n}{2}-c})$ such that the cardinality of the first level is $b_1 = 1$. The algorithms *FindTree* and *QTable* can be used to compute such a tree, up to the knowledge of its height l. This can be done with a total computational cost of $O(n^5)$, mainly due to the construction of the table Q, see Lemma 3.5.12. Being l of the order of n, we can start parallel computations, one for each possible choice of the parameter, thus resulting in a total running time of $O(n^6)$, that is still polynomial in the size of the input. □

3.5.4 Generalization and open problems

We showed that given two positive integers n and $n - 2 \leq c \leq \binom{n-1}{2}$, it is possible to construct in polynomial time a tree having $\gamma = (1^c, 0^{\binom{n}{2}-c})$ as its co-degree sequence, if such a tree exists. However, the class of trees does not include all the simple graphs having a binary co-degree sequence. Indeed, the algorithm *FindTree* fails on some input, even if there exists a simple graph having the required co-degree sequence. Here we exhibit an example of such a case. See also Table 3.1 for a complete overview concerning the existence of a solution in the first cases, for $2 \leq n \leq 10$.

Example 3.5.14. Choosing $n = 15$ and $c = 46$, *FindTree* ends without providing any solution, meaning that there exists no tree on 15 vertices in which exactly 46 pairs of them have one common neighbor. On the other hand, there exists (at least) a simple graph realizing the binary co-degree sequence $\gamma = (1^{46}, 0^{59})$, see Figure 3.16.

The graph in Fig. 3.16 can be seen as a slightly adaptation of a tree, where a further edge is added to form a cycle of length three. This observation, together with the result given in Theorem 3.5.1, suggests to deeper investigate graphs made of triangles. As a preliminary result, we prove that

Theorem 3.5.15. Let G be a graph on n vertices having co-degree sequence γ. Then, γ is binary if and only if G is C_4-free, meaning that no cycle of length four occurs.

Proof. Let us suppose that a cycle of length four occurs in G, $\{v_1, v_2, v_3, v_4, v_1\}$. Then, by definition of cycle, the edges $\{v_1, v_2\}$,$\{v_2, v_3\}$,$\{v_3, v_4\}$ and $\{v_1, v_4\}$ belong to the graph. As a consequence, v_1 and v_3 have (at least) two common neighbors, as well as v_2 and v_4. It follows that γ is not binary, in contradiction with our hypothesis.

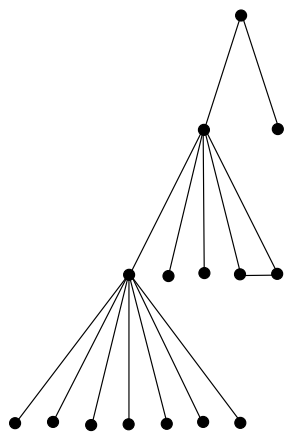


Figure 3.16: A simple graph not in Υ_{15} and having binary co-degree sequence, $\gamma = (1^{46}, 0^{59})$.

Number of vertices n	**Admitted values for c**	**Instances with no solution in Υ_n**
2	$0 \leq c \leq 0$	$c = 0$
3	$1 \leq c \leq 1$	$--$
4	$2 \leq c \leq 3$	$--$
5	$3 \leq c \leq 6$	$c = 5$
6	$4 \leq c \leq 10$	$c = 8, 9$
7	$5 \leq c \leq 15$	$c = 10, 12, 13, 14$
8	$6 \leq c \leq 21$	$c = 11, 14, 15, 17, 18, 19, 20$
9	$7 \leq c \leq 28$	$c = 15, 19, 20, 21, 23, 24, 25, 26, 27$
10	$8 \leq c \leq 36$	$c = 22, 25, 26, 27, 28, 30, 31, 32, 33, 34, 35$

Table 3.1: The table shows the first cases such that the co-degree sequence $\gamma = (1^c, 0^{\binom{n}{2}-c})$ cannot be realized by a tree, for $2 \leq n \leq 10$ and $n - 2 \leq c \leq \binom{n-1}{2}$.

On the other hand, let us suppose that γ is not binary. Then, there exist two vertices, say v_1 and v_2, having at least two common neighbors, say u_1 and u_2. By definition, the edges $\{v_1, u_1\}$,$\{v_1, u_2\}$,$\{v_2, u_1\}$ and $\{v_2, u_2\}$ are in G and form a cycle of length four, so that G is not C_4-free, as claimed. $\square$

From Theorem 3.5.15, we deduce that the general reconstruction problem for binary co-degree sequences consists of looking for C_4-free graphs. Actually, it is possible to reformulate it in terms of degree sequences.
Let us consider a C_4-free graph on n vertices with co-degree sequence $\gamma = (1^c, 0^{\binom{n}{2}-c})$. If we denote its degree sequence as $\pi = (d_1, \dots, d_n)$, we can express the number of pairs of vertices sharing one neighbor as

$$c = \sum_{i=1}^{n} \binom{d_i}{2}.$$

Then, the general version of the reconstruction problem for binary co-degree sequences becomes

Problem: given two positive integers n and $c \in [0, \binom{n}{2}]$, find $d_1 \geq \cdots \geq d_n$ positive integers such that $c = \sum_{i=1}^{n} \binom{d_i}{2}$ and $\pi = (d_1, \dots, d_n)$ admits a C_4-free realization.

As we did in case of trees, we can reformulate the problem in terms of two sums, that is,

$$\begin{cases} \sum_{i=1}^{n} d_i = 2m \\ \sum_{i=1}^{n} d_i^2 = 2(c+m), \end{cases} \tag{3.14}$$

where m is the number of edges of the graph, that we can suppose to have as prior information.
The similarities between Eq. (3.14) and Eq. (3.12) are evident, and may inspire an algorithm based on dynamic programming to solve the general instance of the reconstruction problem, as done in case of trees. Actually, we have to be more careful in the general case. Indeed, the values d_i have to be a solution of (3.14), but also the degrees of some graph, meaning that the inequalities of the Erdős-Gallai condition have to be satisfied (see Theorem 3.1.3). These constraints have to be considered in the development of a reconstruction algorithm, that we did not define yet.

Conclusion and final remarks

As a conclusion of the chapter, we propose a short list of possible research lines that arise from our work.

- ▷ Section 3.2: A class of sequences, dual to $\mathcal{P}$, can be defined by fixing a constant positive pattern $\mathbf{p}^+$ and then allowing the negative part $\mathbf{p}^-$ to vary. By duality, for

the higher degrees we expect a behavior similar to that one of tails. Combining all the results could be the starting point for the reconstruction of the whole class $\mathcal{D}$.

Problem 1. Deepen and generalize the concept of pattern as the starting point for the solution of the reconstruction problem in $\mathcal{D}$.

- ▷ Section 3.3: We provided a complete characterization of the degree sequences associated to the ideals of the poset $\mathcal{T}_n$ having one or two generators, as well as a deep analysis concerning their uniqueness. The study of the general case of sequences in the class $\mathcal{D}^{ext}$ is still missing.

  Problem 2. Analyze the degree sequences realized by ideals with three or more generators, and characterize the sequences of $\mathcal{D}^{ext} \setminus \mathcal{D}$ for which the uniqueness property still holds. Finally, define a reconstruction strategy that takes advantage from the structure of the hypergraphs associated to down-sets.

- ▷ Section 3.4: The randomized algorithm we described improves previous results in the literature. Indeed, we have shown that an integer sequence such that $d_1 = o(\sigma^{1-2/k})$ is k-graphic, against the condition $d_1 = o(\min\{\sigma^{1/2}, \sigma^{1-2/k}\})$ presented in [37]. On the other hand, the authors in [16] proved that each integer sequence such that $d_1 = o(\sigma^{1-1/k})$ is k-graphic. Even if their condition is weaker than ours, they did not provide any reconstruction strategy, while we also defined an algorithm that finds a realization of the given sequence with positive probability and in polynomial time. Finally, we also described a variance of the strategy that allows to ignore higher degrees.

  Problem 3. Enhance the bound for the characterization of k-graphic sequences, as well as new strategies for their reconstruction.

- ▷ Section 3.5: Given a binary co-degree sequence, we are able to solve the reconstruction problem if a tree realizing it exists. However, we have shown that not all of them can be realized by trees, since the general solution lies in the class of C_4-free graphs.

  Problem 4. Characterize the degree sequences associated to C_4-free graphs, and then develop a reconstruction strategy to find, if it exists, a realization for a given binary co-degree sequence.

Chapter 4

Homogeneous configurations in the discrete plane

In this chapter, we consider binary configurations in $\mathbb{Z}^2$ and we inspect them through windows of exact polyomino shapes. This study introduces a generalization to the standard notion of projection, since data are collected by means of bi-dimensional objects instead of the usual linear X-rays. In particular, we focus our research on h-homogeneous configurations, i.e., those ones such that for any possible position of a given polyomino W on $\mathbb{Z}^2$, the value of the collected projection is always constant and equal to h.
After a classification of exact polyominoes, we provide some results about their periodicity properties, and then focus our attention on a conjecture proposed by Frosini and Nivat in 2005 [42]. The authors start from a decomposition theorem that holds for rectangles and stating that, given a configuration of the plane that is h-homogeneous w.r.t. a given rectangle W, it is always possible to decompose it in 1-homogeneous disjoint ones [56].
The conjecture they propose claims that the same result can be extended to the whole class of exact polyominoes. As an example, it holds for diamonds and all those exact polyominoes that are open sets in $\mathbb{Z}^2$ [14].
Our main result is the solution of such a conjecture: we show that it is false in general, providing different counterexamples starting from an exhaustive analysis of tiles and homogeneous configurations. At the end of the chapter, we make some observations that lead us to propose a further conjecture in the same research line, stating that the decomposition theorem holds in case the area of the considered polyomino is a prime number. We also present some partial results supporting our claim, in particular, we define some classes of tiles and h-homogeneous configurations that cannot be decomposed, due to the fact that h is a divisor of the area of the involved polyomino.
Part of the results described in this chapter are joint work with A. Frosini in [4].

4.1 Introduction and preliminaries

In this section we give the motivations of our work together with the notations and the mathematical tools we are going to use.

Polyominoes and boundary words

A polyomino is defined as a finite subset of $\mathbb{Z}^2$. It can be represented as a set of 4-connected cells on a squared surface, each cell representing a point of $\mathbb{Z}^2$. We consider polyominoes that have no holes, in other words, whose boundary (when considered as set of cells) is a single, closed and non-intersecting path made of unit steps in $\mathbb{Z}^2$. Since we are in the square lattice, there are only four allowed directions for unit steps, corresponding to the directions North, South, East, and West. We consider each polyomino up to translation.

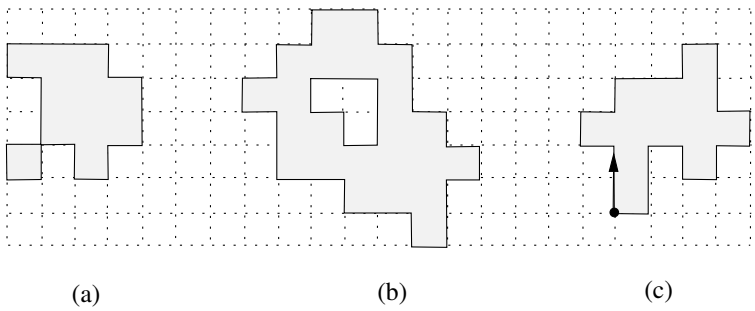


Figure 4.1: Three sets of cells in $\mathbb{Z}^2$. According to the definition, only (c) is a polyomino. Indeed, (a) in not 4-connected, while in (b) a hole is present.

A polyomino is uniquely identified by the boundary of its square lattice representation, and can be coded using a word on a four letters alphabet, according to the Freeman chain code ([40, 41]). Each letter of the alphabet corresponds to a unit step made on the path in one of the possible directions. We choose the alphabet $\Sigma = \{1, 0, \overline{1}, \overline{0}\}$, where the letters correspond, in order, to the directions $\{\uparrow, \rightarrow, \downarrow, \leftarrow\}$. Letters corresponding to opposite directions, namely 1 and $\overline{1}$, 0 and $\overline{0}$, are called *complement*. The coding of the boundary is obtained as follows: choosing a starting point and an orientation, i.e., clockwise or counterclockwise, the boundary is traveled step-by-step taking note of the letter of Σ that codes the corresponding unit step. As an example, starting from the lower-left point and choosing the clockwise orientation, the polyomino in Fig. 4.1(c) is coded by the word $P = 11\overline{0}1010010\overline{1}\overline{1}0\overline{1}0\overline{1}0\overline{1}0\overline{1}0\overline{1}10$.
As a matter of fact, this coding is not unique, since we can arbitrarily choose both the orientation and the starting point of the boundary, and all the codings of a polyomino form an equivalence class. In our work, we fix the clockwise orientation. Then, the unit square turns out to be $U = 10\overline{1}\overline{0}$, for example. From now on we will use a capital letter to point out both the polyomino, interpreted as a set of cells, and the word that codes its boundary.
As a standard notation, we write Σ^* for the set of all words defined on the alphabet Σ, with ε the empty word, and $\Sigma^+ = \Sigma^* \setminus \varepsilon$. Given a word $w \in \Sigma^*$, a letter $\alpha \in \Sigma$ and a positive integer k, we write $|w|$ for the length of w, that, in case of a boundary word, is also the *perimeter* of the polyomino, $|w|_\alpha$ for the number of occurrences of the letter α in w, and w^k for the concatenation of k copies of w, $w^k = ww \dots w$. By convention, $w^0 = \varepsilon$.

A word $v \in \Sigma^*$ is a *factor* of w if there exist $x, y \in \Sigma^*$ such that $w = xvy$. If $x = \varepsilon$, v is a *prefix* of w, while if $y = \varepsilon$ it is a *suffix*. The factor v is *proper* if $v \neq \varepsilon, w$. Finally, the word w is said to be *periodic* if there exist a factor v and an integer $k \geq 2$ such that $w = v^k$.

Property 4.1.1. If $P \in \Sigma^+$ is the boundary word of a polyomino, then $|P|_\alpha = |P|_{\overline{\alpha}}$ for all $\alpha \in \Sigma$ (the boundary is a closed path), and, for any Q proper factor of P, $|Q|_\alpha \neq |Q|_{\overline{\alpha}}$ for all $\alpha \in \Sigma$ (the boundary does not self-intersect).

Given $v, w \in \Sigma^*$, they are said to be *conjugate*, say $v \equiv w$, if there exist $x, y \in \Sigma^*$ such that $v = xy$ and $w = yx$. In other words, w can be obtained as a cyclic shift of v. Sometimes we will use the notation $v \equiv_d w$ to underline the length d of the cyclic shift that from w leads to v. From the point of view of polyominoes, the conjugacy class of a boundary word, once an orientation is fixed, contains all its cyclic shifts, each of them corresponding to a different choice for the starting point of the coding. As an example, the conjugacy class corresponding to the unit square is $U = 10\overline{1}\overline{0} \equiv 0\overline{1}\overline{0}1 \equiv \overline{1}\overline{0}10 \equiv \overline{0}10\overline{1}$.

We finally define three operators on a word $w = w_1 \dots w_n \in \Sigma^*$:

i) the *complement* of w is the word obtained by replacing each letter of w with its complement in the alphabet Σ, $\overline{w} = \overline{w}_1\overline{w}_2 \dots \overline{w}_n$,

ii) the *reversal* of w is the word $\tilde{w} = w_n w_{n-1} \dots w_1$. If $\tilde{w} = w$, then w is called a *palindrome*,

iii) the *hat* of w is the antimorphic involution given by the composition of the previous operations, $\widehat{w} = \tilde{\overline{w}}$.

Example 4.1.2. The path represented in Fig. 4.2 is coded by the word $w = 101100\overline{1}010110\overline{1}00$. Applying the described operators, we obtain

i) $\overline{w} = \overline{101100}1\overline{010110}1\overline{00}$,

ii) $\tilde{w} = 00\overline{1}011010\overline{1}001101$ (the word is not a palindrome),

iii) $\widehat{w} = \overline{00}1\overline{011010}1\overline{001101}$.

From the point of view of paths, the word $\widehat{w}$ corresponds to the path coded by w traveled in the opposite direction, see Fig. 4.2.

Exact polyominoes

We are interested in a special class of polyominoes, i.e., those ones that tile the discrete plane by translation, also called *exact polyominoes* or *tiles*. Given a polyomino P, a *tiling of* $\mathbb{Z}^2$ *by* P is defined as a set of non-overlapping copies of P that cover the whole discrete plane. The following trivial result holds:

Theorem 4.1.3. A polyomino is a tile if and only if it is possible to completely surround it with copies of itself.

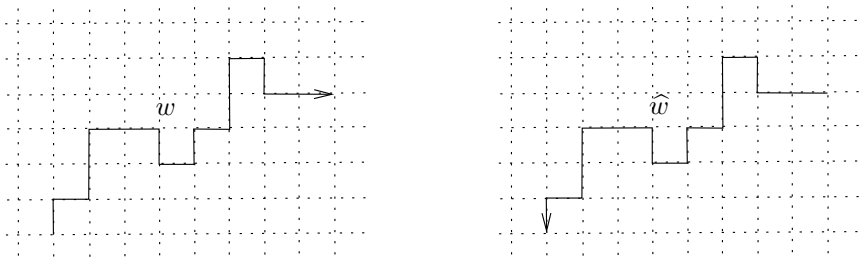


Figure 4.2: On the left, the path coded by the word $w = 101100\overline{1}010110\overline{1}00$. On the right, the same path traveled in the opposite direction, that is coded by the word $\widehat{w} = \overline{00}1\overline{011010}1\overline{001101}$.

Remark 4.1.4. Even though $\mathbb{Z}^2$ is infinite, to check if a polyomino is exact it is sufficient to consider only one copy of it and look for possible surroundings. Indeed, if it is possible to surround one tile completely with copies of itself, then the same procedure can be iterated to cover the whole discrete plane.

Example 4.1.5. The polyomino $P = 1\overline{00}10110100\overline{1}0\overline{1011010}$ is exact, since it is possible to surround it with four copies of itself, see Figure 4.3.

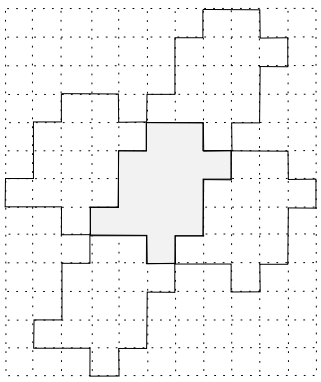


Figure 4.3: An exact polyomino surrounded with four copies of itself.

Exact polyominoes have been characterized by Beauquier and Nivat in 1991. Thanks to their result, it is possible to establish if a polyomino tiles the plane by translation looking at its boundary word only.

Theorem 4.1.6 (Beauquier and Nivat [15])**.** A polyomino P is exact if and only if there exist $A, B, C \in \Sigma^*$ such that

$$P = ABC\widehat{A}\widehat{B}\widehat{C},$$

where at most one among A, B and C can be the empty word. Moreover, this factorization may not be unique, and there could be different factorizations for the same tile.

By the authors' name, we address the factorization described in Theorem 4.1.6 as *BN-factorization*, and we call A, B and C the *BN-factors* of P. As already mentioned, the factors A and $\widehat{A}$ code the same path in $\mathbb{Z}^2$, but traveled in opposite directions. Starting

from this observation, we give a geometrical interpretation to a BN-factorization: the factors A,B, C and $\widehat{A}$,$\widehat{B}$, $\widehat{C}$, respectively, define the translations of the polyomino P in $\mathbb{Z}^2$ that can be used to surround it with copies of itself, see Figure 4.4. In other words, given an exact polyomino $P = ABC\widehat{A}\widehat{B}\widehat{C}$, a surrounding can be found by matching, in neighbor tiles, the sides of the boundary corresponding to A and $\widehat{A}$ (B and $\widehat{B}$, C and $\widehat{C}$, respectively).

Remark 4.1.7. Different BN-factorizations define different translations, associated to different ways in which the exact polyomino P can be surrounded with copies of itself. Consequently, different BN-factorizations induce different tilings of the plane.

A polyomino having all the BN-factors different from the empty word is called a *pseudo-hexagon*, while in case $C = \varepsilon$ it is called a *pseudo-square*. Notice that such a definition depends on the BN-factorization we consider, that we remind is not unique in general. Indeed, the same tile can be both a pseudo-square and a pseudo-hexagon, as described in Example 4.1.8. We refer to Section 4.2 for more details concerning the classification of exact polyominoes.

Example 4.1.8. The polyomino $P = 11\overline{0}1101010\overline{1}\overline{1}0\overline{1}\overline{1}\overline{0}1\overline{0}10$ admits the following BN-factorizations,

$$P_{(a)} = (11\overline{0}11)(01010)(\overline{1}\overline{1}0\overline{1}\overline{1})(\overline{0}1\overline{0}1\overline{0}),$$
$$P_{(b)} = (10101)(0\overline{1}\overline{1}0)(\overline{1})(\overline{1}0\overline{1}0\overline{1})(\overline{0}11\overline{0})(1),$$

with a different number of BN-factors. The induced tilings are depicted in Fig. 4.4.

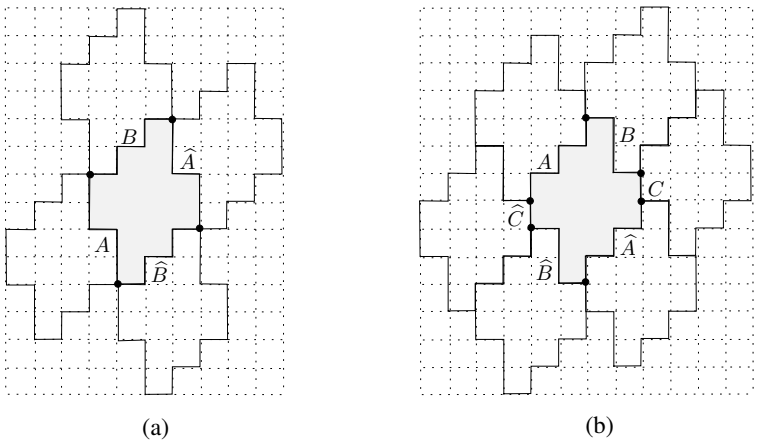


Figure 4.4: An exact polyomino admitting two BN-factorizations, as a pseudo-square and as a pseudo-hexagon. The induced tilings are represented in (a) and (b), respectively. Black dots highlight the BN-factorization that defines the translations inducing the corresponding tiling.

As a matter of fact, the number of non-empty BN-factors describes the number of copies of P used for its surrounding in the induced tiling.

We observe that tilings are *periodic* by construction, meaning that there exist one or two linear independent vectors in $\mathbb{Z}^2$ such that the tiling is invariant by translation along

them. In the first case, the tiling is *semi-regular*, while in the second case it is *regular* [15]. Both the tilings in Fig. 4.4 are regular, while Fig. 4.5 shows an example of semi-regular tiling. Such directions of periodicity can be deduced from the BN-factorization that induces the tiling itself.

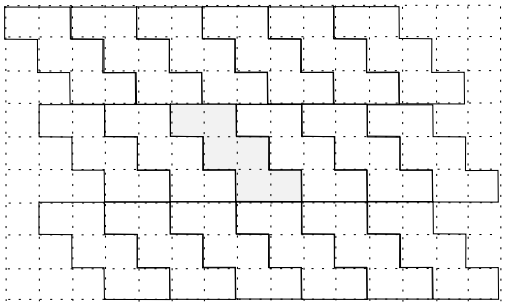


Figure 4.5: An example of semi-regular tiling, where only one direction of periodicity, $v = (2, 0)$, is present.

We postpone to Section 4.2 the analysis and classification of exact polyominoes with respect to the number and type of admitted BN-factorization(s), as well as the description of the directions of periodicity of the tilings they induce.

Homogeneous configurations and decomposability

We introduce here the notion of *configuration of the plane* and the conjecture that is the subject of our work.
A *configuration of the plane* A is a subset of points of $\mathbb{Z}^2$. We describe the set A through an infinite binary matrix, using the entry 1 when the corresponding point belongs to A, and 0 otherwise. To simplify the notation, we will omit the entries equal to 0. A configuration A is said to be *periodic with respect to* v if, for any $a \in A$, the point $a \pm v \in A$, where the notation $a \pm v$ stands for the translation of the point a along the discrete direction v. Since we are in $\mathbb{Z}^2$, at most two directions of periodicity exist, up to a linear combination of linearly independent vectors.
Once a configuration A is given, it is possible to inspect it using a polyomino, technique introduced in [42] as a generalization of the concept of linear projection. The designated polyomino W, also called *window*, is used as a probe to scan the discrete plane: for each possible position in $\mathbb{Z}^2$, we count the number of points of A that lie inside the window, also called *scan*. This number is always finite, and can span from 0, if all the points of the discrete plane in the window are equal to 0, to ω, with ω the area of the polyomino W, if all the points inside W are equal to 1. We are interested in special configurations, where the number of 1s in the scan is a constant.
Given a configuration A and a polyomino W, the configuration is said to be *h-homogeneous w.r.t. W* if, for any possible position of W in $\mathbb{Z}^2$, there are always h elements of A in the scan of W. We underline that the property of being homogeneous strictly depends on W: the same configuration can be homogeneous with respect to a polyomino but not homogeneous with respect to a different one.

Nivat achieved interesting results in case the chosen window is an exact polyomino. In particular,

Theorem 4.1.9 (Nivat [56])**.** Given a polyomino W, there exists a 1-homogeneous configuration w.r.t. W if and only if the polyomino is exact. Moreover, such a configuration is periodic w.r.t. the directions of periodicity of W.

Example 4.1.10. As an example, the configuration depicted in Fig. 4.6(a) is 1-homogeneous w.r.t. the tile $W = 1010\overline{1}01010\overline{1}010\overline{1}0\overline{1010}1\overline{01010}1\overline{010}1\overline{0}$. It is periodic with respect to the vectors $v_1 = (4,3)$ and $v_2 = (-4,1)$. Indeed, there exists a tiling induced by W having v_1 and v_2 as directions of periodicity.

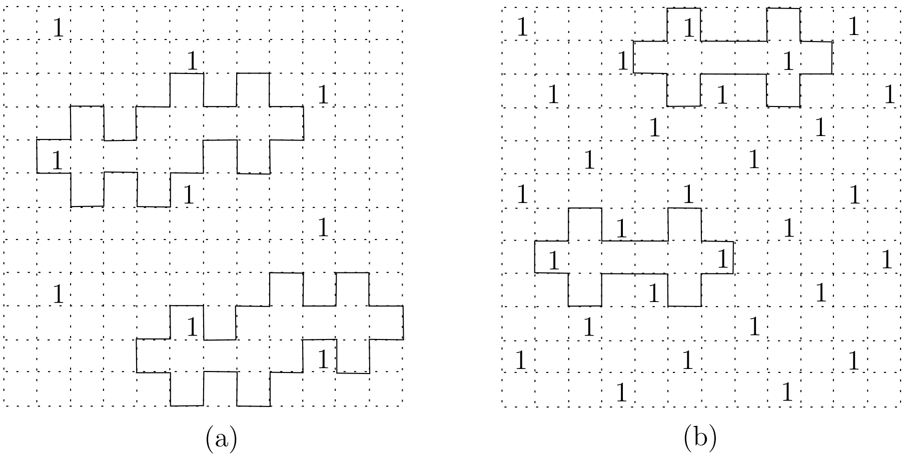


Figure 4.6: (a) A configuration of the plane that is 1-homogeneous w.r.t. an exact polyomino. (b) A 2-homogeneous configuration with respect to a polyomino that is not exact.

On the other hand, if we increase the homogeneity value to $h \geq 2$, homogeneous configurations can be found even in case of polyominoes that are not exact, as shown in Fig. 4.6(b). We remark that this is not in contradiction with Theorem 4.1.9, being $h \neq 1$. A further result has been obtained in case of rectangles:

Theorem 4.1.11 (Nivat [56])**.** Let us consider A a configuration of the plane that is h-homogeneous w.r.t. a rectangle W, with $2 \leq h \leq \omega$. Then, there exist h disjoint configurations, $A_1, \ldots, A_h$, such that A_i is 1-homogeneous w.r.t. W, for all $1 \leq i \leq h$, and $A = \bigcup_{i=1}^{h} A_i$.

Theorem 4.1.11 is in fact a decomposition theorem for those configurations that are homogeneous w.r.t. rectangles, where periodicity plays a special role (see the proof in [56]). Since exact polyominoes show similar periodic behaviors, specially in the tilings they induce, the authors in [42] conjectured that the decomposition property could be extended to all exact polyominoes. We are interested in the study of such a conjecture.

Conjecture (Frosini and Nivat [42]): given an exact polyomino W and a configuration A that is h-homogeneous w.r.t. W, for some $1 \leq h \leq \omega$, it is always possible to decompose A in h disjoint configurations that are still 1-homogeneous w.r.t. W.

The decomposition property has been proven in [14] for diamonds and polyominoes that are open sets in $\mathbb{Z}^2$, but we finally showed in [4] that it does not hold in general. As a consequence, the detection of further classes of tiles for which the decomposition property holds is the new focus of our research. We present our results in the next sections of this chapter, while we conclude this section introducing some technical definitions and results.

Definition 4.1.12. Given an exact polyomino W of area ω, and an integer $1 \leq h \leq \omega$, W is said to be *h-decomposable* if any configuration A that is h-homogeneous w.r.t. W can be decomposed in h disjoint 1-homogeneous configurations.

Remark 4.1.13. Notice that the decomposition property holds for a polyomino W if and only if W is h-decomposable for all $2 \leq h \leq \omega$.

Theorem 4.1.14. Let W be an exact polyomino of area ω that is h-decomposable for some $2 \leq h \leq \lfloor \frac{\omega}{2} \rfloor$. Then, W is $(\omega - h)$-decomposable.

Proof. Let A be a $(\omega - h)$-homogeneous configuration w.r.t. W, and let us denote $\overline{A}$ the configuration obtained by exchanging 0s with 1s in A. The new configuration is clearly h-homogeneous w.r.t. W and so, by hypothesis, we can decompose it in h (periodic) 1-homogeneous configurations, $\overline{A}_1, \ldots, \overline{A}_h$.
Let us now consider $\mathcal{C}_1$ the set of all the possible ω configurations of the plane that are 1-homogeneous w.r.t. W, and let us denote $\{A_1, \ldots, A_{\omega-h}\} = \mathcal{C}_1 \setminus \{\overline{A}_1, \ldots, \overline{A}_h\}$. Since $A_i \cap A_j = \emptyset$ for all $1 \leq i \neq j \leq \omega - h$, the configuration $A = \bigcup_{i=1}^{\omega-h} A_i$ is such that A is $(\omega - h)$-homogeneous w.r.t. W. Moreover, the set $\{A_i\}_{1 \leq i \leq \omega-h}$ defines a decomposition of A in disjoint 1-homogeneous configurations, i.e., W is $(\omega - h)$-decomposable. $\square$

As a consequence, from now on we will consider the homogeneity values $h = 2, \ldots, \lfloor \frac{\omega}{2} \rfloor$ only.

4.2 Classification of exact polyominoes

The decomposition property was conjectured to be valid for all exact polyominoes. For this reason, we dedicate this section to their analysis and classification, in order to provide useful properties to inspect the configurations of the plane. We deserve particular interest to the property of periodicity.

4.2.1 Pseudo-squares and pseudo-hexagons

We start from the definition of BN-factorization given in Theorem 4.1.6, so we consider the boundary word of an exact polyomino as

$$P = ABC\widehat{A}\widehat{B}\widehat{C},$$

where we remind that at most one factor, say C, may be empty.
The BN-factor C plays a role in a preliminary classification of tiles. Indeed, if $C = \varepsilon$, the polyomino P is addressed as a *pseudo-square*, since only four BN-factors are present in its factorization. The name is due to the fact that such a factorization defines a (regular)

tiling in which each polyomino is surrounded with exactly four copies of itself. On the other hand, if $C \neq \varepsilon$, then P is called a *pseudo-hexagon*, again from the number of its BN-factors. In this case, we have to be careful in the analysis of the induced tiling: indeed, each polyomino is surrounded with exactly six copies of itself if the tiling is regular, see Fig. 4.4(b), but this number can sometimes reduce to five or four if the tiling is semi-regular, as in Fig. 4.5.
As underlined by Beauquier and Nivat, the factorization that defines an exact polyomino is not unique, in general. Indeed, generally speaking, the boundary word of a polyomino may admit several different BN-factorizations, each of them inducing a different regular tiling, as shown both in Fig. 4.4 and in the following example.

Example 4.2.1. If we consider a rectangle, the tile admits one BN-factorization as a pseudo-square, but many different ones as a pseudo-hexagon. As an example,

$$\begin{aligned} P_{(a)} &= (11)(000)(\overline{11})(\overline{000}) \\ P_{(b)} &= (1)(1)(000)(\overline{1})(\overline{1})(\overline{000}) \\ P_{(c)} &= (11)(0)(00)(\overline{11})(\overline{0})(\overline{00}) \\ P_{(d)} &= (11)(00)(0)(\overline{11})(\overline{00})(\overline{0}) \end{aligned}$$

are all the possible BN-factorizations of the rectangle of size 3×2. See Figure 4.7 for the induced regular tilings.

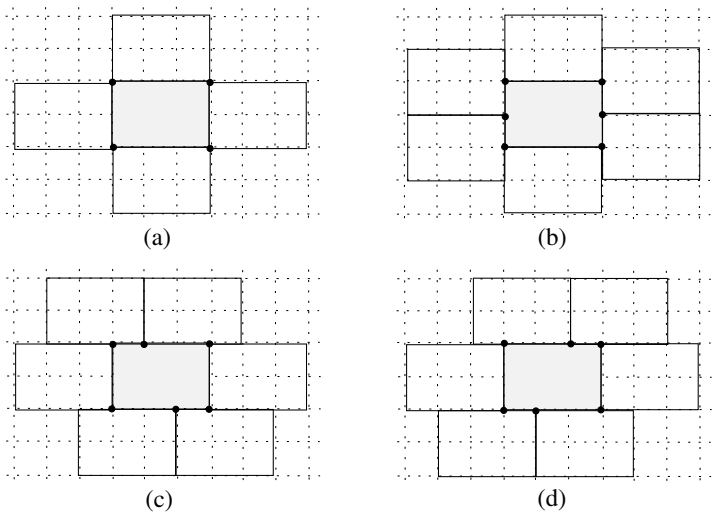


Figure 4.7: All the possible regular tilings induced by the rectangle of size 3×2.

Since the two classes of pseudo-squares and pseudo-hexagons are not disjoint, we introduce the definition of *perfect pseudo-square*, that refers to those tiles that do not admit any factorization as a pseudo-hexagon. An example of perfect pseudo-square is the tile in Fig. 4.3.
A necessary condition for a polyomino to be a perfect pseudo-square is the following:

Proposition 4.2.2 ([4])**.** Given a pseudo-square $P = AB\widehat{A}\widehat{B}$, if A or B is periodic, then P is not perfect.

Proof. By contradiction, let us suppose that A is periodic, $A = u^k$ for some $u \in \Sigma^+$ and $k \geq 2$. As a consequence, $\widehat{A} = \widehat{u}^k$ holds. If follows that the boundary word can be factorized as

$$P = (u)(u^{k-1})B(\widehat{u})(\widehat{u}^{k-1})\widehat{B},$$

that is a BN-factorization as a pseudo-hexagon, contradiction. □

Proposition 4.2.2 does not characterize the class of perfect squares. As an example, the polyomino in Fig. 4.4(b) does not have periodic BN-factors, but admits a BN-factorization as a pseudo-hexagon.
A further class of exact polyominoes has been introduced in [20] as a consequence of the following result.

Theorem 4.2.3 (Blondin Massé et al. [20])**.** An exact polyomino tiles the plane as a pseudo-square in at most two distinct ways.

The exact polyominoes satisfying this property constitute the class of *double squares*. Moreover, we call *single square* a perfect pseudo-square that is not a double square. As an example, the polyomino in Fig. 4.3 is in this class.

Example 4.2.4. An example of double square is $P = 101010\overline{1}0\overline{10101010}1\overline{0}$, with its two BN-factorizations

$$P_{(a)} = (1\overline{0}1)(01010)(\overline{1}0\overline{1})(\overline{01010}),$$
$$P_{(b)} = (10101)(0\overline{1}0)(\overline{10101})(\overline{0}1\overline{0}).$$

The induced tilings, both regular, are in Fig. 4.8.

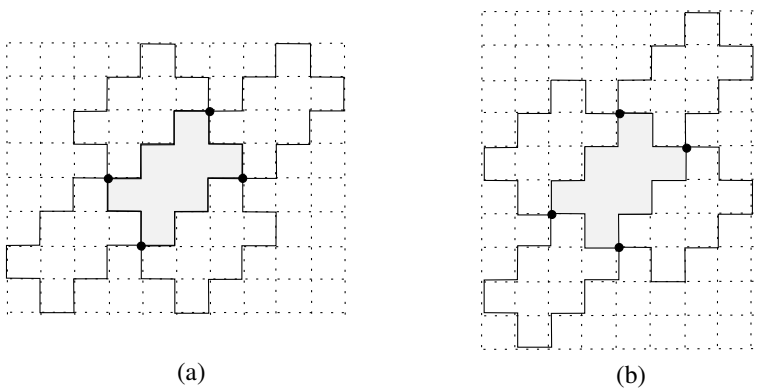


Figure 4.8: Two distinct tilings induced by a double square.

Moreover, as observed by the authors, the proof in [20] can be adapted to show the following result too:

Theorem 4.2.5 (Blondin Massé et al. [20])**.** Pseudo-hexagons and double squares are disjoint classes of exact polyominoes, that is, all double squares are perfect.

4.2.2 Periodicity

In order to study a possible extension of the decomposition theorem to all exact polyominoes, we now focus on the periodicity properties of tiles, relevant when dealing with homogeneous configurations. We put more emphasis on the simpler class of perfect squares.

Theorem 4.2.6 ([4]). Let P be a perfect pseudo-square. Then, each possible tiling induced by P has two directions of periodicity and, as a consequence, is regular.

It is possible to compute the directions of periodicity of the tiling starting from the BN-factorization that induces it:

Theorem 4.2.7 ([4]). Let us consider P a perfect pseudo-square and $P = AB\widehat{A}\widehat{B}$ (one of) its BN-factorization(s). Each BN-factor defines a direction of periodicity in the induced regular tiling,

$$v_1 = (|A|_0 - |A|_{\overline{0}}, |A|_1 - |A|_{\overline{1}}),$$
$$v_2 = (|B|_0 - |B|_{\overline{0}}, |B|_1 - |B|_{\overline{1}}).$$

Proof. We consider the BN-factor B, that we remind matches with the BN-factor $\widehat{B}$ of a copy of P in the induced tiling. Such a translation is defined by the steps made along the side of the polyomino coded by the BN-factor A, so we have to count the number of steps made along the path connecting the sides B and $\widehat{B}$. In the horizontal direction, the number of steps is given by the difference between East and West steps, that is, $|A|_0 - |A|_{\overline{0}}$. Analogously, the vertical ones are given by $|A|_1 - |A|_{\overline{1}}$. Putting all together, we get the vector that identifies the first direction of periodicity of the tiling, $v_1 = (|A|_0 - |A|_{\overline{0}}, |A|_1 - |A|_{\overline{1}})$. The same argument leads to the computation of the second vector v_2, this time starting from the translation defined by the BN-factor A and counting the steps made along the path coded by B. □

Remark 4.2.8. In case P is a double square, the pair $\{v_1, v_2\}$ of directions of periodicity in the first induced tiling is different from the pair of directions of periodicity in the second one.

Example 4.2.9. The cross tile, in Fig. 4.9, is a double square, admitting the following BN-factorizations:

$$P_{(a)} = (1\overline{0}1)(010)(\overline{1}0\overline{1})(\overline{010}),$$
$$P_{(b)} = (101)(0\overline{1}0)(\overline{101})(\overline{0}1\overline{0}).$$

The directions of periodicity in the induced tilings are $v_1 = (-1, 2)$ and $v_2 = (2, 1)$, and $u_1 = (-2, 1)$ and $u_2 = (1, 2)$, respectively.

We underline that if we consider a cell of the polyomino, and then move along one of the directions of periodicity of the tiling, we reach the cell in the same position of a neighbor tile. This trivial observation will become relevant in the study of decomposability.

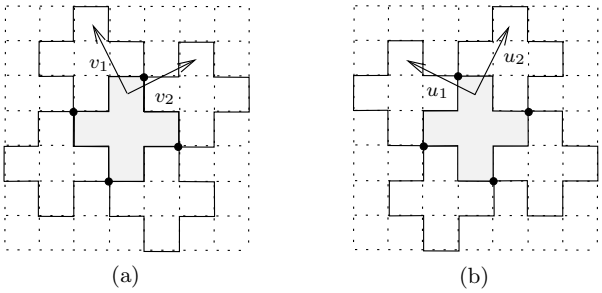


Figure 4.9: The cross tile and the directions of periodicity of the related tilings.

4.3 The decomposition conjecture

In order to provide experimental evidence to (the presence or not of) the decomposition property, we exhaustive generate all the exact polyominoes up to perimeter $n = 20$, and then, for each of them, all the possible h-homogeneous configurations for $2 \leq h \leq \omega$, being ω the area of the involved polyomino. Once a homogeneous configuration A has been generated, we decompose it, if possible, in h disjoint 1-homogeneous configurations, $A_1, \ldots, A_h$, by exploiting the results given in Theorems 4.1.9, 4.2.6 and 4.2.7. In particular, Theorems 4.2.6 and 4.2.7 provide the directions of periodicity of the 1-homogeneous sub-configurations $A_1, \ldots, A_h$, while the existence of such directions is guaranteed by Theorem 4.1.9. We consider perfect pseudo-squares only, so that we have exactly two directions of periodicity for each polyomino.
We proceed as follows: given a perfect pseudo-square W, with $\{v_1, v_2\}$ its directions of periodicity, and A an h-homogeneous configuration w.r.t. W, we look for a set of elements in A that are periodic with respect to v_1 or v_2. If such a set exists, we choose it as A_1, and then iterate the procedure on the configuration $A' = A \setminus A_1$. Two important remarks:

i) The configuration A_1 is 1-homogeneous w.r.t. W, by Theorem 4.1.9.

ii) The configuration A' is $(h-1)$-homogeneous w.r.t. W (trivial by construction).

We iterate until the starting set of points of $\mathbb{Z}^2$ is empty, meaning that a decomposition $A = A_1 \cup \cdots \cup A_h$ has been achieved. In case of negative answer, the starting h-homogeneous configuration A cannot be decomposed, and so the polyomino W gives a counterexample to the conjecture.

4.3.1 Implementation

We implement the described procedure in the algorithm *Exh-Dec*, that consists of two main phases: the exhaustive generation of exact polyominoes and homogeneous configurations, and then their decomposition, if possible, in 1-homogeneous ones. We sketch here the main steps.

Generation of h-homogeneous configurations

Input: a perfect pseudo-square W of area ω and an integer $2 \leq h \leq \lfloor \frac{\omega}{2} \rfloor$.

Step 1. Create an empty matrix A that is big enough to contain the polyomino W surrounded by four copies of itself (basic tiling). A will simulate $\mathbb{Z}^2$.

Step 2. Place the window W at the center of A and insert in W h elements 1s, in all possible ways. For each different configuration, start a new line of computation.

Step 3. Move W on A of a single discrete step, in all possible directions, until you visit all the positions of A. For each new position, complete the assignment of the entries of W to obtain exactly h elements 1s in the window, in all possible ways. If such an assignment is not possible, stop that computation line. For each remaining assignment, start a new line of computation.

Step 4. Iterate Step 3 until the whole matrix A has been visited.

Output: each line of computation that has not been pruned provides an h-homogeneous configuration w.r.t. W.

We underline that:

- ▷ In Step 4, we avoid to place the window W over A in exactly the same position more than once, to reduce the computational cost of the procedure. We only ensure to visit all the entries of the matrix A and complete their assignments. To do this, in our implementation we move the window W over A following a spiral-shaped path.
- ▷ To be sure to complete the assignment in the whole matrix A, we actually initialize a bigger matrix, A', that contains A. This way, the polyomino W can be actually placed over all the possible entries of A, without exceeding its borders.
- ▷ The algorithm works in exponential time w.r.t. the dimension of W, whose area is ω, and the value h. This is true since it computes all the possible h-homogeneous configurations through an exhaustive generation of all the possible assignments. Such a computational cost is of the order of $\Omega(\omega^h)$.
- ▷ Many lines of computation are pruned because of forbidden assignments.

Decomposition

For each configuration we get as an output of the previous step, we look for a (possible) decomposition in 1-homogeneous ones.

Input: A perfect square W and a configuration A that is h-homogeneous w.r.t. W.

Step 1. Compute $\{v_1, v_2\}$ the directions of periodicity of W.

Step 2. Choose an element $x = 1$ in A (at random) and detect all the 1s in A that constitute, with x, a configuration A_1 that is periodic in the direction v_1 or v_2. Remove A_1 from A.

Step 3. Iterate Step 2 for h times.

Output: Flag for the success or failure of the decomposition. If at the end of the procedure A is empty, then the h-homogeneous configuration given in input has been correctly decomposed. Otherwise, W is not h-decomposable.

We underline that:

- ▷ It is necessary to choose W in the class of perfect squares, since this guarantees the existence of two directions of periodicity. The decomposition procedure can be slightly modified if we want to include the analysis of pseudo-hexagons too, keeping in consideration the existence of semi-regular tilings.
- ▷ The failure of the procedure on one configuration only is enough to state that W is not h-decomposable.
- ▷ If all the h-homogeneous configurations are successfully decomposed, we have experimental evidence of the h-decomposability of W, but this is not sufficient for a rigorous proof. Indeed, we have to keep in mind that the matrix A used to simulate the discrete plane $\mathbb{Z}^2$ has finite dimension.

4.3.2 Exhaustive generation of perfect squares

As already mentioned, we test the conjecture on perfect pseudo-squares. The presence of two directions of periodicity in the tilings allows a fast decomposability detection of the homogeneous configurations, since semi-regular tilings are not taken into account.
We exhaustively generate them by generating their boundary words. Even in this case, the algorithm takes an exponential running time, so we stop at perimeter $n = 20$. We perform an exhaustive generation of all the boundary words, and then we keep only those ones coding perfect pseudo-squares. We work up to symmetries and rotations, obtaining: one tile of perimeter 12, two tiles of perimeter 14, seven tiles of perimeter 16, twenty one tiles of perimeter 18, and seventy three tiles of perimeter 20.
We tested the decomposability of perfect squares up to perimeter $n = 18$ for all the admitted homogeneity values h, while for $n = 20$ we achieved only partial results, due to the huge running time of the algorithm that generates homogeneous configurations. Indeed, for each tile W we have a complexity of $\Omega(\omega^h)$, being ω the area of W and h the considered homogeneity value. However, the 2-decomposability has been tested for all of them. We summarize the results concerning the first polyominoes in Table 4.1.

Remark 4.3.1. The computational cost of the algorithm increases with the value of the area of the window ω and of the homogeneity value h, while the perimeter of W is not strictly relevant.

4.3.3 Counterexamples to the decomposition conjecture

Our tests show that the conjecture is false, providing many counterexamples. Several different cases arose, showing that a tile can be either h-decomposable for any possible h, or h-decomposable only for some homogeneity values, or such that the decomposition property does not hold for any admissible h.

ID	Boundary word	Perimeter	Area	Class	Non-decomposability
1	$1\bar{0}1010\bar{1}0\overline{1010}$	12	5	Double square	$--$
2	$11\bar{0}1010\bar{1}0\overline{11010}$	14	7	Single square	$--$
3	$1\bar{0}10100\bar{1}0\overline{10010}$	14	7	Single square	$--$
4	$111\bar{0}1010\bar{1}0\overline{111010}$	16	9	Single square	$--$
5	$11\bar{0}10110\bar{1}0\overline{110110}$	16	8	Single square	$--$
6	$110101000\overline{10110010}$	16	8	Single square	$--$
7	$11\bar{0}10100\bar{1}0\overline{110010}$	16	10	Single square	$h = 5$
8	$1\bar{0}10110\bar{0}1\bar{0}\overline{100110}$	16	8	Single square	$h = 4$
9	$1\bar{0}1010\bar{1}0\bar{1}\overline{01010}$	16	8	Double square	$h = 4$
10	$1\overline{00}10110\bar{1}00\overline{10110}$	16	8	Single square	$h = 4$
11	$1111\bar{0}1010\bar{1}0\overline{1111010}$	18	11	Single square	$--$
12	$1110110101\overline{110111010}$	18	9	Single square	$--$
13	$111\bar{0}11010\overline{11}0\overline{111010}$	18	11	Single square	$--$
14	$11\bar{0}1101101\overline{10}\overline{110110}$	18	10	Double square	$h = 4, 5, 6$
15	$1110101001\overline{01110010}$	18	11	Single square	$--$
16	$111\bar{0}10100\bar{1}0\overline{1110010}$	18	13	Single square	$--$
17	$1101101001\overline{10110010}$	18	11	Single square	$--$
18	$11\bar{0}110100\overline{11}0\overline{110010}$	18	13	Single square	$--$
19	$1\bar{0}1110100\overline{111}0\overline{10010}$	18	13	Single square	$--$
20	$11\bar{0}10110\bar{0}1\bar{0}\overline{1100110}$	18	11	Single square	$--$
21	$1\bar{0}1101100\overline{11}0\overline{100110}$	18	11	Single square	$--$
22	$1\bar{0}10111001\bar{0}\overline{1001110}$	18	9	Single square	$h = 3$
23	$11\bar{0}1010\bar{1}0\bar{1}\overline{1101010}$	18	11	Single square	$--$
24	$111\overline{00}1010\bar{1}0\overline{111010}$	18	10	Single square	$h = 5$
25	$1101\bar{0}1010\bar{1}0\overline{1011010}$	18	8	Single square	$--$
26	$11\bar{0}1\bar{0}1010\bar{1}0\bar{1}\overline{11010}$	18	10	Single square	$h = 5$
27	$1\bar{0}1011010\bar{1}0\overline{1010110}$	18	9	Single square	$--$
28	$11\overline{00}11010\overline{11}00\overline{11010}$	18	10	Single square	$h = 5$
29	$11\overline{00}10110\bar{1}00\overline{110110}$	18	10	Single square	$h = 5$
30	$1\overline{00}101110\bar{1}00\overline{101110}$	18	10	Single square	$h = 5$
31	$1\bar{0}1010100\overline{10}\bar{1}0\overline{10010}$	18	9	Single square	$--$
32	$11\bar{0}101110\bar{1}0\overline{1101110}$	18	9	Single square	$h = 3, 4$

Table 4.1: List of all the perfect pseudo-squares with perimeter $12 \leq n \leq 18$ (exhaustively generated), with the main parameters that characterize them. The last column provides the values, if any, for which the respective polyomino is not h-decomposable.

In this section we focus on the tiles that are not decomposable, showing planar configurations where the decomposition property fails. Then, in Section 4.4, we deduce some common properties that lead to the definition of classes of non-decomposable tiles, as well as some preliminary results that give rise to the formulation of a new conjecture in the field.

Counterexamples for $\mathbf{h = 2}$

We start our tests choosing the smallest homogeneity value, $h = 2$, that is also the fastest analysis from a computational point of view. The algorithm *Exh-Dec* decomposes all the 2-homogeneous configurations w.r.t. polyominoes of perimeter $n \leq 18$ (see Table 4.1); moving to perimeter $n = 20$, we meet the first polyominoes that do not satisfy the decomposition property, and that are depicted in Fig. 4.10.

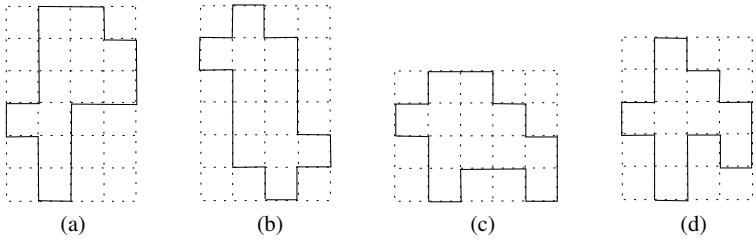


Figure 4.10: Pseudo-squares of perimeter $n = 20$ that do not satisfy the 2-decomposability.

Looking at Figure 4.10, we see that the non-decomposable polyominoes do not share a specific shape or characteristic, and are apparently very dissimilar. For each of them, we present a 2-homogeneous non-decomposable configuration and deepen the study of its characteristics, see Table 4.2. To simplify the notation, we refer to these polyominoes using the letters from Fig. 4.10.

ID	Boundary word	Perimeter	Area	Class	Periodicity	Non-dec
(a)	$11\bar{0}1011110\bar{0}1\bar{0}11001110$	20	12	Single square	$(-1,3)$ and $(3,3)$	$h = 2,4,5,6$
(b)	$1\bar{0}111\bar{0}1010\bar{1}0111\bar{0}1010$	20	12	Single square	$(-2,5)$ and $(2,1)$	$2 \leq h \leq 6$
(c)	$11\bar{0}10100\bar{1}0\bar{1}01\bar{1}01\bar{0}010$	20	12	Single square	$(-1,3)$ and $(4,0)$	$2 \leq h \leq 6$
(d)	$11\bar{0}10110\bar{1}0\bar{1}01\bar{1}01\bar{0}110$	20	10	Single square	$(-1,3)$ and $(3,1)$	$h = 2,4$

Table 4.2: List of the perfect pseudo-squares that are not 2-decomposable, with the main parameters that characterize them. They are all depicted in Fig. 4.10.

Some remarks:

- ▷ The non-decomposability is tested looking at 1-homogeneous sub-configurations, so configurations that, by Theorem 4.1.9, are periodic w.r.t. the directions of periodicity of the tile. It is clear from Fig. 4.11-4.14 that such sub-configurations do not exist in the counterexamples we found.

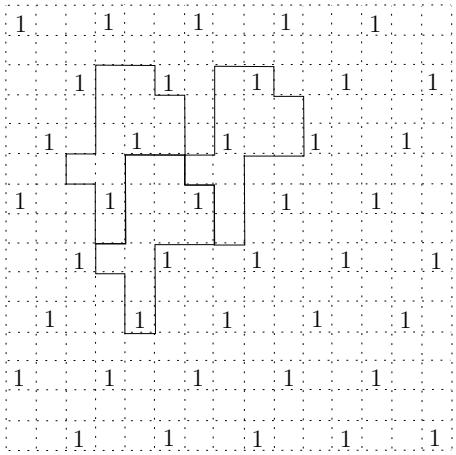


Figure 4.11: A 2-homogeneous configuration that cannot be decomposed.

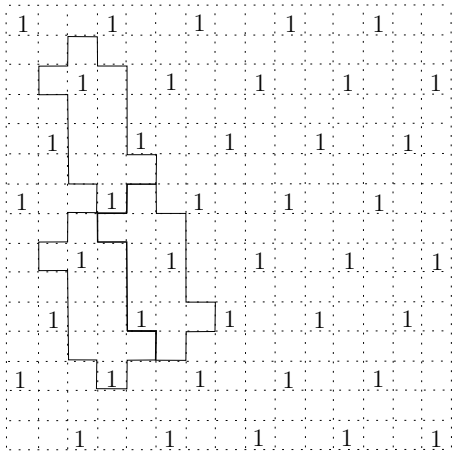


Figure 4.12: A 2-homogeneous configuration that cannot be decomposed.

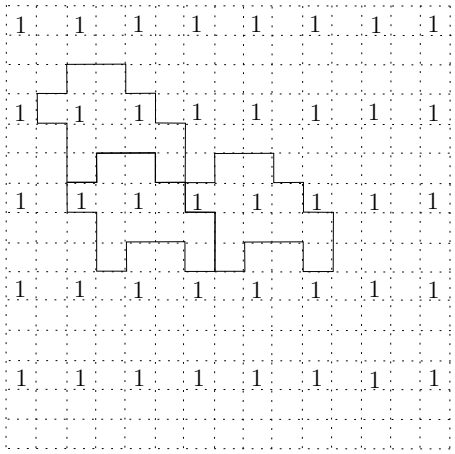


Figure 4.13: A 2-homogeneous configuration that cannot be decomposed.

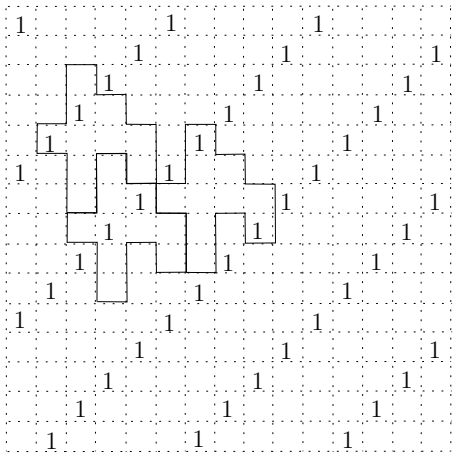


Figure 4.14: A 2-homogeneous configuration that cannot be decomposed.

- ▷ The 2-homogeneous configurations we provide as counterexamples are all periodic with respect to two directions of periodicity, that do *not* coincide with the directions of periodicity of the tiling. Two directions of periodicity are not necessary to build a non-decomposable homogeneous configuration. In our work [4], the reader can find an example of non-decomposable configuration, 2-homogeneous w.r.t. the tile (c), that has one direction of periodicity only.

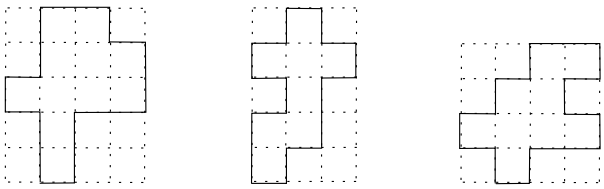
- ▷ At the moment, all the homogeneous and non-decomposable configurations we found have at least one direction of periodicity. Such a behavior reminds the regularity and semi-regularity of the tilings induced by exact polyominoes.

- ▷ If we fix a tiling of the plane, we observe that the lack of a periodical behavior along the two directions of periodicity of the tiling reflects in finding the elements 1s of the configuration in different positions inside the tiles. This will be a key point in the study of properties related to the decomposability, deepened in the next sections.

Counterexamples for $\mathbf{h \geq 2}$

When increasing the homogeneity value h, many counterexamples arise even among polyominoes with smaller perimeter. The reader can find a summary of all the results, exhaustively tested, in the last column of Table 4.1.
We put our attention on some illustrative cases, referring to the polyominoes using the ID in the first column of Table 4.1.

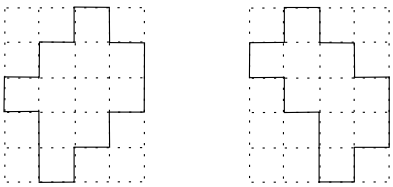
Polyominoes 20, 25 and 31
The tile nr. 20 is similar in shape to the exact polyomino (a) in Fig. 4.10, but satisfies the decomposition property for all admissible h. The same is true for the polyominoes nr. 25 and 31, that can be compared, for their shape, to the tile (c) in Fig. 4.10.



ID 20 ID 25 ID 31

Figure 4.15: The exact polyominoes with ID 20, 25 and 31 in Table 4.1.

Polyominoes 23 and 26
The exact polyominoes nr. 23 and 26 are similar in shape, but while the first one satisfies the decomposition property for any admissible homogeneity value h, the latter is non-decomposable for $h = 5$.



ID 23 ID 26

Figure 4.16: The exact polyominoes with ID 23 and 26 in Table 4.1.

The previous examples suggest again that the shape of the tiles does not play a relevant role in the decomposition of homogeneous configurations. On the other hand, the area of the polyomino seems to be relevant. Indeed, in most cases, when the decomposition property does not hold we notice that the value h is a divisor of the area ω.

4.4 Non-decomposability

Starting from the previous counterexamples, we deduce and formalize some necessary and/or sufficient conditions for the decomposability of homogeneous configurations, as well as classes of exact polyominoes for which the decomposition property does not hold.

Definition 4.4.1. Let us consider A an h-homogeneous configuration w.r.t. an exact polyomino W. We call *pattern* the set of the positions of the elements of A that lie inside W when placing the window on the configuration.

It is clear that there exist at most $\binom{\omega}{h}$ distinct patterns, with ω the area of W and h the homogeneity value of the configuration.

Property 4.4.2. If we fix a tiling of the plane induced by a single square W, a configuration A is 1-homogeneous w.r.t. W if and only if all the copies of W in the tiling display the same pattern. We underline that the same property does *not* hold for double squares.

This is a direct consequence of Theorem 4.1.9 and of the fact that single squares have one pair of directions of periodicity only. The property is lost in case of double squares, due to the existence of two distinct pairs of directions of periodicity.
From Property 4.4.2, we deduce a necessary and sufficient condition for a homogeneous configuration to be decomposable, in case of single squares:

Theorem 4.4.3. Given W a single square of area ω, and A an h-homogeneous configuration w.r.t. W, with $1 \leq h \leq \lfloor \frac{\omega}{2} \rfloor$, let us fix on $\mathbb{Z}^2$ a tiling induced by W. Then, A can be decomposed in h disjoint 1-homogeneous configurations w.r.t. W if and only if all the copies of W in the tiling display the same pattern.

Proof. We fix on $\mathbb{Z}^2$ a tiling induced by the exact polyomino W, and prove the statement. Let us assume that A can be decomposed in h disjoint configurations $A_1, \ldots, A_h$, all 1-homogeneous w.r.t. W. By Theorem 4.1.9, each configuration A_i is periodic w.r.t. the directions of periodicity of W, and so the unique element that appears in each copy of the tile is always in the same position. By the disjointedness of the configurations, it follows that when considering the whole configuration A, the h elements in each tile always appear in the same positions, that is, all the tiles display the same pattern.
On the other hand, if each tile of the tiling displays the same pattern, the h elements that appear inside it follow the directions of periodicity of the tiling. As a consequence of Theorem 4.1.9, there are exactly h configurations all 1-homogeneous w.r.t. W, whose (disjoint) union gives all the elements in A, that so can be decomposed. □

Corollary 4.4.4. Given W a single square of area ω and $2 \leq h \leq \lfloor \frac{\omega}{2} \rfloor$, W is not h-decomposable if and only if there exists a configuration A that is h-homogeneous w.r.t. W and contains two different patterns for a fixed tiling.

Notice that all the examples we provided in Figures 4.11-4.14 are in accordance with the previous result.

4.4.1 Irreducible configurations

Starting from Corollary 4.4.4 and from the periodicity properties that seem to characterize homogeneous configurations, we want to construct *ad hoc* configurations of the plane to define classes of non-decomposable tiles.

Definition 4.4.5. Let us consider W an exact polyomino and A an h-homogeneous configuration w.r.t. W, for some value h. The configuration A is *h-irreducible* if it cannot be decomposed in two configurations A_1 and A_2 that preserve the homogeneity w.r.t. W. Otherwise, A is *reducible*.

It is clear by definition that if a configuration A is h-decomposable, with $h \geq 2$, then it is reducible.

Property 4.4.6. A 2-homogeneous configuration is irreducible if and only if it is not decomposable.

Property 4.4.6 is not true for $h \geq 3$, in general. Indeed, an h-homogeneous configuration could be reducible even if it is not possible to decompose it in 1-homogeneous ones, see next example.

Example 4.4.7. All the configurations we provided as counterexamples in Section 4.3.3 are irreducible. On the other hand, Fig. 4.17 shows a non-decomposable 4-homogeneous configuration that is reducible, since it can be decomposed as the disjoint union of a 3-homogeneous and a 1-homogeneous configuration.

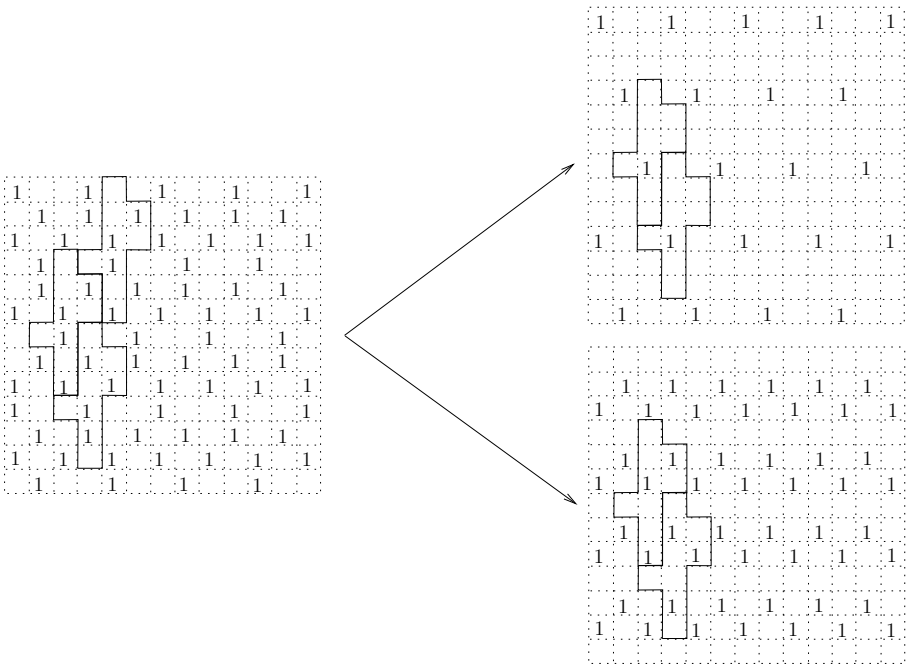


Figure 4.17: A reducible 4-homogeneous configuration (on the left) and its decomposition in irreducible configurations (on the right).

The non-decomposable 4-homogeneous configuration in Fig. 4.17 is obtained as the union of a 3-homogeneous non-decomposable configuration and a 1-homogeneous one. This construction suggests the possibility of defining non-decomposable configurations starting from an irreducible one, say A, and then adding 1-homogeneous (periodic) configurations that are disjoint w.r.t. A. Unfortunately, this construction is not always possible.

Remark 4.4.8. The union of (disjoint) non-decomposable configurations can be decomposable, in general. An Example is given in Fig. 4.18.

As a consequence of our observations, we conclude that two strategies can be exploited for the construction of homogeneous and non-decomposable configurations:

i) Look for irreducible configurations. From our experiments, it seems that an h-homogeneous irreducible configuration can be constructed only if h is a divisor of ω, with ω the area of the considered tile.

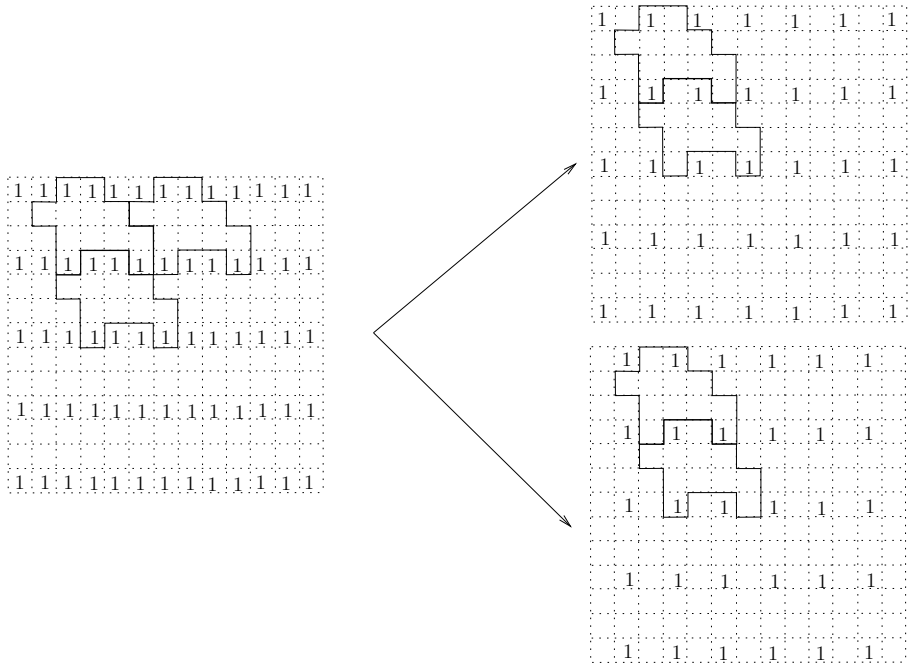


Figure 4.18: The configuration on the left is 4-homogeneous and periodic w.r.t. W, so satisfies the decomposition property. On the other hand, it can be obtained as the union of two disjoint, irreducible 2-homogeneous configurations (on the right).

ii) Starting from an h-homogeneous irreducible configuration A, we can iteratively add disjoint 1-homogeneous configurations, if possible, to build h'-homogeneous non-decomposable configurations, with $h' > h$. As an example, all the non-decomposable h-homogeneous configurations we obtained for the polyomino in Fig. 4.10(c), with $3 \leq h \leq 6$, follow this construction, starting from the irreducible configuration in Fig. 4.13.

4.4.2 Classes of non-decomposable tiles

We start from the observation pointed out at the end of the previous section, point *i*), and we define classes of non-decomposable tiles basing on the area of the polyomino. We introduce the following notation: given W a perfect square of area ω, we define the vector $C = (c_1, \ldots, c_p)$ such that c_i is the number of cells in the i-th column of the polyomino, counted from left to right. Analogously, the vector $R = (r_1, \ldots, r_q)$ contains the number of cells in each row of the polyomino, counted from the top to the bottom. The vectors $D_1 = (d_1, \ldots, d_s)$ and $D_2 = (d'_1, \ldots, d'_t)$ contain the number of cells along the diagonals having direction $e_1 = (1, 1)$ and $e_2 = (-1, 1)$, respectively, starting from the leftmost cell on the top of W.

Remark 4.4.9. Since polyominoes are sets of 4-connected cells, the entries of C, R, D_1 and D_2 are all different from zero.

Example 4.4.10. The vectors C, R, D_1 and D_2 defined for the single square in Fig. 4.19 are

$$\begin{aligned} C &= (1,7,4), \\ R &= (1,2,3,2,2,1,1), \\ D_1 &= (1,2,2,2,2,2,1), \\ D_2 &= (2,2,2,3,1,1,1). \end{aligned}$$

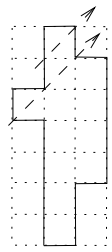


Figure 4.19: The figure illustrates the perfect square described in Example 4.4.10. The direction $e_1 = (1,1)$ along which the vector D_1 is computed is also highlighted.

Starting from the vectors C, R, D_1 and D_2 we construct, if possible, irreducible configurations. We describe the procedure for the values in R, but the same method can be applied to the other vectors.

Let us consider a polyomino W of area ω, with $R = (r_1, \ldots, r_q)$ the vector of its row cardinalities, and a value $1 \leq h \leq \lfloor \frac{\omega}{2} \rfloor$ such that h divides ω. The idea is to partition the values in R in $\frac{\omega}{h}$ disjoint sets such that:

i) The sum of the values in each set is exactly h.

ii) There exists $2 \leq \lambda \leq q-1$ such that the elements in each set can be expressed as $\{r_i, r_{\min\{i+\lambda,q\}}, r_{\min\{i+2\lambda,q\}}, \ldots, r_{\min\{i+\lfloor \frac{q}{\lambda} \rfloor \cdot \lambda, q\}}\}$, for $i = 1, \ldots, \lambda$.

If such a decomposition exists, we consider the configuration of the plane $A_{u,v}$, periodic w.r.t. the directions $u = (1,0)$ and $v = (0,\lambda)$, that is h-homogeneous w.r.t. W by construction. Indeed, we have partitioned the rows of the polyomino W in groups such that each group contains exactly h cells, and rows belonging to the same group are always at distance $(0,\lambda)$ each other. Example 4.4.11, concerning the tile in Fig. 4.19, clarifies the described construction.

Example 4.4.11. The polyomino W in Fig. 4.19 has area $\omega = 12$ and row cardinalities equal to $R = (1,2,3,2,2,1,1)$. Choosing $h = 4$, we can partition the values of R in three groups such that the sum of the elements in each group is four, and the distance between the rows in each group is constant and equal to $\lambda = 3$. We underline in the following vectors the elements belonging to each group, highlighted in Fig. 4.20 as well.

$$\begin{aligned} R_1 &= (\underline{1},2,3,\underline{2},2,1,\underline{1}), \\ R_2 &= (1,\underline{2},3,2,\underline{2},1,1), \\ R_3 &= (1,2,\underline{3},2,2,\underline{1},1). \end{aligned}$$

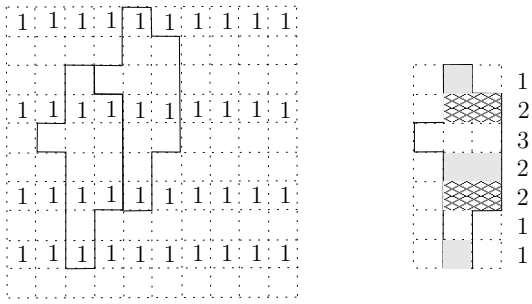


Figure 4.20: A 4-homogeneous irreducible configuration that is constructed starting from the vector R of row cardinalities.

We consider now the directions $u = (1, 0)$ and $v = (0, 3)$, and the configuration of the plane $A_{u,v}$ that is periodic along them. It is clear from Fig. 4.20 that such a configuration is 4-homogeneous w.r.t. W. Moreover, it cannot be decomposed in 1-homogeneous disjoint configurations, since distinct patterns are present in the same tiling.

After the construction of $A_{u,v}$, two cases arise:

i) The directions $u = (1, 0)$ and $v = (0, \lambda)$ coincide with the directions of periodicity of W. In this case, $A_{u,v}$ can be decomposed in disjoint configurations that are 1-homogeneous w.r.t. W (Theorem 4.1.9), and the construction does not give any hint concerning the h-decomposability of W.

ii) The directions $u = (1, 0)$ and $v = (0, \lambda)$ do not coincide with the directions of periodicity of W, as in case of Example 4.4.11, and so the polyomino is not h-decomposable. Moreover, in this case the configuration $A_{u,v}$ is also irreducible, by construction.

As already mentioned, the same construction can be made starting from the vectors C, D_1 and D_2 as well. We describe here some examples, also considering a case in which such a construction does non provide the non-decomposability of the polyomino.

Example 4.4.12. The polyomino $W = 1\overline{0}10100\overline{1}00\overline{1}0\overline{1}\overline{00}1\overline{0}\overline{0}10$, of area $\omega = 10$, has $v_1 = (5, 0)$ and $v_2 = (-1, 2)$ as directions of periodicity. The vector collecting its column cardinalities is $C = (1, 3, 2, 1, 2, 1)$, whose elements can be grouped in two sets having sum $h = 5$ and values at distance $\lambda = 2$. The configuration in Fig. 4.21 is periodic along $(0, 1)$ and $(2, 0)$. As a consequence, it is 5-homogeneous w.r.t. W, but cannot be decomposed.

Example 4.4.13. The pseudo-square $W = 1101010010\overline{110010}$, Fig. 4.22, has area $\omega = 8$ and directions of periodicity $v_1 = (1, 3)$ and $v_2 = (3, 1)$. Its diagonal cardinalities are $D_1 = (3, 4, 1)$ and $D_2 = (1, 1, 1, 2, 1, 1, 1)$. In both cases, we can group the cells of the polyomino for an appropriate h:

$h = 4$**:** the entries of D_1 can be grouped in two sets by choosing elements at distance $\lambda_1 = 2$ each other. The configuration of the plane having directions of periodicity

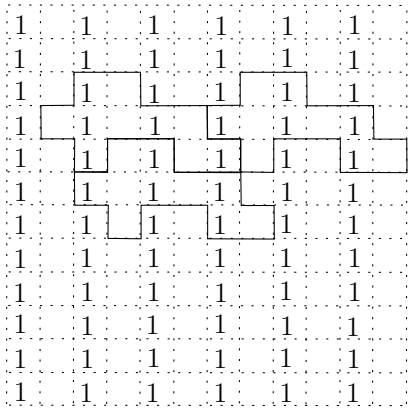


Figure 4.21: An irreducible 5-homogeneous configuration obtained starting from the column cardinalities of the polyomino.

$(1,1)$ and $(2,0)$, Fig. 4.22(a), is 4-homogeneous w.r.t. W, but can be decomposed since it follows the directions of periodicity of the tiling too.

$h = 2$**:** the same construction leads to a decomposable 2-homogeneous configuration, Fig. 4.22(b), grouping the elements of D_2 in four sets. In this case, the elements are at distance $\lambda_2 = 4$.

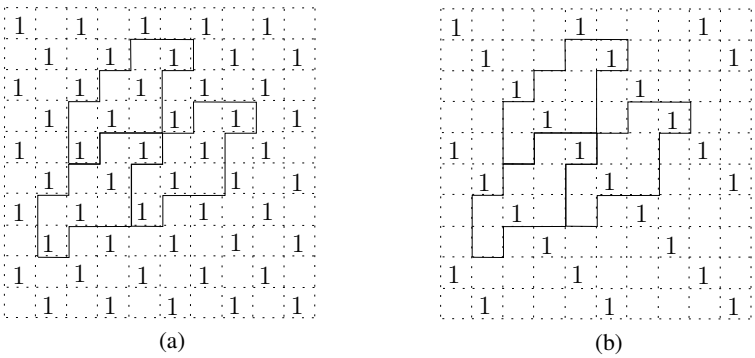


Figure 4.22: An exact polyomino whose diagonal cardinalities do not provide any counterexample to the decomposability property.

Generalization

The construction we described for the vectors C, R, D_1 and D_2 can be generalized to any discrete direction v, i.e., collecting the number of cells of a polyomino W along a generic vector v, grouping them, if possible, in sets whose elements sum to the same value h, and then constructing the corresponding periodic configuration of the plane. As in the previous cases, such a configuration is h-homogeneous w.r.t. W by construction and, if

its directions of periodicity are different from those ones of the tiling, it is irreducible too.
For the sake of clarity, we propose an example in which the construction works for the direction $v = (2, 1)$. The involved polyomino is the counterexample depicted in Figure 4.10(a).

Example 4.4.14. Let us consider the exact polyomino $W = 11\bar{0}1011100\bar{1}0\overline{11001110}$ of area $\omega = 12$, and the vector $v = (1, 2)$. If we count the number of cells of W along v, we collect the vector $V = (1, 2, 2, 2, 2, 2, 1)$, see Fig. 4.23 on the right.
Choosing $h = 2$, the elements of V can be easily grouped in six sets summing to h. Indeed, it is sufficient to merge the two values 1, at distance $\lambda = 5$, to get such a partition. The parameter λ, together with v, provide the directions of periodicity of the configuration A, depicted in Fig. 4.23 on the left. A is 2-homogeneous w.r.t. W by construction, but cannot be decomposed since its directions of periodicity do not coincide with those ones of W. As a conclusion, we have constructed a 2-homogeneous and irreducible configuration starting from a grouping of the cells of W along $v = (1, 2)$, thus showing again that the decomposition property does not hold for the considered exact polyomino.

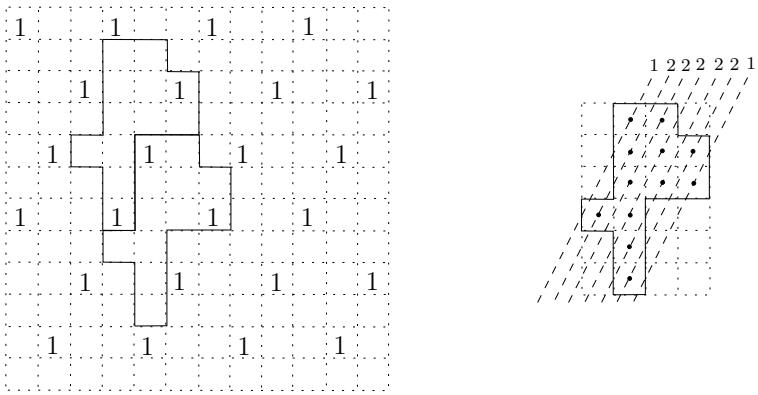


Figure 4.23: The construction of a 2-homogeneous irreducible configuration through a grouping of the cells of the polyomino.

Conclusion and final remarks

As a conclusion of the chapter, we propose a short list of possible research lines that arise from our work.

- ▷ In our work, we did not consider pseudo-hexagons and semi-regular tilings. It would be of interest to extend the study of the decomposition property to this class of exact polyominoes, also exploiting the decomposition proof given for rectangles.

  Problem 1. Consider pseudo-hexagons and exploit their similarities with rectangles, first of all concerning the existence of semi-regular tilings.

- ▷ All the homogeneous configurations we found, through construction or via our algorithm, always admit at least one direction of periodicity, thus reminding to the behavior of regular and semi-regular tilings.

  Problem 2. Study the periodicity properties of homogeneous configurations.

- ▷ The construction of irreducible configurations seems not to be linked with the shape of the considered tile, but with the possibility of grouping its cells along a suitable direction. As a new research line in the field, we propose to deepen the study of such notions, as well as the development of new techniques and methods for the detection of irreducible configurations.

  Problem 3. Study the properties of tiles with respect to the possibility of grouping their cells according to some (rigorous) criteria.

- ▷ From all our experiments, as well as from the observations and constructions we described in Sections 4.4.1 and 4.4.2, it is plausible to suppose that the construction of an h-homogeneous irreducible configuration w.r.t. an exact polyomino W is possible only if h is a divisor of the area of the tile ω. On the other hand, if h does not divide ω, then an h-homogeneous non-decomposable configuration, if it exists, can be constructed starting from an irreducible one only. As a consequence, we leave as an open problem the following

  Conjecture. If the area of a perfect pseudo-square W is a prime number, then the decomposition property holds for any admissible h.

Chapter 5

Prime double square polyominoes

In this chapter, we focus on the class of double square tiles, more specifically, on the subclass designed as *prime* in [21]. In order to prove a conjecture in [21], we characterize the boundary words that code prime double square polyominoes, and then carry on the exhaustive generation and enumeration, with respect to the semi-perimeter, of a subclass of them.
The results described in this chapter are joint work with A. Frosini and S. Rinaldi in [6, 7, 11].

5.1 Introduction and preliminaries

In this section we give the motivations of our work together with the notations and the mathematical tools we are going to use.

Prime double squares and the couple free conjecture

We address the reader to Section 4.1, Chapter 4, for the basic concepts and results concerning polyominoes and exact tiles. We keep the same notations too.
We recall here the definition of double square, as well as relevant results about this class of exact polyominoes.
A *double square* is an exact polyomino admitting exactly two different BN-factorizations as a pseudo-square, i.e., $P = AB\widehat{A}\widehat{B} \equiv XY\widehat{X}\widehat{Y}$. We remark that double squares do not admit any further BN-factorization as a pseudo-hexagon.
From its double BN-factorization, the boundary word coding a polyomino in this class can always be written as

$$P = w_1w_2w_3w_4w_5w_6w_7w_8, \tag{5.1}$$

where $A = w_1w_2$, $B = w_3w_4$, $\widehat{A} = w_5w_6$, $\widehat{B} = w_7w_8$, and $X = w_2w_3$, $Y = w_4w_5$, $\widehat{X} = w_6w_7$, $\widehat{Y} = w_8w_1$, with $w_i \neq \varepsilon$ for all $i = 1, \ldots, 8$ (Corollary 6 in [24])[1].
Moreover, the following property holds:

[1] Sometimes, we will use the notation w_i, with $i > 8$, to point out the factors of P. In these cases, the index has to be intended as $i - 8$, being P a circular word.

Property 5.1.1 (Blondin Massé et al. [21]). For $i = 1, \dots, 8$, there exist unique words $u_i, v_i \in \Sigma^*$ and unique $n_i \geq 0$ such that

$$\begin{cases} w_i = (u_i v_i)^{n_i}, \\ \widehat{w}_{i-3} w_{i-1} = u_i v_i. \end{cases}$$

Double squares can be composed to obtain new exact polyominoes, see Example 5.1.2. To introduce this operation, we need the notion of *homologous morphism*. A morphism is a function that preserves the concatenation of letters, i.e., $\varphi : \Sigma \to \Sigma^*$ such that $\varphi(\alpha\beta) = \varphi(\alpha)\varphi(\beta)$ for any $\alpha, \beta \in \Sigma$. A morphism is *homologous* if it preserves the hat operation too, i.e., $\varphi(\widehat{A}) = \widehat{\varphi(A)}$ for all $A \in \Sigma^*$. If not differently specified, we work under the assumption of homologous morphism.
Given a pseudo-square $P = AB\widehat{A}\widehat{B}$, the *trivial morphism* φ_P is the homologous morphism that maps the unit square $U = 10\overline{1}\overline{0}$ in P, and it is defined as $\varphi_P(1) = A$, $\varphi_P(0) = B$, and, consequently, $\varphi_P(\overline{1}) = \widehat{A}$, $\varphi_P(\overline{0}) = \widehat{B}$. Apart from this trivial case, in general the boundary word P of an exact polyomino can be obtained, starting from the unit square, through the composition of two or more homologous morphisms. In other words, the trivial morphism φ_P can be decomposed, in general, in two or more morphisms, $\varphi = \varphi_1 \circ \varphi_2 \circ \cdots \circ \varphi_t$ for some $t \geq 1$.

Example 5.1.2. We consider the following pseudo-squares, represented in Fig. 5.1:

$$\begin{aligned} P_{(a)} &= (10101)(0\overline{1}0\overline{1}0)(\overline{10101})(\overline{0}1\overline{0}1\overline{0}) \\ P_{(b)} &= (11\overline{0}11)(01010)(\overline{11}0\overline{11})(\overline{01010}). \end{aligned}$$

The trivial morphisms that define them are, respectively,

$$\begin{aligned} &\varphi_{P_{(a)}}(0) = 0\overline{1}0\overline{1}0, \quad \varphi_{P_{(a)}}(1) = 10101, \quad \varphi_{P_{(a)}}(\overline{0}) = \overline{0}1\overline{0}1\overline{0}, \quad \varphi_{P_{(a)}}(\overline{1}) = \overline{10101}, \\ &\varphi_{P_{(b)}}(0) = 01010, \quad \varphi_{P_{(b)}}(1) = 11\overline{0}11, \quad \varphi_{P_{(b)}}(\overline{0}) = \overline{01010}, \quad \varphi_{P_{(b)}}(\overline{1}) = \overline{11}0\overline{11}. \end{aligned}$$

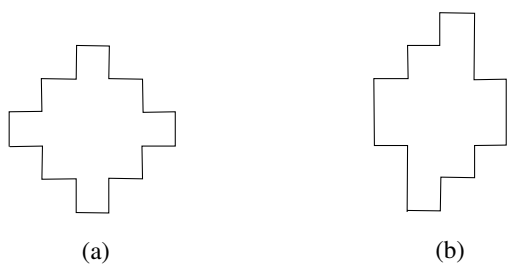


Figure 5.1: The pseudo-squares having boundary word $P_{(a)}$, on the left, and $P_{(b)}$, on the right.

We consider the composition of the two trivial morphisms defined above, $\psi_1 = \varphi_{P_{(a)}} \circ \varphi_{P_{(b)}}$ and $\psi_2 = \varphi_{P_{(b)}} \circ \varphi_{P_{(a)}}$, that lead to the construction of the two pseudo-squares in Fig. 5.2. Indeed, their boundary words

$$Q_{(a)} = (1010110101\bar{0}1\bar{0}1\bar{0}1010110101)(0\bar{1}0\bar{1}0101010\bar{1}0\bar{1}0101010\bar{1}0\bar{1}0)$$
$$(\overline{1010110101}0\bar{1}0\bar{1}0\overline{1010110101})(\bar{0}1\bar{0}1010101010\bar{1}0\bar{1}0101010\bar{1}0\bar{1}0)$$

$$Q_{(b)} = (11\bar{0}110101011\bar{0}110101011\bar{0}11)(0101010\overline{11}0\overline{11}0101010\overline{11}0\overline{11}01010)$$
$$(\overline{11}0\overline{11}0101011\overline{0}110101011\overline{0}\overline{11})(\bar{0}1010110\overline{11}0\overline{11}0101011\overline{0}110\overline{11}01010)$$

can be obtained as $Q_{(a)} = \varphi_{P_{(a)}}\big(\varphi_{P_{(b)}}(U)\big)$ and $Q_{(b)} = \varphi_{P_{(b)}}\big(\varphi_{P_{(a)}}(U)\big)$, with $U = 10\overline{10}$ the boundary word of the unit square.

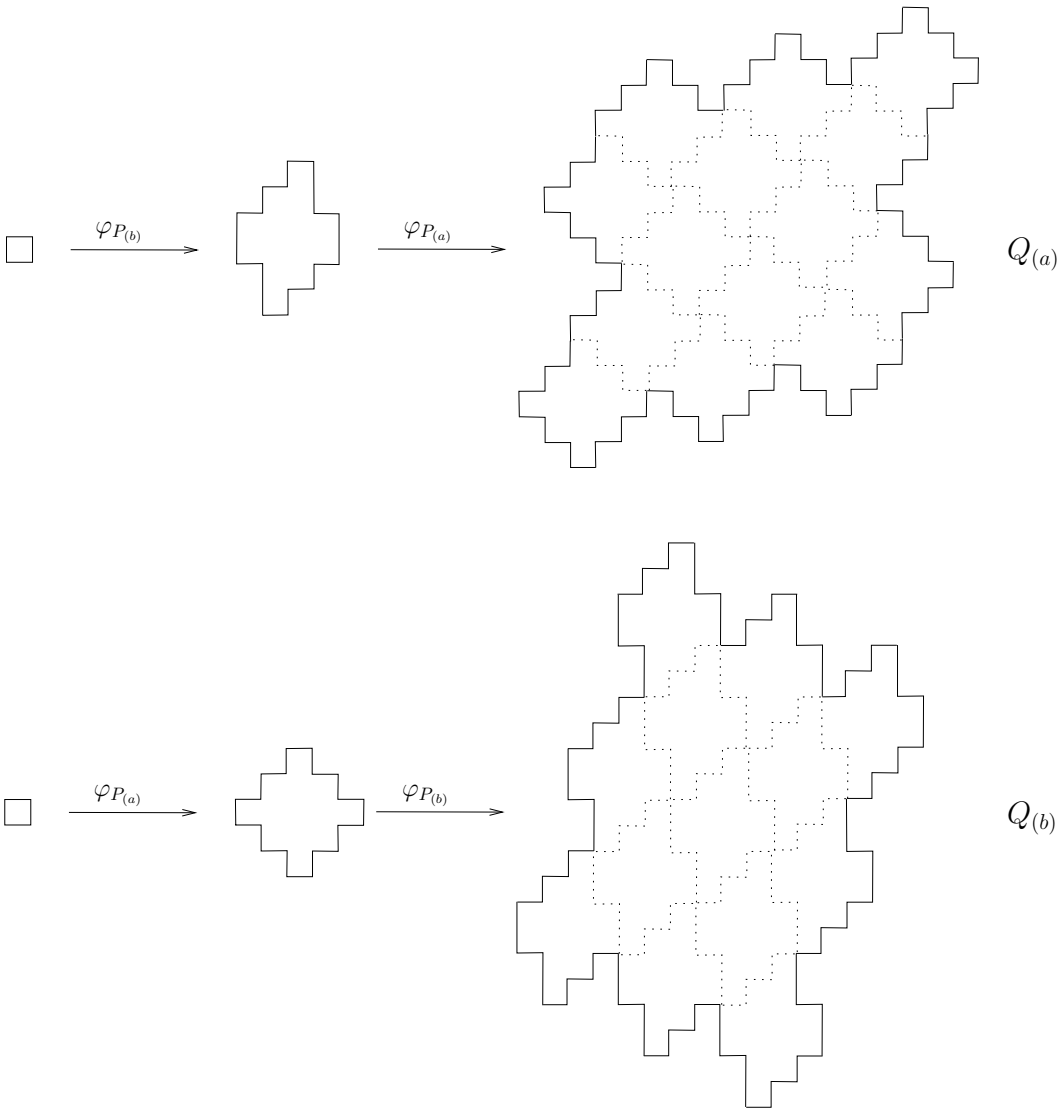


Figure 5.2: The two tiles, $Q_{(a)}$ on the left and $Q_{(b)}$ on the right, obtained by combining the trivial morphisms that define the squares in Fig. 5.1.

Notice that the composition of homologous morphisms is *not* commutative, and that

such a composition consists, essentially, in constructing new exact tiles by gluing copies of smaller exact tiles, see Fig. 5.2.

The authors in [21] defined the subclass of *prime double squares* as follows: a double square is *prime* if its boundary word P is such that, for any homologous morphism φ, $P = \varphi(U)$ implies $P = U$ or $\varphi = \varphi_P$. In words, P can be obtained from the unit square through the application of the trivial morphism only or, equivalently, the trivial morphism φ_P cannot be obtained as the composition of more non-trivial morphisms.

Example 5.1.3. The following double squares are all prime:

$$\begin{aligned} P_{(a)} &= (1)(01)(0)(\overline{1}0)(\overline{1})(\overline{01})(\overline{0})(\overline{10}) \\ P_{(b)} &= (1)(0101)(0)(\overline{1}0)(\overline{1})(\overline{0101})(\overline{0})(1\overline{0}) \\ P_{(c)} &= (1)(0101)(0)(\overline{1}0\overline{1}0)(\overline{1})(\overline{0101})(\overline{0})(1\overline{0}1\overline{0}). \end{aligned}$$

In each boundary word we highlighted the double BN-factorization as in Eq. (5.1), while the corresponding polyominoes are depicted in Fig. 5.3.

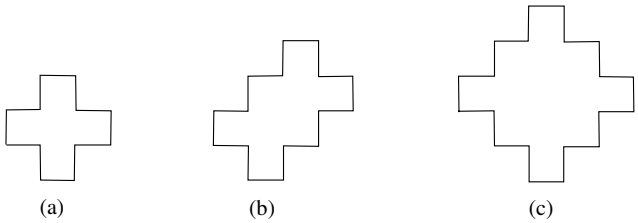


Figure 5.3: Three examples of prime double squares. Among them, the cross (a) and the diamond (c).

The adjective *prime* refers to the fact that this type of double square cannot be obtained as the composition of smaller polyominoes, and, for this reason, can be seen as the seed for the generation of all double square tiles, that can indeed be produced by applying homologous morphisms to the prime ones.
The authors in [21] proposed an algorithm for the exhaustive generation of double squares that does not involve the notion of homologous morphism. Even if such a procedure generates all the double squares, some drawbacks are present: among the produced objects, also sets of cells that are not polyominoes are present and, in general, the same double square is produced more than once. As a consequence, their enumeration cannot be deduced from the algorithm itself, thus resulting in an open problem.
While analyzing the properties of prime double squares, Blondin Massé et al. noticed that in their boundary word two consecutive equal letters never occur. So, they proposed the following

Conjecture (Blondin Massé et al. [21]): given P the boundary word of a double square and $\alpha \in \Sigma$, if P is prime, then $\alpha\alpha$ is not a factor of P.

We started our investigation on prime double squares with the aim of proving the conjecture above, and finally reached a complete characterization of the boundary words of these polyominoes.

We conclude introducing the following notations: given a symbol α of the alphabet Σ and a word $w \in \Sigma^*$, we write $\alpha\alpha \subseteq w$ (respectively, $\alpha\alpha \not\subseteq w$) to point out that $\alpha\alpha$ is (respectively, is not) a factor of w. On the contrary, we say that the word w is *couple free* if $\alpha\alpha \not\subseteq w$ for any $\alpha \in \Sigma$, in other words, two consecutive equal letters do not occur in w. From now on, we refer to the conjecture proposed in [21] as the *couple free conjecture*.

Properties of the boundary word of prime double squares

We present here some preliminary results in the literature, mainly from [21], concerning properties that hold for the boundary word of double squares and prime double squares. In the next section, we exploit such results to achieve a standard factorization for the generic boundary word of a prime double square.

Theorem 5.1.4 (Blondin Massé et al. [21])**.** Given a double square $P = AB\widehat{A}\widehat{B} \equiv XY\widehat{X}\widehat{Y}$, if P is prime, then all its BN-factors are palindrome.

Lemma 5.1.5 (Blondin Massé et al. [21])**.** Let $P = w_1w_2w_3w_4w_5w_6w_7w_8$ be the boundary word of a prime double square, factorized as in (5.1). Then, $w_{i+1} = \overline{w}_i$ holds, for all $i = 1, \ldots, 4$.

Combining Lemma 5.1.5 and Property 5.1.1, and starting form Eq. (5.1), we write the boundary word of a prime double square in the following form,

$$\begin{aligned} P = {} & (\tilde{w}_2\overline{w}_4)^{n_1}u_1(\tilde{w}_3w_1)^{n_2}u_2(\tilde{w}_4w_2)^{n_3}u_3(\widehat{w}_1w_3)^{n_4}u_4 \\ & (\widehat{w}_2w_4)^{n_1}\widehat{u}_1(\widehat{w}_3\overline{w}_1)^{n_2}\widehat{u}_2(\widehat{w}_4\overline{w}_2)^{n_3}\widehat{u}_3(\tilde{w}_1\overline{w}_3)^{n_4}\widehat{u}_4. \end{aligned} \tag{5.2}$$

The factorization in (5.2) is reached after noticing that $w_{i+1} = \overline{w}_i$ and $w_i = (u_iv_i)^{n_i}$ imply $u_{i+4} = \overline{u}_i$, $v_{i+1} = \overline{v}_i$ and $n_{i+4} = n_i$, for $i = 1, \ldots, 4$, and then writing u_iv_i in terms of w_{i-1} and w_{i-3} (see Property 5.1.1).
Moreover, the exponents n_i that appear in (5.2) satisfy the following property:

Lemma 5.1.6 (Blondin Massé et al. [21])**.** Given the boundary word P of a prime double square as in (5.2), there are no two consecutive exponents n_1, n_2, n_3 and n_4 that are different from zero.

As a direct consequence of Lemma 5.1.6, we proved the following result:

Theorem 5.1.7 ([6])**.** The boundary word P of a prime double square, factorized as in (5.2), can always be written in terms of the factors u_i only, for $i = 1, \ldots, 4$.

Proof. If $n_i = 0$ for all possible i, then Eq. (5.2) turns out to be

$$P = u_1u_2u_3u_4\overline{u}_1\overline{u}_2\overline{u}_3\overline{u}_4,$$

and it is expressed in terms of the factors u_i only, for $i = 1, \ldots, 4$.
So, let us assume without loss of generality $n_1 > 0$ (in the other cases, the proof is similar). By Lemma 5.1.6, $n_2 = n_4 = 0$ and so, comparing Eq. (5.2) and Eq. (5.1), we immediately argue $w_2 = u_2$ and $w_4 = u_4$. Since w_1 and w_3 are expressed in terms of w_2, w_4, u_1 and u_3 only, see (5.2), the proof is given. □

We conclude this section providing the generic form of the boundary word of a prime double square, in Eq. (5.3). From now on, we will use a vertical bar to point out the half of the boundary word, and dashed vertical bars to separate the factors w_i, for $i = 1, \dots, 8$.

$$\begin{aligned} P = (\tilde{u}_2\overline{u}_4)^{n_1}u_1 \vdots (\tilde{u}_3u_1)^{n_2}u_2 \vdots (\tilde{u}_4u_2)^{n_3}u_3 \vdots (\widehat{u}_1u_3)^{n_4}u_4| \\ (\widehat{u}_2u_4)^{n_1}\overline{u}_1 \vdots (\widehat{u}_3\overline{u}_1)^{n_2}\overline{u}_2 \vdots (\widehat{u}_4\overline{u}_2)^{n_3}\overline{u}_3 \vdots (\tilde{u}_1\overline{u}_3)^{n_4}\overline{u}_4. \end{aligned} \tag{5.3}$$

Remark 5.1.8. By Lemma 5.1.6, Eq. (5.3) is always well defined, since $n_1 = n_3 = 0$ or $n_2 = n_4 = 0$ (or both) hold. As a consequence, the factors w_i in (5.2) can always be expressed in terms of u_i only.

5.2 Standard factorization of prime double squares

We analyze the presence of a factor of type $\alpha\alpha$ in the boundary word of a prime double square, with $\alpha \in \Sigma$, and finally introduce a standard form for its factorization.

5.2.1 Possible position of consecutive equal letters

Theorem 5.2.1 ([6]). Let $P = AB\widehat{A}\widehat{B}$ be the BN-factorization of a prime double square, and $\alpha \in \Sigma$. If $\alpha\alpha \subseteq P$, then $\alpha\alpha \subseteq A$ or $\alpha\alpha \subseteq B$.

Proof. The proof proceeds by contradiction. Let us suppose that $\alpha\alpha$ is split between two BN-factors, i.e., there exist $a_1, a_2 \in \Sigma^*$ such that $A = a_1\alpha$ and $B = \alpha a_2$. Being P prime, from Theorem 5.1.4 we have that the BN-factors are such that $A = (\alpha b_1\alpha)$ and $B = (\alpha b_2\alpha)$, for some $b_1, b_2 \in \Sigma^*$. The boundary word turns out to be

$$P = (\alpha b_1\alpha)(\alpha b_2\alpha)(\overline{\alpha}\widehat{b}_1\overline{\alpha})(\overline{\alpha}\widehat{b}_2\overline{\alpha}).$$

Being $\alpha\overline{\alpha}$ a closed path and a factor of P, we reached a contradiction with the definition of polyomino, and this concludes the proof. □

From the previous result, two corollaries trivially follow:

Corollary 5.2.2 ([6]). If $P = w_1w_2w_3w_4\overline{w}_1\overline{w}_2\overline{w}_3\overline{w}_4$ is the boundary word of a prime double square, and $\alpha\alpha \subseteq P$ for some $\alpha \in \Sigma$, then the factor $\alpha\alpha$ cannot be split between two consecutive factors w_i and w_{i+1}. In other words, $\alpha\alpha \subseteq w_i$ for some $i = 1, \dots, 4$.

The result is a direct consequence of Theorem 5.2.1.

Corollary 5.2.3 ([6]). If $P = w_1w_2w_3w_4w_5w_6w_7w_8$ is the boundary word of a prime double square, then $|w_i| \neq |w_{i+1}|$ for all $i = 1, \dots, 8$.

Proof. By contradiction, let us suppose $|w_i| = |w_{i+1}|$ for some $i = 1, \dots, 8$. Since P is prime and w_iw_{i+1} is one of its BN-factors, say A, from Theorem 5.1.4 we know that it is palindrome, i.e., $A = w_iw_{i+1} = \tilde{w}_{i+1}\tilde{w}_i$. From the hypothesis on the lengths and the palindromicity, $A = w_i\tilde{w}_i$, meaning that the first letter of w_{i+1}, say α, coincides with the last letter of w_i. So, $\alpha\alpha \subseteq w_iw_{i+1}$ and it is split between the two factors, in contradiction with Corollary 5.2.2. □

All the previous results can be exploited to prove a generalization of Theorem 5.2.1, concerning the factors u_i that appear in Eq. (5.3), for $i = 1, \dots, 4$.

Theorem 5.2.4 ([6]). Let us consider P the boundary word of a prime double square in form (5.3), and $\alpha \in \Sigma$. If $\alpha\alpha \in P$, then $\alpha\alpha \subseteq u_i$ for some $i = 1, \dots, 4$. In words, a factor $\alpha\alpha$ is entirely contained in some u_i, for $i = 1, \dots, 4$, and cannot be split between two consecutive factors u_i.

Proof. From Corollary 5.2.2, the factor $\alpha\alpha$ is entirely contained in w_i, for some $i = 1, \dots, 4$. Without loss of generality, we suppose $\alpha\alpha \subseteq w_1 = (\tilde{u}_2\overline{u}_4)^{n_1}u_1$.
The proof proceeds by contradiction: we suppose that $\alpha\alpha$ is split between two factors u_i, by considering all the possible cases for the indices i and the values of n_1 and n_3.

$n_1 \geq 2$: we can suppose $\alpha\alpha \subseteq \overline{u}_4\tilde{u}_2$. By hypothesis, the last letter of $\overline{u}_4$ and the first letter of $\tilde{u}_2$ is α, so that u_4 ends and $\widehat{u}_2$ begins with $\overline{\alpha}$. So, being $n_1 > 0$, in the boundary word P we have

$$w_4|w_5 = \dots u_4|\widehat{u}_2 \cdots = \dots \overline{\alpha}|\overline{\alpha} \dots,$$

in contradiction with Corollary 5.2.2.

$n_1 > 0$ and $\alpha\alpha \subseteq \tilde{u}_2\overline{u}_4$: $\tilde{u}_2$ ends and $\overline{u}_4$ begins with the letter α. We now consider the factor $w_3 = (\tilde{u}_4u_2)^{n_3}u_3$. If $n_3 > 0$, then $\overline{\alpha}\alpha \subseteq \tilde{u}_4u_2$, and the polyomino self-intersects, contradiction. So, $n_3 = 0$.

By the hypothesis on n_1 and Lemma 5.1.6, $n_2 = n_4 = 0$ holds, and so we have in P the BN-factor $X = w_2w_3 = u_2u_3$, that is palindrome by Theorem 5.1.4. It follows that u_3 ends with the letter α, and so $w_3w_4 = u_3u_4$ contains the factor $\alpha\overline{\alpha}$, again in contradiction with the definition of polyomino.

$n_1 > 0$ and $\alpha\alpha \subseteq \overline{u}_4u_1$: u_4 ends with the letter $\overline{\alpha}$ and u_1 begins with the letter α. By Property 5.1.1, first equation, w_1 begins with α as well. As a consequence, $\overline{\alpha\alpha} \subseteq w_4\overline{w}_1$, in contradiction with Corollary 5.2.2.

Since we reached a contradiction for all the possible cases, the thesis follows. □

5.2.2 The standard factorization

The aim of this section is to provide a unique form in which the boundary word of any prime double square can be expressed. We start from (5.3).
First of all, we recall that the boundary word P is a circular word, since it is defined up to the starting point of the coding of the polyomino. So, we fix such a starting point by adding a further hypothesis on the factors u_i, that is,

$$|u_1| \leq |u_i| \text{ for all } i = 2, 3, 4. \tag{5.4}$$

From now on, condition (5.4) will always hold.

Let us analyze the BN-factors of P, that we recall are palindrome by Theorem 5.1.4:

$$\begin{aligned} A &= w_1 w_2 = (\tilde{u}_2 \overline{u}_4)^{n_1} u_1 (\tilde{u}_3 u_1)^{n_2} u_2, \\ B &= w_3 w_4 = (\tilde{u}_4 u_2)^{n_3} u_3 (\widehat{u}_1 u_3)^{n_4} u_4, \\ X &= w_2 w_3 = (\tilde{u}_3 u_1)^{n_2} u_2 (\tilde{u}_4 u_2)^{n_3} u_3, \\ Y &= w_4 \widehat{w}_1 = (\widehat{u}_1 u_3)^{n_4} u_4 (\widehat{u}_2 u_4)^{n_1} \widehat{u}_1. \end{aligned}$$

From condition (5.4), in particular u_1 has minimum length, for any possible choice of the values of n_i it is possible to express $u_2 = k\tilde{u}_1$ and $u_4 = \widehat{u}_1\overline{p}$, for some proper non-empty words $k, p \in \Sigma^+$. Notice that an analogous for the factor u_3 cannot be deduced, since we do not know its mutual length w.r.t. u_2 and u_4.
From this remark, and considering all the possible combinations of values for the exponents n_i, we deduce all the feasible forms for the boundary word of a prime double square:

Theorem 5.2.5 ([6])**.** If P is the boundary word of a prime double square as in (5.3), then it has one of the following seven forms:

(a) $P = u_1 \vdots k\tilde{u}_1 \vdots u_3 \vdots \widehat{u}_1\overline{p} | \overline{u}_1 \vdots \overline{k}\widehat{u}_1 \vdots \overline{u}_3 \vdots \tilde{u}_1 p$, with $k, p \in \Sigma^+$ and palindrome.

(b) $P = (u_1 k\tilde{u}_1 p)^{n_1} u_1 \vdots k\tilde{u}_1 \vdots u_3 \vdots \widehat{u}_1\overline{p} | (\overline{u}_1\overline{k}\widehat{u}_1\overline{p})^{n_1}\overline{u}_1 \vdots \overline{k}\widehat{u}_1 \vdots \overline{u}_3 \vdots \tilde{u}_1 p$, with $k, p \in \Sigma^+$ and palindrome.

(c) $P = u_1 \vdots (\tilde{u}_3 u_1)^{n_2} k\tilde{u}_1 \vdots u_3 \vdots \widehat{u}_1\overline{p} | \overline{u}_1 \vdots (\widehat{u}_3\overline{u}_1)^{n_2}\overline{k}\widehat{u}_1 \vdots \overline{u}_3 \vdots \tilde{u}_1 p$, with $k, p \in \Sigma^+$ and p palindrome.

(d) $P = u_1 \vdots k\tilde{u}_1 \vdots (\overline{p}\overline{u}_1 k\tilde{u}_1)^{n_3} u_3 \vdots \widehat{u}_1\overline{p} | \overline{u}_1 \vdots \overline{k}\widehat{u}_1 \vdots (p u_1\overline{k}\widehat{u}_1)^{n_3}\overline{u}_3 \vdots \tilde{u}_1 p$, with $k, p \in \Sigma^+$ and palindrome.

(e) $P = u_1 \vdots k\tilde{u}_1 \vdots u_3 \vdots (\widehat{u}_1 u_3)^{n_4}\widehat{u}_1\overline{p} | \overline{u}_1 \vdots \overline{k}\widehat{u}_1 \vdots \overline{u}_3 \vdots (\tilde{u}_1\overline{u}_3)^{n_4}\tilde{u}_1 p$, with $k, p \in \Sigma^+$ and k palindrome.

(f) $P = (u_1 k\tilde{u}_1 p)^{n_1} u_1 \vdots k\tilde{u}_1 \vdots (\overline{p}\overline{u}_1 k\tilde{u}_1)^{n_3} u_3 \vdots \widehat{u}_1\overline{p} | (\overline{u}_1\overline{k}\widehat{u}_1\overline{p})^{n_1}\overline{u}_1 \vdots \overline{k}\widehat{u}_1 \vdots (p u_1\overline{k}\widehat{u}_1)^{n_3}\overline{u}_3 \vdots \tilde{u}_1 p$, with $k, p \in \Sigma^+$ and palindrome.

(g) $P = u_1 \vdots (\tilde{u}_3 u_1)^{n_2} k\tilde{u}_1 \vdots u_3 \vdots (\widehat{u}_1 u_3)^{n_4}\widehat{u}_1\overline{p} | \overline{u}_1 \vdots (\widehat{u}_3\overline{u}_1)^{n_2}\overline{k}\widehat{u}_1 \vdots \overline{u}_3 \vdots (\tilde{u}_1\overline{u}_3)^{n_4}\tilde{u}_1 p$, with k and p generic non-empty words.

Proof. We proceed case by case, according to the values of n_1, n_2, n_3 and n_4. In each case, the word P is obtained from Eq. (5.3) after replacing $u_2 = k\tilde{u}_1$ and $u_4 = \widehat{u}_1\overline{p}$. We proceed with the proof of the properties of the words k and p in the different situations. At first, we underline that $k, p \neq \varepsilon$ always holds, otherwise we go in contradiction with Theorem 5.2.4.

(a) $n_1 = n_2 = n_3 = n_4 = 0$,

$$P = u_1 \vdots k\tilde{u}_1 \vdots u_3 \vdots \widehat{u}_1\overline{p}|\overline{u}_1 \vdots \overline{k}\widehat{u}_1 \vdots \overline{u}_3 \vdots \tilde{u}_1 p.$$

Since $w_1w_2 = u_1k\tilde{u}_1$ and $w_4\overline{w}_1 = \widehat{u}_1\overline{p}\overline{u}_1$ are palindrome (Theorem 5.1.4), both k and p must be palindrome.

(b) $n_1 > 0$, $n_2 = n_3 = n_4 = 0$,

$$P = (u_1k\tilde{u}_1p)^{n_1}u_1 \vdots k\tilde{u}_1 \vdots u_3 \vdots \widehat{u}_1\overline{p}|(\overline{u}_1\overline{k}\widehat{u}_1\overline{p})^{n_1}\overline{u}_1 \vdots \overline{k}\widehat{u}_1 \vdots \overline{u}_3 \vdots \tilde{u}_1 p.$$

As in case **(a)**, k and p are palindrome from the palindromicity of w_1w_2 and $w_4\overline{w}_1$. In particular, if n_1 is odd we deduce that p is palindrome looking at w_1w_2, and that k is palindrome looking at $w_4\overline{w}_1$. Vice versa if n_1 is even.

(c) $n_2 > 0$, $n_1 = n_3 = n_4 = 0$,

$$P = u_1 \vdots (\tilde{u}_3u_1)^{n_2}k\tilde{u}_1 \vdots u_3 \vdots \widehat{u}_1\overline{p}|\overline{u}_1 \vdots (\widehat{u}_3\overline{u}_1)^{n_2}\overline{k}\widehat{u}_1 \vdots \overline{u}_3 \vdots \tilde{u}_1 p.$$

As in case **(b)**, p is palindrome since the BN-factor $w_4\overline{w}_1$ is palindrome. We cannot say anything about k.

(d) $n_3 > 0$, $n_1 = n_2 = n_4 = 0$,

$$P = u_1 \vdots k\tilde{u}_1 \vdots (\overline{p}\overline{u}_1k\tilde{u}_1)^{n_3}u_3 \vdots \widehat{u}_1\overline{p}|\overline{u}_1 \vdots \overline{k}\widehat{u}_1 \vdots (pu_1\overline{k}\widehat{u}_1)^{n_3}\overline{u}_3 \vdots \tilde{u}_1 p.$$

Similarly to case **(a)**, k is palindrome from w_1w_2 palindrome, while p is palindrome from $w_4\overline{w}_1$ palindrome.

(e) $n_4 > 0$, $n_1 = n_2 = n_3 = 0$,

$$P = u_1 \vdots k\tilde{u}_1 \vdots u_3 \vdots (\widehat{u}_1u_3)^{n_4}\widehat{u}_1\overline{p}|\overline{u}_1 \vdots \overline{k}\widehat{u}_1 \vdots \overline{u}_3 \vdots (\tilde{u}_1\overline{u}_3)^{n_4}\tilde{u}_1 p.$$

The word k is palindrome from the palindromicity of the BN-factor w_1w_2, while nothing can be stated about p.

(f) $n_1, n_3 > 0$, $n_2 = n_4 = 0$,

$$P = (u_1k\tilde{u}_1p)^{n_1}u_1 \vdots k\tilde{u}_1 \vdots (\overline{p}\overline{u}_1k\tilde{u}_1)^{n_3}u_3 \vdots \widehat{u}_1\overline{p}|(\overline{u}_1\overline{k}\widehat{u}_1\overline{p})^{n_1}\overline{u}_1 \vdots \overline{k}\widehat{u}_1 \vdots (pu_1\overline{k}\widehat{u}_1)^{n_3}\overline{u}_3 \vdots \tilde{u}_1 p.$$

We look again at the palindrome BN-factors w_1w_2 and $w_4\overline{w}_1$. From each of them we deduce that k or p is palindrome, depending on the parity of the exponent n_1.

(g) $n_2, n_4 > 0$, $n_1 = n_3 = 0$,

$$P = u_1 \vdots (\tilde{u}_3 u_1)^{n_2} k \tilde{u}_1 \vdots u_3 \vdots (\widehat{u}_1 u_3)^{n_4} \widehat{u}_1 \overline{p} | \overline{u}_1 \vdots (\widehat{u}_3 \overline{u}_1)^{n_2} \overline{k} \widehat{u}_1 \vdots \overline{u}_3 \vdots (\tilde{u}_1 \overline{u}_3)^{n_4} \tilde{u}_1 p.$$

We cannot say anything about k and p, since we do not know the mutual lengths of k, p and u_3.

□

Remark 5.2.6. Actually, we can consider the cases **(f)** and **(g)** only, on varying $n_i \geq 0$ for $i = 1, \ldots, 4$ and according to Lemma 5.1.6. Indeed, all the other cases are included in these last two.

In our recent work [11], we proved that form **(g)** is superfluous too, since any boundary word of a prime double square factorized as in **(g)** can be rephrased in form **(f)**. Hereafter the result.

Theorem 5.2.7 ([11])**.** Let P be the boundary word of a prime double square factorized as **(g)** of Theorem 5.2.5. Then, it is always possible to factorize P as in case **(f)**.

Proof. If $n_2 = n_4 = 0$, the proof is trivial. We show that in the generic case it is always possible to move from **(g)** to **(f)** with a proper choice of new factors k' and p'.
We first assume $n_2 \geq 2$ and consider the BN-factor $X = w_2 w_3 = (\tilde{u}_3 u_1)^{n_2} k \tilde{u}_1 u_3$, that is palindrome by Theorem 5.1.4. As a consequence, $(\tilde{u}_3 u_1)^{n_2 - 1} k$ is palindrome too, and this holds if and only if one of the following cases occurs:

i) $k = u_3$. In this case, w_2 can be written as in **(f)** by choosing $k' = (k u_1)^{n_2} k$. By construction, being $|k| = |u_3|$, and $(\tilde{u}_3 u_1)^{n_2 - 1} k$ and k palindrome, we have that u_1 is palindrome too. As a consequence, k' is palindrome, as required by Theorem 5.2.5.

ii) $\tilde{u}_3 = k \tilde{v}$, for some v suffix of $u_1 = uv$. So, w_2 is written as in **(f)** by choosing $k' = (k \tilde{v} u v)^{n_2} k$. Even in this case, k' is palindrome by construction, after showing, following the same argument used in *i)*, that u is palindrome for any choice of n_2.

iii) $k = \tilde{u}_3 v u_3$ for some palindrome $v \in \Sigma^*$. In this case, the BN-factor X is palindrome if and only if $(u_1 \tilde{u}_3)^{n_2 - 1} v$ is palindrome, if and only if $v = (u_1 \tilde{u}_3)^{\alpha} u_1$ for some $\alpha \geq 0$ and with u_1 and u_3 that are palindrome (from Proposition 6 in [20]). So, the choice of $k' = (\tilde{u}_3 u_1)^{n_2} k$, palindrome by construction, leads to the writing of w_2 as in **(f)**.

If $n_2 = 1$, a similar proof holds by considering the BN-factor $A = u_1 \tilde{u}_3 u_1 k \tilde{u}_1$ instead of X, leading again to the writing of w_2 as in **(f)**.
On the other hand, it is possible to find a proper palindrome $p' \in \Sigma^+$ to write w_4 as in **(f)**, by considering $n_4 \geq 2$ or $n_4 = 1$, and exploiting the palindromicity of the BN-factors B and Y. This concludes the proof. □

To conclude, we define the *standard factorization* of the boundary word of a prime double square as

$$P = (u_1 k \tilde{u}_1 p)^{n_1} u_1 \vdots k \tilde{u}_1 \vdots (\overline{p} \overline{u}_1 k \tilde{u}_1)^{n_3} u_3 \vdots \widehat{u}_1 \overline{p} | (\overline{u}_1 \overline{k} \widehat{u}_1 \overline{p})^{n_1} \overline{u}_1 \vdots \overline{k} \widehat{u}_1 \vdots (p u_1 \overline{k} \widehat{u}_1)^{n_3} \overline{u}_3 \vdots \tilde{u}_1 p, \qquad (5.5)$$

where we recall that:

i) $|u_1| \leq |u_3|$;

ii) k and p are non-empty palindromes;

iii) $n_1, n_3 \geq 0$ and vary independently each other;

iv) if $\alpha\alpha \subseteq P$ for some $\alpha \in \Sigma$, then $\alpha\alpha$ is entirely contained in a factor u_1, u_3, k or p.

All the statements follow from Theorems 5.2.5 and 5.2.7, Condition (5.4), and Theorem 5.2.4.

5.3 Proof of the couple free conjecture

The standard factorization we defined in (5.5) is fundamental to study the couple free conjecture, since it allows to analyze the boundary word of a generic prime double square in terms of four factors only, $u_1, u_3, k, p \in \Sigma^+$. These factors have interesting peculiarities concerning consecutive equal letters. Indeed, if an element of type $\alpha\alpha$ is in P, then we know that it is entirely contained in one of the factors above, and this property considerably simplifies the analysis of the boundary word.
In this section, we propose a proof to the couple free conjecture that proceeds by contradiction: we suppose that two (or more) consecutive equal letters are in the boundary word P, factorized as in (5.5), considering all the possible positions, and then we show that in each case a contradiction is reached.
We first recall a technical lemma useful in the sequel.

Lemma 5.3.1 ([6]). Given $a, b, u \in \Sigma^+$, if $b\tilde{u}a$ and $b\widehat{u}a$ are both palindrome, then $b = \tilde{a}$ and u is palindrome. The same result holds when $\tilde{a}u\tilde{b}$ and $\overline{b}\tilde{u}\overline{a}$ are both palindrome.

The reader can find the proof in [6].

5.3.1 Couple free factors

We consider a factor of type $\alpha\alpha$, with α a letter of the alphabet Σ, and its position in the factors u_1, u_3, k and p of the boundary word P in its standard factorization. We recall that k and p are both palindrome.

Remark 5.3.2. In all our proofs, we consider P the boundary word of a prime double square factorized as in (5.5), and assume $n_1 = n_3 = 0$. We discuss how to generalize to the cases $n_1, n_3 > 0$ at the end of the section.

Lemma 5.3.3 ([6]). Let be $\alpha \in \Sigma$. If $\alpha\alpha \subseteq u_1$, and u_3, k and p are couple free, then P is not a prime double square.

Proof. Let us suppose that there is only one occurrence of consecutive equal letters in P, i.e., $u_1 = a\alpha\alpha b$ for some $a, b \in \Sigma^*$, both couple free. Replacing in (5.5), the boundary word turns out to be

$$P = a\alpha\alpha b \vdots k\tilde{b}\alpha\alpha\tilde{a} \vdots u_3 \vdots \widehat{b\overline{\alpha\alpha}\widehat{a}p} | \overline{a\alpha\alpha}\overline{b} \vdots \widehat{\overline{k}\overline{b}\overline{\alpha\alpha}\widehat{a}} \vdots \overline{u}_3 \vdots \tilde{b}\alpha\alpha\tilde{a}p.$$

As P is prime, by Theorem 5.1.4 all its BN-factors are palindromes, in particular $X = w_2w_3 = k\tilde{b}\alpha\alpha\tilde{a}u_3$ and $B = w_3w_4 = u_3\widehat{b}\overline{\alpha\alpha\widehat{a}p}$. Since no consecutive equal letters occur in a, b, u_3, k and p by hypothesis, we get $k\tilde{b} = \tilde{u}_3a$ (from X palindrome) and $u_3\widehat{b} = \overline{pa}$, since B is palindrome. Let us analyze the mutual lengths of k and u_3 in the first equality:

If $|k| < |u_3|$, there exists $y \in \Sigma^+$ such that $u_3 = yk$, and so $b = \tilde{a}y$ and $yk\widehat{ya} = \overline{pa}$. As a consequence, $p = \overline{y}\overline{k}\tilde{y}$ is palindrome if and only if $y = \varepsilon$, contradiction.

If $|u_3| < |k|$, there exists $k' \in \Sigma^+$ such that $k = \tilde{u}_3k'$, and so $a = k'\tilde{b}$ and $u_3\widehat{b} = \overline{p}\overline{k'}\widehat{b}$, that is, $u_3 = \overline{p}\overline{k'}$. Replacing in k, we have that $k = \widehat{k'}\overline{p}k'$ is palindrome if and only if $k' = \varepsilon$, contradiction.

We conclude that u_3 and k have the same length and so, replacing in the equalities we are studying, $u_3 = k = \overline{p}$ (palindrome) and $b = \tilde{a}$. So, the boundary word becomes

$$P = a\alpha\alpha\tilde{a} \vdots u_3a\alpha\alpha\tilde{a} \vdots u_3 \vdots \overline{\alpha\alpha\widehat{a}}u_3|\overline{\alpha\alpha\widehat{a}} \vdots \overline{u}_3\overline{\alpha\alpha\widehat{a}} \vdots \overline{u}_3 \vdots a\alpha\alpha\tilde{a}\overline{u}_3.$$

Let us now define the morphism $\varphi(0) = u_3$, $\varphi(1) = a\alpha\alpha\tilde{a} = u_1$. Since u_1 and u_3 are both palindrome, φ preserves the operation $\widehat{\cdot}$, i.e., it is homologous. Moreover, if we consider the boundary word of the cross, $Q = 1010\overline{1}0\overline{1010}1\overline{0}$, it holds $P = \varphi(Q)$, and so P is not prime, contradiction.
To conclude the proof, we have to consider the case in which more occurrences of consecutive equal letters are in u_1. So, let us make explicit the first and last one, that is, $u_1 = a\alpha\alpha c\beta\beta b$, with $\alpha, \beta \in \Sigma$, $a, b, c \in \Sigma^*$, and a, b couple free. In this case, the boundary word of the polyomino is

$$P = a\alpha\alpha c\beta\beta b \vdots k\tilde{b}\beta\beta\tilde{c}\alpha\alpha\tilde{a} \vdots u_3 \vdots \widehat{b\beta\beta}\widehat{c\alpha\alpha\widehat{a}p}|\overline{\alpha\alpha c}\overline{\beta\beta b} \vdots \widehat{k\widehat{b\beta\beta}}\widehat{c\alpha\alpha\widehat{a}} \vdots \overline{u}_3 \vdots \tilde{b}\beta\beta\tilde{c}\alpha\alpha\tilde{a}p.$$

According to Theorem 5.1.4, $X = w_2w_3 = k\tilde{b}\beta\beta\tilde{c}\alpha\alpha\tilde{a}u_3$ is palindrome and, being a, b, k and u_3 couple free, $\alpha = \beta$ immediately follows, as well as $u_3 = k = \overline{p}$ and $b = \tilde{a}$ (using the same argument of the previous case). Moreover, c is palindrome too.
As a consequence, the boundary word becomes

$$P = a\alpha\alpha c\alpha\alpha\tilde{a} \vdots u_3a\alpha\alpha c\alpha\alpha\tilde{a} \vdots u_3 \vdots \overline{\alpha\alpha c\alpha\alpha\widehat{a}}u_3|\overline{\alpha\alpha c\alpha\alpha\widehat{a}} \vdots \overline{u}_3\overline{\alpha\alpha c\alpha\alpha\widehat{a}} \vdots \overline{u}_3 \vdots a\alpha\alpha c\alpha\alpha\tilde{a}\overline{u}_3,$$

with u_1 and u_3 both palindrome. So, we can define again the homologous morphism φ that maps the cross in P, reaching a contradiction with the hypothesis of prime double square. This concludes the proof. □

Remark 5.3.4. We underline that, when considering more occurrences of consecutive equal letters in u_1, the proof of Lemma 5.3.3 proceeds analogously to the case of one occurrence only. This is true in general, so in the following proofs we will always consider the case of one occurrence of type $\alpha\alpha$ only, to simplify the calculations.

Lemma 5.3.5 ([6]). Let be $\alpha \in \Sigma$. Then P is not a prime double square when any of the following conditions holds:

i) u_1 is couple free and $\alpha\alpha \subseteq p$, or

ii) u_1 is couple free and $\alpha\alpha \subseteq k$, or

iii) u_1 is couple free and $\alpha\alpha \subseteq u_3$.

Proof. We proceed case by case.

i) The factor p can be written as $p = a\alpha\alpha\tilde{a}$, for some couple free $a \in \Sigma^*$. By Theorem 5.1.4, the BN-factor $B = w_3w_4 = u_3\widehat{u}_1\overline{a\alpha\alpha}\widehat{a}$ is palindrome and so, being u_1 and a couple free, it follows that $u_3 = \overline{a\alpha\alpha}\tilde{b}$ for some $b, c \in \Sigma^*$ such that $\overline{a} = cb$. Then

$$P = u_1 \vdots k\tilde{u}_1 \vdots cb\overline{\alpha\alpha}\tilde{b} \vdots \widehat{u}_1 cb\overline{\alpha\alpha}\tilde{b}\tilde{c}|\overline{u}_1 \vdots \overline{k}\widehat{u}_1 \vdots \overline{c}\overline{b}\alpha\alpha\widehat{b} \vdots \tilde{u}_1\overline{c}\overline{b}\alpha\alpha\widehat{b}\widehat{c}.$$

The BN-factor $X = w_2w_3 = k\tilde{u}_1cb\overline{\alpha\alpha}\tilde{b}$ is palindrome and so, being k palindrome as well, $k = b\overline{\alpha\alpha}\tilde{b}$ holds. We replace in the boundary word, and we get

$$P = u_1 \vdots b\overline{\alpha\alpha}\tilde{b}\tilde{u}_1 \vdots cb\overline{\alpha\alpha}\tilde{b} \vdots \widehat{u}_1 cb\overline{\alpha\alpha}\tilde{b}\tilde{c}|\overline{u}_1 \vdots \overline{b}\alpha\alpha\widehat{b}\widehat{u}_1 \vdots \overline{c}\overline{b}\alpha\alpha\widehat{b} \vdots \tilde{u}_1\overline{c}\overline{b}\alpha\alpha\widehat{b}\widehat{c}.$$

Now, again by the palindromicity of the BN-factors w_2w_3 and w_3w_4, we observe that $\tilde{u}_1c$ and $\widehat{u}_1c$ are both palindrome, if and only if $c = \varepsilon$ and $u_1 = \tilde{u}_1$.

We finally reach

$$P = u_1 \vdots b\overline{\alpha\alpha}\tilde{b}u_1 \vdots b\overline{\alpha\alpha}\tilde{b} \vdots \overline{u}_1 b\overline{\alpha\alpha}\tilde{b}|\overline{u}_1 \vdots \overline{b}\alpha\alpha\widehat{b}\overline{u}_1 \vdots \overline{b}\alpha\alpha\widehat{b} \vdots u_1\overline{b}\alpha\alpha\widehat{b},$$

that can be obtained as $P = \varphi(Q)$, with Q the boundary word of the cross and φ the homologous morphism defined as $\varphi(0) = b\overline{\alpha\alpha}\tilde{b} = u_3$, $\varphi(1) = u_1$. Then P is not prime, contradiction.

ii) The factor k can be written as $k = a\alpha\alpha\tilde{a}$, for some couple free $a \in \Sigma^*$, and the boundary word of the polyomino is

$$P = u_1 \vdots a\alpha\alpha\tilde{a}\tilde{u}_1 \vdots u_3 \vdots \widehat{u}_1\overline{p}|\overline{u}_1 \vdots \overline{a\alpha\alpha}\widehat{a}\widehat{u}_1 \vdots \overline{u}_3 \vdots \tilde{u}_1 p.$$

Using the same argument of case *i)*, considering the palindrome w_2w_3, we argue $u_3 = b\alpha\alpha\tilde{a}$ and $a = cb$, for some $b, c \in \Sigma^*$, and so

$$P = u_1 \vdots cb\alpha\alpha\tilde{b}\tilde{c}\tilde{u}_1 \vdots b\alpha\alpha\tilde{b}\tilde{c} \vdots \widehat{u}_1 b\alpha\alpha\tilde{b}|\overline{u}_1 \vdots \overline{c}\overline{b}\overline{\alpha\alpha}\widehat{b}\widehat{c}\widehat{u}_1 \vdots \overline{b}\overline{\alpha\alpha}\widehat{b}\widehat{c} \vdots \tilde{u}_1\overline{b}\overline{\alpha\alpha}\widehat{b}.$$

Being w_2w_3 and w_3w_4 palindrome, the words $\tilde{c}\tilde{u}_1$ and $\tilde{c}\widehat{u}_1$ are palindrome at the same time, if and only if $c = \varepsilon$ and $u_1 = \tilde{u}_1$.

We finally reach

$$P = u_1 \vdots b\alpha\alpha\tilde{b}u_1 \vdots b\alpha\alpha\tilde{b} \vdots \overline{u}_1 b\alpha\alpha\tilde{b}|\overline{u}_1 \vdots \overline{b}\overline{\alpha\alpha}\widehat{b}\overline{u}_1 \vdots \overline{b}\overline{\alpha\alpha}\widehat{b} \vdots u_1\overline{b}\overline{\alpha\alpha}\widehat{b},$$

that can be obtained as $P = \varphi(Q)$, with Q the boundary word of the cross and φ the homologous morphism defined as $\varphi(0) = b\alpha\alpha\tilde{b} = u_3$, $\varphi(1) = u_1$. Then P is not prime, contradiction.

iii) We replace in the boundary word the factor $u_3 = a\alpha\alpha b$, with $a, b \in \Sigma^*$ both couple free,

$$P = u_1 \vdots k\tilde{u}_1 \vdots a\alpha\alpha b \vdots \widehat{u}_1\overline{p}|\overline{u}_1 \vdots \overline{k}\widehat{u}_1 \vdots \overline{a\alpha\alpha}\overline{b} \vdots \tilde{u}_1 p.$$

We proceed as in the previous cases: since w_2w_3 and w_3w_4 are palindromes, we deduce $k = \tilde{b}\alpha\alpha b$ and $\overline{p} = a\alpha\alpha\tilde{a}$, respectively. Replacing in P, the boundary word turns out to be

$$P = u_1 \vdots \tilde{b}\alpha\alpha b\tilde{u}_1 \vdots a\alpha\alpha b \vdots \widehat{u}_1 a\alpha\alpha\tilde{a}|\overline{u}_1 \vdots \widehat{b\overline{\alpha\alpha}\overline{b}}\widehat{u}_1 \vdots \overline{a\alpha\alpha}\overline{b} \vdots \tilde{u}_1\overline{a\alpha\alpha}\widehat{a}.$$

Again from the palindromes w_2w_3 and w_3w_4, we observe that $b\tilde{u}_1a$ and $b\widehat{u}_1a$ are palindrome at the same time. So, by Lemma 5.3.1, $u_1 = \tilde{u}_1$ and $b = \tilde{a}$ hold.

The boundary word is now

$$P = u_1 \vdots a\alpha\alpha\tilde{a}u_1 \vdots a\alpha\alpha\tilde{a} \vdots \overline{u}_1 a\alpha\alpha\tilde{a}|\overline{u}_1 \vdots \overline{a\alpha\alpha\widehat{a}\overline{u}_1} \vdots \overline{a\alpha\alpha\widehat{a}} \vdots u_1\overline{a\alpha\alpha\widehat{a}},$$

and can be obtained as $P = \varphi(Q)$, with Q the boundary word of the cross and φ the homologous morphism defined as $\varphi(0) = a\alpha\alpha\tilde{a} = u_3$, $\varphi(1) = u_1$. Then P is not prime, contradiction.

We reached a contradiction for all the cases, and this concludes the proof. □

Lemma 5.3.6 ([6]). *P is not a prime double square when any of the following conditions is valid:*

i) u_1 *and* p *are not couple free, while* u_3 *and* k *are, or*

ii) u_1 *and* k *are not couple free, while* u_3 *and* p *are, or*

iii) u_1, k *and* p *are not couple free, while* u_3 *is.*

Proof. We proceed case by case.

i) Let $u_1 = a\alpha\alpha b$ and $p = c\beta\beta\tilde{c}$, with $\alpha, \beta \in \Sigma$ and $a, b, c \in \Sigma^*$ couple free. When considering the BN-factors $X = w_2w_3 = k\tilde{b}\alpha\alpha\tilde{a}u_3$ and $B = u_3\widehat{b\overline{\alpha\alpha}\widehat{a}\overline{c}\overline{\beta\beta}\widehat{c}}$, both palindrome by Theorem 5.1.4, we argue $\alpha = \beta$, $k\tilde{b} = \tilde{u}_3a$ and $\widehat{c} = \overline{b}\tilde{u}_3$.

The boundary word turns out to be

$$P = a\alpha\alpha b \vdots \tilde{u}_3 a\alpha\alpha\tilde{a} \vdots u_3 \vdots \widehat{b\overline{\alpha\alpha}\widehat{a}}u_3\widehat{b\overline{\alpha\alpha}\widehat{a}}\tilde{u}_3|\overline{a\alpha\alpha}\overline{b} \vdots \widehat{u}_3\overline{a\alpha\alpha\widehat{a}} \vdots \overline{u}_3 \vdots \tilde{b}\alpha\alpha\tilde{a}\overline{u}_3\tilde{b}\alpha\alpha\tilde{a}\widehat{u}_3$$

and, by the palindrome BN-factors $A = w_1w_2$ and $B = w_3w_4$, we deduce that $b\tilde{u}_3a$ and $\widehat{a}u_3\widehat{b}$, respectively, are palindrome too. As a consequence of Lemma 5.3.1, $u_3 = \tilde{u}_3$ and $b = \tilde{a}$, and so

$$P = a\alpha\alpha\tilde{a} \vdots u_3 a\alpha\alpha\tilde{a} \vdots u_3 \vdots \overline{a\alpha\alpha\widehat{a}}u_3\overline{a\alpha\alpha\widehat{a}}u_3|\overline{a\alpha\alpha\widehat{a}} \vdots \overline{u}_3\overline{a\alpha\alpha\widehat{a}} \vdots \overline{u}_3 \vdots a\alpha\alpha\tilde{a}\overline{u}_3 a\alpha\alpha\tilde{a}\overline{u}_3.$$

The boundary word can be obtained as $P = \varphi(Q)$, with $Q = 1010\overline{1}0\overline{1}0\overline{1010}1\overline{0}1\overline{0}$ and $\varphi(0) = u_3$, $\varphi(1) = a\alpha\alpha\tilde{a} = u_1$, that is an homologous morphism since u_1 and u_3 are both palindrome. Then P is not prime, contradiction.

ii) Let $u_1 = a\alpha\alpha b$ and $k = c\beta\beta\tilde{c}$, with $\alpha, \beta \in \Sigma$ and $a, b, c \in \Sigma^*$ couple free. Since the BN-factors $X = w_2w_3 = c\beta\beta\tilde{c}\tilde{b}\alpha\alpha\tilde{a}u_3$ and $B = w_3w_4 = u_3\widehat{b\overline{\alpha\alpha}\widehat{a}\overline{p}}$ are palindrome by Theorem 5.1.4, then $\alpha = \beta$, $c = \tilde{u}_3a$ and $\widehat{a}\overline{p} = \overline{b}\tilde{u}_3$ hold.

The boundary word turns out to be

$$P = a\alpha\alpha b \vdots \tilde{u}_3a\alpha\alpha\tilde{a}u_3\tilde{b}\alpha\alpha\tilde{a} \vdots u_3 \vdots \widehat{b\overline{\alpha\alpha}\overline{b}\tilde{u}_3}|\overline{a\alpha\alpha}\overline{b} \vdots \widehat{u}_3\overline{a\alpha\alpha\widehat{a}}\overline{u}_3\widehat{b\overline{\alpha\alpha}\widehat{a}} \vdots \overline{u}_3 \vdots \tilde{b}\alpha\alpha b\widehat{u}_3$$

and, by the palindrome BN-factors $X = w_2w_3$ and $Y = w_4\overline{w}_1$, we deduce, respectively, that $\tilde{a}u_3\tilde{b}$ and $\overline{b}\tilde{u}_3\overline{a}$ are palindrome at the same time. By Lemma 5.3.1, $b = \tilde{a}$ and $u_3 = \tilde{u}_3$, and so

$$P = a\alpha\alpha\tilde{a} \vdots u_3a\alpha\alpha\tilde{a}u_3a\alpha\alpha\tilde{a} \vdots u_3 \vdots \overline{a\alpha\alpha\widehat{a}}u_3|\overline{a\alpha\alpha\widehat{a}} \vdots \overline{u}_3\overline{a\alpha\alpha\widehat{a}}\overline{u}_3\overline{a\alpha\alpha\widehat{a}} \vdots \overline{u}_3 \vdots a\alpha\alpha\tilde{a}\overline{u}_3.$$

As in the previous case, the boundary word can be obtained as $P = \varphi(Q)$, with $Q = 101010\overline{1}0\overline{1010101}\overline{1}\overline{0}$ and $\varphi(0) = u_3$, $\varphi(1) = a\alpha\alpha\tilde{a} = u_1$, that is an homologous morphism since u_1 and u_3 are both palindrome. Then P is not prime, contradiction.

iii) Let $u_1 = a\alpha\alpha b$, $k = c\beta\beta\tilde{c}$ and $p = d\gamma\gamma\tilde{d}$, with $\alpha, \beta, \gamma \in \Sigma$ and $a, b, c, d \in \Sigma^*$ couple free. Replacing in the boundary word, we obtain

$$P = a\alpha\alpha b \vdots c\beta\beta\tilde{c}\tilde{b}\alpha\alpha\tilde{a} \vdots u_3 \vdots \widehat{b\overline{\alpha\alpha}\widehat{a}}\overline{d}\overline{\gamma\gamma}\widehat{d}|\overline{a\alpha\alpha}\overline{b} \vdots \overline{c\beta\beta}\widehat{c}\widehat{b\overline{\alpha\alpha}\widehat{a}} \vdots \overline{u}_3 \vdots \tilde{b}\alpha\alpha\tilde{a}d\gamma\gamma\tilde{d}.$$

The proof proceeds as in the previous cases: first, from the palindrome BN-factors $X = w_2w_3$ and $B = w_3w_4$, we see that $\alpha = \beta = \gamma$, $c = \tilde{u}_3a$ and $\widehat{d} = \overline{b}\tilde{u}_3$. Then, exploiting Lemma 5.3.1, we get $b = \tilde{a}$ and $u_3 = \tilde{u}_3$, and so

$$P = a\alpha\alpha\tilde{a} \vdots u_3a\alpha\alpha\tilde{a}u_3a\alpha\alpha\tilde{a} \vdots u_3 \vdots \overline{a\alpha\alpha\widehat{a}}u_3\overline{a\alpha\alpha\widehat{a}}u_3|\overline{a\alpha\alpha\widehat{a}} \vdots \overline{u}_3\overline{a\alpha\alpha\widehat{a}}\overline{u}_3\overline{a\alpha\alpha\widehat{a}} \vdots \overline{u}_3 \vdots a\alpha\alpha\tilde{a}\overline{u}_3a\alpha\alpha\tilde{a}\overline{u}_3.$$

The boundary word can be obtained as $P = \varphi(Q)$, with $Q = 101010\overline{1}0\overline{1}0\overline{1010101}0\overline{1}0\overline{1}0$ and $\varphi(0) = u_3$, $\varphi(1) = a\alpha\alpha\tilde{a} = u_1$, that is an homologous morphism since u_1 and u_3 are both palindrome. Then P is not prime, contradiction.

We reached a contradiction for all the cases, and this concludes the proof. □

Lemma 5.3.7 ([6]). If u_1 and u_3 are not couple free, while k and p are, then P is not a prime double square.

Proof. We write $u_1 = a\alpha\alpha b$ and $u_3 = c\beta\beta d$, with $\alpha, \beta \in \Sigma$ and $a, b, c, d \in \Sigma^*$ couple free, and then replace them in the boundary word:

$$P = a\alpha\alpha b \vdots k\tilde{b}\alpha\alpha\tilde{a} \vdots c\beta\beta d \vdots \widehat{b\overline{\alpha\alpha}\widehat{a}\overline{p}}|\overline{a\alpha\alpha}\overline{b} \vdots \overline{k}\widehat{b\overline{\alpha\alpha}\widehat{a}} \vdots \overline{c\beta\beta d} \vdots \tilde{b}\alpha\alpha\tilde{a}p.$$

By Theorem 5.1.4, $X = w_2w_3 = k\tilde{b}\alpha\alpha\tilde{a}c\beta\beta d$ and $B = w_3w_4 = c\beta\beta d\widehat{b\overline{\alpha\alpha}\widehat{a}\overline{p}}$ are palindrome and so, being a, b, c, d and k, p couple free by hypothesis, $\beta = \alpha$ and $\beta = \overline{\alpha}$ hold at the same time, and this is a contradiction. □

Lemma 5.3.8 ([6]). The following statements hold:

i) if u_1, u_3 and k are not couple free, while p is, then P is not a prime double square.

ii) if u_1, u_3 and p are not couple free, while k is, then P is not a prime double square.

Proof. We study the two cases separately.

i) Let be $u_1 = a\alpha\alpha b$, $u_3 = c\beta\beta d$ and $k = e\lambda\lambda\tilde{e}$, with $\alpha, \beta, \lambda \in \Sigma$ and $a, b, c, d, e \in \Sigma^*$ couple free. By Theorem 5.1.4, the BN-factors $X = w_2w_3 = e\lambda\lambda\tilde{e}\tilde{b}\alpha\alpha\tilde{a}c\beta\beta d$ and $B = w_3w_4 = c\beta\beta d\widehat{b}\overline{\alpha\alpha}\widehat{a}\overline{p}$ are palindrome, so that $\lambda = \beta = \overline{\alpha}$, $e = \tilde{d}$, $c = \overline{pa}$ and $\tilde{e}\tilde{b} = \tilde{c}a$ hold. Moreover, $d\widehat{b}$ is palindrome too.

Replacing in the boundary word,

$$P = a\alpha\alpha b \vdots \tilde{d}\overline{\alpha\alpha}d\tilde{b}\alpha\alpha\tilde{a} \vdots \overline{pa\alpha\alpha}d \vdots \widehat{b}\overline{\alpha\alpha}\widehat{a}\overline{p}|\overline{a\alpha\alpha}\overline{b} \vdots \widehat{d}\alpha\alpha\overline{d}\widehat{b}\overline{\alpha\alpha}\widehat{a} \vdots pa\alpha\alpha\overline{d} \vdots \tilde{b}\alpha\alpha\tilde{a}p.$$

Similarly, $d\tilde{b} = \widehat{a}\overline{p}a$ from the palindrome w_2w_3, and so

$$P = a\alpha\alpha\tilde{a}\overline{pa\alpha\alpha}\widehat{a}\overline{p}a\alpha\alpha\tilde{a}\overline{pa\alpha\alpha}d\widehat{b}\overline{\alpha\alpha}\widehat{a}\overline{p}|\overline{a\alpha\alpha}\widehat{a}pa\alpha\alpha\tilde{a}p\overline{a\alpha\alpha}\widehat{a}pa\alpha\alpha\overline{d}\tilde{b}\alpha\alpha\tilde{a}p,$$

where we recall that $d\widehat{b}$ is palindrome and $d\tilde{b} = \widehat{a}\overline{p}a$.

In P, the factor $v = \overline{pa\alpha\alpha}\widehat{a}pa\alpha\alpha\tilde{a}$ occurs, with $|v|_0 = |v|_{\overline{0}}$ and $|v|_1 = |v|_{\overline{1}}$. So, the boundary of the polyomino self-intersects, contradiction.

ii) Let be $u_1 = a\alpha\alpha b$, $u_3 = c\beta\beta d$ and $p = e\gamma\gamma\tilde{e}$, with $\alpha, \beta, \gamma \in \Sigma$ and $a, b, c, d, e \in \Sigma^*$ couple free. The BN-factors $X = w_2w_3 = k\tilde{b}\alpha\alpha\tilde{a}c\beta\beta d$ and $B = w_3w_4 = c\beta\beta d\widehat{b}\overline{\alpha\alpha}\widehat{a}e\overline{\gamma\gamma}\widehat{e}$ are palindrome, so that $\alpha = \beta = \overline{\gamma}$, $d = bk$, $d\widehat{b} = \widehat{e}\overline{a}$ and $\overline{e} = c$ hold. Moreover, $\tilde{a}c = \tilde{c}a$ is palindrome.

The boundary word of the polyomino turns out to be

$$P = a\alpha\alpha b \vdots k\tilde{b}\alpha\alpha\tilde{a} \vdots c\alpha\alpha bk \vdots \widehat{b}\overline{\alpha\alpha}\widehat{a}c\alpha\alpha\tilde{c}|\overline{a\alpha\alpha}\overline{b} \vdots \overline{k}\widehat{b}\overline{\alpha\alpha}\widehat{a} \vdots \overline{c\alpha\alpha}\overline{bk} \vdots \tilde{b}\alpha\alpha\tilde{a}\overline{c\alpha\alpha}\widehat{c},$$

with $\tilde{c}a$ palindrome and $\tilde{c}\overline{a} = bk\widehat{b}$, from the palindromicity of w_3w_4.

In P, the factor $v = \widehat{a}c\alpha\alpha\tilde{c}\overline{a\alpha\alpha}\overline{bk} = \overline{b}k\tilde{b}\alpha\alpha bk\widehat{b}\overline{\alpha\alpha}\overline{bk}$ occurs, and it contains the factor $v' = \tilde{b}\alpha\alpha bk\widehat{b}\overline{\alpha\alpha}\overline{bk}$, with $|v'|_0 = |v'|_{\overline{0}}$ and $|v'|_1 = |v'|_{\overline{1}}$. So, the boundary of the polyomino self-intersects, contradiction.

In both cases we reached a contradiction with the definition of polyomino, and this concludes the proof. □

Lemma 5.3.9 ([6]). If u_1, u_3, p and k are not couple free, then P is not a prime double square.

Proof. We write the factors of P as $u_1 = a\alpha\alpha b$, $u_3 = c\beta\beta d$, $p = e\gamma\gamma\tilde{e}$ and $k = f\lambda\lambda\tilde{f}$, with $\alpha, \beta, \gamma, \lambda \in \Sigma$ and $a, b, c, d, e, f \in \Sigma^*$ couple free, and so

$$P = a\alpha\alpha b \vdots f\lambda\lambda\tilde{f}\tilde{b}\alpha\alpha\tilde{a} \vdots c\beta\beta d \vdots \widehat{b\overline{\alpha\alpha}\widehat{a}e\gamma\gamma\widehat{e}}|\overline{a\alpha\alpha}\overline{b} \vdots \overline{f\lambda\lambda}\widehat{f}\widehat{b}\overline{\alpha\alpha}\widehat{a} \vdots \overline{c\beta\beta d} \vdots \tilde{b}\alpha\alpha\tilde{a}e\gamma\gamma\tilde{e}.$$

From Theorem 5.1.4, the BN-factors $X = f\lambda\lambda\tilde{f}\tilde{b}\alpha\alpha\tilde{a}c\beta\beta d$ and $B = c\beta\beta d\widehat{b\overline{\alpha\alpha}\widehat{a}e\gamma\gamma\widehat{e}}$ are palindrome, so that $\beta = \lambda = \overline{\gamma}$, $f = \tilde{d}$, $\overline{e} = c$, $\tilde{f}\tilde{b} = \tilde{c}a$ and $\widehat{a}\overline{e} = \overline{b}\tilde{d}$.
As a consequence, we get

$$P = a\alpha\alpha\tilde{a}c\beta\beta\tilde{c}a\alpha\alpha\tilde{a}c\beta\beta\tilde{c}\overline{a\alpha\alpha\widehat{a}}c\beta\beta\tilde{c}|\overline{a\alpha\alpha\widehat{a}\overline{c}}\overline{\beta\beta}\widehat{c}\overline{a\alpha\alpha\widehat{a}\overline{c}}\overline{\beta\beta}\widehat{c}a\alpha\alpha\tilde{a}\overline{c}\overline{\beta\beta}\widehat{c}.$$

The boundary word can be obtained as $P = \varphi(Q)$, with $Q = 1010\overline{1}0\overline{1010}1\overline{0}$ and $\varphi(0) = c\beta\beta\tilde{c} = u_3$, $\varphi(1) = a\alpha\alpha\tilde{a} = u_1$, that is an homologous morphism since u_1 and u_3 are both palindrome. Then P is not prime, contradiction. □

Remark 5.3.10. We remark that all the proofs given for Lemmas 5.3.3-5.3.9 hold when considering more occurrences of consecutive equal letters in the factors that are not couple free. We address the reader to our work [6], Example 4, for a deeper analysis of such cases.

5.3.2 Main result and generalization

We are now ready to state the main result of this section, that is the proof of the couple free conjecture when $n_1 = n_3 = 0$ holds in the boundary word of a prime double square in its standard factorization, Eq. (5.5).

Theorem 5.3.11 ([6])**.** If P is the boundary word of a prime double square such that $n_1 = n_3 = 0$ in its standard factorization, then P is couple free.

Proof. The proof is a direct consequence of Lemmas 5.3.3-5.3.9, where we reached a contradiction for any possible position of a factor of type $\alpha\alpha$ in the boundary word P. So, by contradiction, let us assume that the boundary word P of a prime double square contains two consecutive equal letters. By Lemma 5.3.5, we know that u_1 cannot be couple free, meaning that $\alpha\alpha \subseteq u_1$ for some $\alpha \in \Sigma$. Let us now consider the factor u_3: from Lemmas 5.3.7 and 5.3.8, if u_3 is not couple free, then two consecutive equal letters must occur both in k and p, or in none of them. In the first case, P is not a prime double square, since in Lemmas 5.3.6 and 5.3.9 we have shown how to reach its boundary word through the composition of two non-trivial morphisms. In the latter, we are in the case in which u_1 is not couple free, while u_3, k and p are all, that is not possible by Lemma 5.3.3. We conclude that $\alpha\alpha$ is not a factor of P for any symbol $\alpha \in \Sigma$, as claimed. □

Generalization to the case of non-null exponents

To conclude the proof of the conjecture, we finally have to consider the case in which the exponents n_1, n_3 in (5.5) are strictly greater than zero. The proof follows the same strategy: Lemmas 5.3.3-5.3.9 can be proven assuming $n_1 > 0$ or $n_3 > 0$, or both, thus getting an analogous of Theorem 5.3.11 for the general case.

Since all the proofs follow exactly the same arguments already used, here we only sketch how to generalize them.

The key point we exploited to reach our results is Theorem 5.1.4, that states that all the BN-factors of the boundary word of a prime double square are palindrome. In fact, when we impose this property on A, B, X and Y, we compare the factors u_1, u_3, k and p among them. In all cases, we finally reach a contradiction: we prove that the boundary of the polyomino self-intersects, as in case of Lemma 5.3.8, or that the factors u_1, u_3, k and p can be written as the concatenation of copies of $u_1, \overline{u}_1, u_3$ and $\overline{u}_3$, that are also palindrome, thus allowing to write the trivial morphism φ_P as the composition of more non-trivial homologous morphisms.
The same contradictions are reached when $n_1, n_3 > 0$, since the structure of the involved BN-factors does not change, essentially (see [6] for more details).

Example 5.3.12. As an example, we describe here the proof of Lemma 5.3.3 in case both $n_1, n_3 > 0$.

Claim: Let be $\alpha \in \Sigma$. If $\alpha\alpha \subseteq u_1$, and u_3, k and p are couple free, then P is not a prime double square.

The proof proceeds by contradiction. We consider the general case in which an arbitrary number of occurrences of consecutive equal letters is in u_1, that is, $u_1 = a\alpha\alpha c\beta\beta b$, with $\alpha, \beta \in \Sigma$, $a, b, c \in \Sigma^*$, and a, b couple free (we explicit the first and last pair of consecutive equal letters in u_1). Then, replacing in the boundary word P factorized as in (5.5), we obtain

$$P = (a\alpha\alpha c\beta\beta bk\tilde{b}\beta\beta\tilde{c}\alpha\alpha\tilde{a}p)^{n_1} a\alpha\alpha c\beta\beta b \vdots k\tilde{b}\beta\beta\tilde{c}\alpha\alpha\tilde{a} \vdots (\overline{pa\alpha\alpha c}\overline{\beta\beta b}k\tilde{b}\beta\beta\tilde{c}\alpha\alpha\tilde{a})^{n_3} u_3 \vdots \widehat{b\beta\beta\tilde{c}\alpha\alpha\tilde{a}p}|$$

$$(\overline{a\alpha\alpha c}\overline{\beta\beta bk}\widehat{b\beta\beta}\widehat{\tilde{c}\alpha\alpha\tilde{a}p})^{n_1} \overline{a\alpha\alpha c}\overline{\beta\beta b} \vdots \widehat{kb\beta\beta\tilde{c}\alpha\alpha\tilde{a}} \vdots (pa\alpha\alpha c\beta\beta b\widehat{kb\beta\beta}\widehat{\tilde{c}\alpha\alpha\tilde{a}})^{n_3} \overline{u}_3 \vdots \tilde{b}\beta\beta\tilde{c}\alpha\alpha\tilde{a}p.$$

We consider the BN-factor $X = k\tilde{b}\beta\beta\tilde{c}\alpha\alpha\tilde{a} \vdots (\overline{pa\alpha\alpha c}\overline{\beta\beta b}k\tilde{b}\beta\beta\tilde{c}\alpha\alpha\tilde{a})^{n_3} u_3$, that is palindrome by Theorem 5.1.4. Then, being a, b, k and u_3 all couple free, it must hold $\alpha = \beta$, $c = \tilde{c}$ (palindrome), $bk = \tilde{a}u_3$ and, finally, $\tilde{a}\overline{pa} = bk\widehat{b}$. From the last two equalities, we also obtain that $\overline{pa} = u_3\widehat{b}$.
Now, we replace $bk = \tilde{a}u_3$ and $\overline{pa} = u_3\widehat{b}$ in P, as well as $\alpha = \beta$ and $c = \tilde{c}$, thus getting

$$P = (a\alpha\alpha c\alpha\alpha\tilde{a}u_3\tilde{b}\alpha\alpha c\alpha\alpha b\widehat{u}_3)^{n_1} a\alpha\alpha c\alpha\alpha b \vdots \tilde{u}_3 a\alpha\alpha c\alpha\alpha\tilde{a} \vdots (u_3\widehat{b\overline{\alpha\alpha c\alpha\alpha}\overline{b}}\tilde{u}_3 a\alpha\alpha c\alpha\alpha\tilde{a})^{n_3} u_3 \vdots \widehat{b\overline{\alpha\alpha c\alpha\alpha}\overline{b}}\tilde{u}_3|$$

$$(\overline{a\alpha\alpha c\alpha\alpha\widehat{a}u_3}\widehat{b\overline{\alpha\alpha c\alpha\alpha}\overline{b}}\tilde{u}_3)^{n_1} \overline{a\alpha\alpha c\alpha\alpha\overline{b}} \vdots \widehat{u}_3\overline{a\alpha\alpha c\alpha\alpha\widehat{a}} \vdots (\overline{u}_3\tilde{b}\alpha\alpha c\alpha\alpha b\widehat{u}_3\overline{a\alpha\alpha c\alpha\alpha\widehat{a}})^{n_3} \overline{u}_3 \vdots \tilde{b}\alpha\alpha c\alpha\alpha b\widehat{u}_3.$$

When considering the BN-factor A, we see that it is palindrome if and only if $b\widehat{u}_3 a$ is palindrome too, if n_1 is odd, or if $\tilde{a}u_3\tilde{b}$ is palindrome too, if n_1 is even.
In the first case, we exploit the fact that $bk = \tilde{a}u_3$ implies $\tilde{a}u_3\tilde{b}$ palindrome and so, by Lemma 5.3.1, $b = \tilde{a}$ and $u_3 = \tilde{u}_3$.
In the second case, the palindromicity of $\overline{b}\tilde{u}_3\overline{a}$ is given by the palindrome BN-factor Y, and so the same conclusion can be reached, again using Lemma 5.3.1.

We conclude that $u_1 = a\alpha\alpha c\alpha\alpha\tilde{a}$ and u_3 are both palindrome, and so

$$P = (u_1u_3u_1\overline{u}_3)^{n_1}u_1 \vdots u_3u_1 \vdots (u_3\overline{u}_1u_3u_1)^{n_3}u_3 \vdots \overline{u}_1u_3|(\overline{u}_1\overline{u}_3\overline{u}_1u_3)^{n_1}\overline{u}_1 \vdots \overline{u}_3\overline{u}_1 \vdots (\overline{u}_3u_1\overline{u}_3\overline{u}_1)^{n_3}\overline{u}_3 \vdots u_1\overline{u}_3.$$

The morphism defined as $\varphi(0) = u_3$, $\varphi(1) = u_1$ is homologous and such that $\varphi(Q) = P$, with $Q = (101\overline{0})^{n_1}101(0\overline{1}01)^{n_3}0\overline{1}0(\overline{101}0)^{n_1}\overline{101}(\overline{0}1\overline{01})^{n_3}\overline{0}1\overline{0}$. So P is not prime, in contradiction with the hypothesis, and this concludes the proof.

We conclude by stating that an analogous of Theorem 5.3.11 holds in the general case, that is,

Theorem 5.3.13 ([6]). If P is the boundary word of a prime double square and $\alpha \in \Sigma$, then $\alpha\alpha \not\subseteq P$.

So, the couple free conjecture holds.

5.4 Characterization of prime double squares

So far, we have shown that if P codes the boundary of a tile in this class, then it admits a standard factorization, (5.5), and no consecutive equal letters are present in it. In this section, we deepen the analysis of the factors u_1, u_3, k and p that appear in (5.5). A strong structure and many properties are present in each of them, mainly concerning the alphabet on which they are defined.

We start recalling some technical lemmas that we will use in our proofs.

Lemma 5.4.1 (Lothaire [55]). Given a word $w \in \Sigma^*$, assume that $w = xy = yz$, with $y \neq \varepsilon$. Then, for some non-empty palindromes $a, b \in \Sigma^+$ and some $i \geq 0$, we have $x = ab$, $y = (ab)^i a$ and $z = ba$.

Lemma 5.4.2 (Blondin Massé et al., from Lemma 1 in [19]). Let $x_1, x_2, x_3 \in \Sigma^+$ be three non-empty palindromes such that $x_1x_2x_3$ is palindrome too. Then, for some non-empty palindromes $a, b \in \Sigma^+$, x_1, x_2 and x_3 can be obtained as their alternate concatenations.

5.4.1 The alphabet of factors

Our aim is to characterize the occurrences of the letters of Σ in the factors u_1, u_3, k and p in the standard factorization of P. Indeed, while carrying on the proof of the couple free conjecture, looking at many examples we found out in several cases that not all the symbols of the alphabet Σ appear in all the factors of (5.5). We provide a proof, showing that for each factor there exists a letter having no single occurrence, in fact. As a consequence, combining such a result with all the properties of prime double squares that we know so far, we are able to characterize the factors u_1, u_3, k and p completely.

Remark 5.4.3. Without loss of generality, we always assume $n_1 = n_3 = 0$ in (5.5). As discussed in the previous section, this hypothesis does not alter the proofs of our results, since the structure of the BN-factors, as a concatenation of u_1, u_3, k and p, does not change essentially.

We start with some preliminary results, and then proceed with the analysis of each factor separately.

Lemma 5.4.4 ([7]). Let $P = AB\widehat{A}\widehat{B} \equiv XY\widehat{X}\widehat{Y}$ be the boundary word of a prime double square. For each BN-factorization, the four BN-factors begin (and end) with a different letter of the alphabet $\Sigma = \{1, 0, \overline{1}, \overline{0}\}$.

The result is a direct consequence of Theorems 5.1.4 and 5.3.13.
Without loss of generality, we can choose the starting letters of the BN-factors A and B still keeping condition (5.4), since we consider tiles up to symmetries.
We fix 1 and 0 as the first letter of A and B, respectively. As a consequence, X starts with the letter 0 while Y with $\overline{1}$, since the polyomino is coded traveling its boundary clockwise.

Theorem 5.4.5 ([7]). Given P the boundary word of a prime double square in its standard factorization, (5.5), the factor u_1 begins and ends with the letter 1, while the factors u_3, k and $\overline{p}$ all begin and end with the letter 0.

The proof is a direct consequence of the choices we made for the starting letters of the BN-factors, combined with their palindromicity (Theorem 5.1.4).

The alphabet of u_1

In our work [7], we reached a crucial result about the factor u_1: the letter $\overline{1} \in \Sigma$ cannot appear in u_1 with a single occurrence. We dedicate this section to the proof of this important result, here stated:

Theorem 5.4.6 ([7]). Given P the boundary word of a prime double square in its standard factorization, (5.5), if the factor u_1 contains the letter $\overline{1} \in \Sigma$, then such a letter occurs more than once.

The proof proceeds by contradiction, after showing that the presence of a unique letter $\overline{1}$ in u_1 makes the polyomino P not a prime double square. So, from now on, we write the factor u_1 as $u_1 = 1a\overline{1}b1$, with $\overline{1} \notin a, b$ and $a, b \neq \varepsilon$.

Remark 5.4.7. The proof we present for the previous result cannot be generalized to the case of more occurrences of $\overline{1}$ in the factor u_1. Indeed, it is not true that $u_1 \in \{1, 0, \overline{0}\}^+$, in general, as showed in the following counterexample.

Example 5.4.8. The double square in Fig. 5.4 is clearly prime, since it is not possible to tile it using an exact polyomino of smaller size. Its boundary word, according to Eq. (5.5), can be factorized as

$$\begin{aligned}
u_1 &= 101\overline{0}1\overline{0}\overline{1}\overline{0}1\overline{0}1\overline{0}1\overline{0}\overline{1}0101\overline{0}1, \\
u_3 &= 0\overline{1}0101\overline{0}1010\overline{1}0\overline{1}\overline{0}\overline{1}0\overline{1}010, \\
k &= 010\overline{1}0\overline{1}\overline{0}\overline{1}0\overline{1}010, \\
\overline{p} &= 0\overline{1}0101\overline{0}1010\overline{1}0.
\end{aligned}$$

Notice that the letter $\overline{1}$ appears in the factor u_1, but this is not in contradiction with Theorem 5.4.6, since two occurrences of it are present.

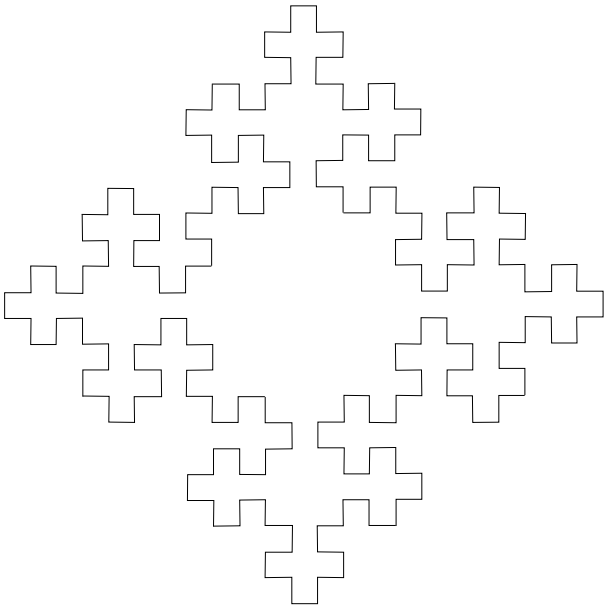


Figure 5.4: A prime double square not in the class $\mathcal{P}^0$.

Proposition 5.4.9 ([7, 11])**.** Let us assume $u_1 = 1a\overline{1}b1$, with $\overline{1} \notin a, b$. Then, $|a|, |b| > 1$.

Proof. It is clear that $a, b \neq \varepsilon$, otherwise the boundary of the polyomino would self-intersect. So, let us assume that $|a| = 1$, that is, $a = \alpha$ with $\alpha \in \{0, \overline{0}\}$. We consider the BN-factor $X = k\tilde{u}_1 u_3$, in particular its factor $\tilde{u}_1 u_3$. We have $\tilde{u}_1 u_3 = 1\tilde{b}\overline{1}\alpha 1 u_3$ and so, since u_3 begins with the letter 0 (Theorem 5.4.5), it must be $\alpha = 0$ to avoid boundary intersections. We now move to the BN-factor $B = u_3\widehat{u}_1\overline{p}$. It holds $\widehat{u}_1\overline{p} = \widehat{\overline{1}b}1\overline{01}\overline{p}$, and so, being 0 the first letter of $\overline{p}$ (Theorem 5.4.5), the boundary of the polyomino self-intersects, contradiction. So, $|a| > 1$. The same argument leads to a contradiction when assuming $|b| = 1$, and this concludes the proof. □

Lemma 5.4.10 ([7])**.** Let us assume $u_1 = 1a\overline{1}b1$, with $\overline{1} \notin a, b$. Then, the position of the occurrence of $\overline{1}$ in u_1 is not the center of the BN-factor $X = k\tilde{u}_1 u_3$.

Proof. The proof proceeds by contradiction. So, let us assume $X = k1\tilde{b}\overline{\mathbf{1}}\tilde{a}1u_3$, with $b1k = \tilde{a}1u_3$ (X is palindrome by Theorem 5.1.4). We highlight in boldface the unique occurrence of $\overline{1}$ in the factor u_1.
We consider the BN-factor $B = u_3\widehat{u}_1\overline{p} = u_3\widehat{\overline{1}b}\mathbf{1}\widehat{a}\overline{1}\overline{p}$, that is palindrome by Theorem 5.1.4.
At first, we suppose that $\overline{1} \in u_3$. By the palindromicity of X, $\overline{1} \in k$ as well.
We make explicit the first occurrence of the letter $\overline{1}$ in u_3, that is, $u_3 = x\overline{1}y$ for some $x, y \in \Sigma^+$, and $\overline{1} \notin x$. Being $b1k = \tilde{a}1u_3$, we have that $k = k'\overline{1}y$, for some $k' \in \Sigma^+$ and such that $\overline{1} \notin k'$, and $b1k' = \tilde{a}1x$. We now analyze the mutual lengths of a and b, and study this last equality.

Case $|b| < |a|$. There exists a factor a' such that $a = a'\tilde{b}$, and so $1k' = \tilde{a}'1x$. Let us now study the palindrome BN-factor $B = u_3\widehat{u}_1\overline{p}$. Replacing $u_1 = 1a'\tilde{b}\overline{\mathbf{1}}b1$ and $u_3 = x\overline{1}y$, we have $B = x\overline{1}y\widehat{\overline{1}b}\mathbf{1}\overline{b}\widehat{a}'\overline{1}\overline{p}$ palindrome, and with $|x| \leq |p|$ since $\overline{1} \notin x$.

First, let us suppose that the inequality is strict, and so the (palindrome) factor can be written as $\overline{p} = x\overline{p}'\overline{1}\tilde{x}$, for some $p' \in \Sigma^*$. So, B is palindrome if and only if $y\overline{1}\widehat{b}\mathbf{1}\overline{b}\widehat{a}'\overline{1}x\overline{p}'$ is. Again, $|y| \leq |x\overline{p}'|$ since $\overline{1} \notin x$. If the inequality is strict, then $|y| < |\overline{p}'|$ and so the letter 1 in boldface matches with some letter 1 in x, when imposing the palindromicity. Being $1 \notin \widehat{b}$ and $\overline{1} \notin x$, it follows that $b = 0$ or $b = \overline{0}$, in contradiction with Proposition 5.4.9. So, $|y| = |x\overline{p}'|$, meaning that B is palindrome if and only if $\widehat{b}\mathbf{1}\overline{b}\widehat{a}'$ is. The letter in boldface is the only occurrence of 1 in the word, meaning that $a' = \varepsilon$. It follows $b = \tilde{a}$, i.e., u_1 is palindrome, and $\overline{p} = u_3$. Moreover, $k' = x$ and so $k = u_3$. Then, the boundary word of the polyomino becomes

$$P = u_1 \vdots u_3 u_1 \vdots u_3 \vdots \overline{u}_1 u_3 | \overline{u}_1 \vdots \overline{u}_3 \overline{u}_1 \vdots \overline{u}_3 \vdots u_1 \overline{u}_3,$$

that is, $P = \varphi(Q)$ with Q the boundary word of the cross and φ the homologous morphism defined as $\varphi(0) = u_3$, $\varphi(1) = u_1$. We reached a contradiction with the definition of prime double square, meaning that $|x| = |p|$ must hold.

Going back to the palindrome $B = x\overline{1}y\overline{1}\widehat{b}\mathbf{1}\overline{b}\widehat{a}'\overline{1}\overline{p}$, being $x = \overline{p}$ and $\overline{1} \notin x$, we have that the BN-factor is palindrome if and only if $y\overline{1}\widehat{b}\mathbf{1}\overline{b}\widehat{a}'$ is.

Let us suppose that $\overline{a}' = y\overline{1}$. We recall that $1k' = \tilde{a}'1x$, and so $k' = \widehat{y}1x$. As a consequence, we have $k = k'\overline{1}y = \widehat{y}1x\overline{1}x$, palindrome, and $u_3 = x\overline{1}y$. So, the last letter of u_3 is different from the first letter of k, in contradiction with Theorem 5.4.5. It follows that B is palindrome if and only if $y\overline{1}\widehat{b}\mathbf{1}\overline{b}\widehat{a}'$ is, and with $y = \overline{a}'\widehat{b}\mathbf{1}y'$ for some $y' \in \Sigma^*$, and $y'\overline{1}\widehat{b}$ is palindrome too. Being $|b| > 1$ (Proposition 5.4.9) and $\overline{1} \notin b$, the second letter of y' must be $\overline{1}$. We finally consider the palindrome $y'\overline{1}\widehat{b}$:

i) $|y'| < |b|$, meaning there exists b' such that $\widehat{b} = \widehat{b}'\tilde{y}'$, and so $b = \overline{y}'b'$ and $1 \notin y'$. In this case, the BN-factor $X = k\tilde{u}_1 u_3$ turns out to be

$$X = (k'\overline{1}y)(1\tilde{b}\overline{\mathbf{1}}\tilde{a}1)(x\overline{1}y) = (k'\overline{1}\overline{a}'\widehat{b}'\tilde{y}'1y')(1\tilde{b}\overline{\mathbf{1}}\tilde{a}1)(\overline{p}\overline{1}\overline{a}'\widehat{b}'\tilde{y}'1y').$$

Since $\overline{1} \notin k'$ and $1 \notin y'$, X is palindrome if they both have length equal to one, or $k' = \varepsilon$, or $y' = \varepsilon$. All three cases lead to a contradiction. In the first one, X is not palindrome; if $k' = \varepsilon$, then the first letter of k would be 1, that is not in accordance with Theorem 5.4.5; finally, $y' \neq \varepsilon$, otherwise $\overline{1}\widehat{b}$ would be palindrome if and only if the first letter of b is 1, leading to $\overline{1}1 \subseteq u_1$, that is not possible.

ii) $|y'| = |b|$. In this case, $y' = \overline{b}$ and $1 \notin y'$. We reach the same contradiction of point *i)*.

iii) $|b| < |y'|$. Since $y'\overline{1}\widehat{b}$ must be palindrome, there exists $y'' \neq \varepsilon$ and palindrome such that $y' = \overline{b}\overline{1}y''$. So, $u_1 = 1a\overline{1}b1$, $u_3 = x\overline{1}y = \overline{p}\overline{1}\overline{a}'\widehat{b}1\overline{b}\overline{1}y'' = \overline{pu}_1 y''$ and $k = k'\overline{1}y = k'\overline{u}_1 y''$, with $\overline{1} \notin \overline{p}, k', a, b$ and y'' palindrome. We replace in the boundary word,

$$P = u_1 \vdots k'\overline{u}_1 y''\tilde{u}_1 \vdots \overline{pu}_1 y'' \vdots \widehat{u}_1\overline{p} | \overline{u}_1 \vdots \overline{k}' u_1 \overline{y}''\widehat{u}_1 \vdots pu_1\overline{y}'' \vdots \tilde{u}_1 p.$$

By Theorem 5.1.4, the BN-factor $A = u_1 k \tilde{u}_1$ is palindrome, if and only if $k'\overline{1}\overline{a}1\overline{b}\overline{1}y''$ is palindrome too. Being $\overline{1} \notin k'$ and $|b| > 1$, there exists y''' such that $y'' = y'''\tilde{k}'$ (palindrome) and $\overline{u}_1 y'''$ is palindrome too. Moving to the palindrome BN-factor $X = k'\overline{u}_1 y'''\tilde{k}'\tilde{u}_1\overline{pu}_1 y'''\tilde{k}'$, we have that $\tilde{k}'\tilde{u}_1\overline{p}$ is palindrome and contains one occurrence only of the letter $\overline{1}$, that one in u_1. So, $\overline{p} = k'$ and u_1 are both palindrome. The boundary word is now

$$P = u_1 \vdots \overline{pu}_1 y'''\overline{pu}_1 \vdots \overline{pu}_1 y'''\overline{u}_1\overline{p} \vdots \overline{u}_1\overline{p}|\overline{u}_1 \vdots pu_1\overline{y}'''pu_1 \vdots pu_1\overline{y}'''u_1 p \vdots u_1 p,$$

with $\overline{u}_1 y'''$ and $y'''\overline{p}$ palindrome (see the BN-factors A and Y). By Theorem 5.4.5, y''' ends with the letter $\overline{1}$ and, since $\overline{1} \notin \overline{p}$, there exists a palindrome q such that $y''' = \overline{p}q$, with $\overline{p}q\overline{p}$ and $\overline{u}_1\overline{p}q$ both palindrome. By Lemma 5.4.2, there exist two palindromes $z_1, z_2 \in \Sigma^+$ such that $\overline{u}_1, \overline{p}$ and q can be written as their concatenation. We underline that at least one among z_1 and z_2 has length strictly greater than one, since $\overline{u}_1$ contains both occurrences of the letter 1 and $\overline{1}$.

Coming to an end, we have finally obtained that all the factors u_1, u_3, k and $\overline{p}$ are concatenations of the palindromes $z_1, z_2, \overline{z}_1$ and $\overline{z}_2$. So, we can define the homologous morphism $\varphi(0) = z_1$, $\varphi(1) = z_2$ such that $P = \varphi(Q)$, for some Q different from the unit square. So P is not prime, contradiction.

Case $|a| < |b|$. We recall that we have $u_3 = x\overline{1}y$, $k = k'\overline{1}y$, with $\overline{1} \notin x, k'$, and $b1k' = \tilde{a}1x$.

Being $|a| < |b|$, there exists a word b' such that $b = \tilde{a}b'$ and $b'1k' = x$. The BN-factor $B = u_3\widehat{u}_1\overline{p} = x\overline{1}y\overline{1}\widehat{b}'\overline{a}\mathbf{1}\widehat{a}\overline{1}\overline{p}$ is palindrome if and only if $\overline{p} = x\overline{p}'\overline{1}\tilde{x}$ for some $p' \in \Sigma^*$, being $\overline{1} \notin x$. By applying the same argument we used in the previous case, we reach again a contradiction. The same result holds if $|a| = |b|$.

So, we can conclude that the factor u_3 does not contain the letter $\overline{1}$. Let us consider again the BN-factors of the boundary word P, palindrome by Theorem 5.1.4.
$B = u_3\overline{1}\widehat{b}\mathbf{1}\widehat{a}\overline{1}\overline{p}$ is palindrome with $\overline{1} \notin u_3$, and so $|u_3| \leq |\overline{p}|$. If they have the same length, then $u_3 = \overline{p}$ and $b = \tilde{a}$, so that u_1 is palindrome. Moreover, studying $X = k\tilde{u}_1 u_3$ we argue $k = u_3$, and so an homologous morphism φ can be defined to show that P is not prime. Then, $|u_3| < |\overline{p}|$, and there exists $\overline{p}'$ such that $\overline{p} = \overline{p}'\overline{1}\tilde{u}_3$ and $\widehat{b}\mathbf{1}\widehat{a}\overline{1}\overline{p}'$ are both palindrome. Since $\overline{1} \notin u_3$, we can write the palindrome as $\overline{p} = u_3\overline{p}''\overline{1}\tilde{u}_3$, for some $p'' \in \Sigma^*$.
We replace in $B = u_3\overline{1}\widehat{b}\mathbf{1}\widehat{a}\overline{1}u_3\overline{p}''\overline{1}\tilde{u}_3$, that is palindrome if and only if $\widehat{b}\mathbf{1}\widehat{a}\overline{1}u_3\overline{p}''$ is, with $\overline{1} \notin u_3$. Since $\overline{1} \notin a, b, u_3$, and $|a|, |b| > 1$ by Proposition 5.4.9, we can iterate the same argument until we show that $\overline{p}$ is a concatenation of the factors u_1 and u_3, both palindromes. Moving to X, we see that the same property holds for k, and so that P is not prime since a non-trivial homologous morphism can be defined to get it from a polyomino that is not the unit square.
Since all the cases have been analyzed, the proof is given.

□

So, the unique occurrence of $\overline{1}$ in u_1 appears in the first or in the second half of the BN-factor $X = k\tilde{u}_1 u_3$. A trivial consequence of the previous result is that $|u_3| \neq |k|$.

We now present two results concerning the position of $\overline{1} \in u_1$ in the first and second half of X.

Lemma 5.4.11 ([7]). Let us assume $u_1 = 1a\overline{1}b1$, with $\overline{1} \notin a, b$. If the (unique) occurrence of the letter $\overline{1}$ in u_1 is in the first half of the BN-factor $X = k\tilde{u}_1 u_3$, then the BN-factor $B = u_3\widehat{u}_1\overline{p}$ is palindrome if and only if

i) the center of B is $z''u_1k$, for a proper $z'' \in \Sigma^+$, or

ii) the center of B is $k\widehat{u}_1\overline{p}''$, for a proper $\overline{p}'' \in \Sigma^+$.

Proof. Being $\overline{1} \in u_1$ in the first half of the palindrome X, we have $|k| < |u_3|$ and $u_3 = x\overline{1}b1k$, for a proper $x \in \Sigma^+$ such that $\tilde{a}1x$ is palindrome. Two cases have to be considered:

Case $|a| < |x|$. It follows $x = z1a$, for a proper palindrome $z \in \Sigma^+$, and so $u_3 = z1a\overline{1}b1k = zu_1k$. The palindrome BN-factor $B = u_3\widehat{u}_1\overline{p}$ turns out to be

$$B = z1a\overline{\mathbf{1}}b1k\overline{\mathbf{1}}\widehat{b}1\widehat{a}\overline{\mathbf{1}}\overline{p}.$$

Let us suppose that z has length equal to one, and so $z = 0$ by Theorems 5.1.4 and 5.4.5. Being $\overline{1} \notin a$ and $\overline{p}$ palindrome, it follows that $01a$ is a prefix of $\overline{p}$. Then, the word $v = \widehat{a}\overline{1}01a$ is a factor of P, and, for any starting letter of a, represents a closed path, in contradiction with the definition of polyomino. So, $|z| \geq 2$ and also $|z| \neq |\overline{p}|$ (B is palindrome). Two cases arise:

i) $|p| < |z|$, and so $z = \overline{p}\overline{1}z'$, for a proper $z' \in \Sigma^+$, and B is palindrome if and only if $z'1a\overline{\mathbf{1}}b1k\overline{\mathbf{1}}\widehat{b}1\widehat{a}$ is. If $z' = \overline{a}$, then $|a| = |b| = 1$ must hold, in contradiction with Proposition 5.4.9. So, being $\overline{1} \notin a$, it holds $z' = \overline{a}1\overline{b}\overline{1}z''$, for a proper $z'' \in \Sigma^+$. Replacing in z, that we recall is a palindrome, we obtain $z = \overline{pu}_1z''$. We get

$$B = \overline{pu}_1z''u_1k\widehat{u}_1\overline{p},$$

that is palindrome if and only if $z''u_1k$ is, that is also its center. So, we are in case *i)* of our claim.

ii) $|z| < |p|$, and so $\overline{p} = \overline{p}'1z$ for a proper $p' \in \Sigma^+$, and B is palindrome if and only if $a\overline{\mathbf{1}}b1k\overline{\mathbf{1}}\widehat{b}1\widehat{a}\overline{\mathbf{1}}\overline{p}'$ is. As discussed in the previous case, $|p'| \neq |a|$, $\overline{p}$ palindrome and $\overline{1} \notin a, b$ lead to write $\overline{p}' = \overline{p}''1b\overline{1}\tilde{a}1z$, and then $\overline{p} = \overline{p}''u_1z$ for a proper $p'' \in \Sigma^+$. We get

$$B = u_3\widehat{u}_1\overline{p} = zu_1k\widehat{u}_1\overline{p}''u_1z,$$

palindrome if and only if its center, that is $k\widehat{u}_1\overline{p}''$, is palindrome too. So, we are in case *ii)* of our claim.

Case $|x| < |a|$**.** It follows $a = a'1x$, for a proper $a' \in \Sigma^+$; in particular, $\overline{1} \notin x$. The palindrome BN-factor turns out to be

$$B = x\overline{\mathbf{1}}b1k\overline{\mathbf{1}}\widehat{b}1\widehat{a}\overline{\mathbf{1}}\overline{p},$$

with $|x| < |p|$ since $\overline{1} \notin x$ and they cannot have the same length (Proposition 5.4.9). So, there exists $\overline{p}' \in \Sigma^+$ such that $\overline{p} = \overline{p}'\overline{1}\tilde{x}$, and

$$B = x\overline{\mathbf{1}}b1k\overline{\mathbf{1}}\widehat{b}1\widehat{a}\overline{\mathbf{1}}\overline{p}'\overline{\mathbf{1}}\tilde{x}.$$

As previously discussed, since $\overline{1} \notin b$, there exists $\overline{p}'' \in \Sigma^+$ such that $\overline{p} = \overline{p}''1\tilde{b}\overline{1}\tilde{x}$, and the center of B is then $k\widehat{u}_1\overline{p}''$. So, we are in case *ii*) of our claim.

We discussed all the possible cases, and this concludes the proof. □

Lemma 5.4.12 ([7])**.** Let us assume $u_1 = 1a\overline{1}b1$, with $\overline{1} \notin a, b$. If the (unique) occurrence of the letter $\overline{1}$ in u_1 is in the second half of the BN-factor $X = k\tilde{u}_1u_3$, then

i) the factor k can be expressed as $k = \tilde{u}_3u_1k''$, for a proper palindrome k'', or

ii) there exist $k', x \in \Sigma^+$ such that $k = \tilde{u}_31a\overline{1}k'$, $u_3 = x\overline{1}k'$ and $\tilde{x}\overline{1}a$ is palindrome.

Proof. Being $\overline{1} \in u_1$ in the second half of the palindrome X, we have $|u_3| < |k|$ and $k = \tilde{u}_31a\overline{1}k'$, for a proper $k' \in \Sigma^+$ such that $k'1\tilde{b}$ is palindrome. Two cases have to be considered:

Case $|b| < |k'|$**.** There exists a palindrome $k'' \in \Sigma^+$ such that $k' = b1k''$ and, replacing, $k = \tilde{u}_3u_1k''$. So, we are in case *i*) of our claim.

Case $|k'| < |b|$**.** There exists $b' \in \Sigma^+$ such that $b = k'1b'$, and so $\overline{1} \notin k'$. We also recall that $k = \tilde{u}_31a\overline{1}k'$ is a palindrome.

If $k' = \tilde{a}1u_3$, then $|u_3| < |k'| < |b| < |u_1|$, and this violets condition (5.4). So, $u_3 = x\overline{1}k'$ must hold, with $x \in \Sigma^+$ such that $\tilde{x}1a$ is palindrome, and we are in case *ii*) of our claim. The same conclusion is reached when considering $|b| = |k'|$.

We discussed all the possible cases, and this concludes the proof. □

Remark 5.4.13. The outcomes we provide in Lemma 5.4.12 show an important symmetrical property. Indeed, in both cases, it results that the study of the palindrome k can be performed as the study of the palindrome $X = k\tilde{u}_1u_3$. In case *i*), just replacing the role of k in X with k''. In case *ii*), we are actually in the same case presented in Lemma 5.4.11, where k' replaces the role of the suffix $b1k$ of u_3 and $\overline{1} \in u_1$ is in the first half of X (see the first lines of the proof).

The symmetrical properties we remarked above allow to iteratively apply Lemmas 5.4.11 and 5.4.12 while analyzing the palindromicity of the BN-factors X and B, thus crumbling them into two different common parts that alternate all over the boundary word of the prime double square P.

Theorem 5.4.14 ([7]). If u_1 is such that $u_1 = 1a\overline{1}b1$, with $\overline{1} \notin a, b$, then there exist $\alpha, \beta, \gamma \geq 0$ and a non-empty palindrome $q \in \Sigma^+$ such that $k = (qu_1)^\alpha q$, $u_3 = (qu_1)^\beta q$ and $\overline{p} = (qu_1)^\gamma q$. Moreover, u_1 is palindrome.

Sketch of the proof. The BN-factors X and B, as well as k, are all palindromes. This property allows to apply Lemmas 5.4.11 and 5.4.12 iteratively, according to the lengths of the involved factors, finally reaching the writing of u_3, k and $\overline{p}$ as concatenations of $u_1, \tilde{u}_1$ and $q, \tilde{q}$, for some non-empty $q \in \Sigma^+$. Then, by imposing the palindromicity of $X = k\tilde{u}_1 u_3$, $B = u_3\widehat{u}_1\overline{p}$ and $k, \overline{p}$ at the same time, we conclude that both u_1 and q are palindromes. As a consequence, k, u_3 and $\overline{p}$ can be written as $k = (qu_1)^\alpha q$, $u_3 = (qu_1)^\beta q$ and $\overline{p} = (qu_1)^\gamma q$, for some $\alpha, \beta, \gamma \geq 0$. □

Remark 5.4.15. It is important to underline that $\overline{u}_1$ is not a factor of q, as arises from the proofs of Lemmas 5.4.11 and 5.4.12. Moreover, $|u_1| > 1$ by hypothesis.

A direct consequence of Theorem 5.4.14 is the following

Corollary 5.4.16 ([7]). The word $u_1 = 1a\overline{1}b1$ cannot be a factor of the boundary word P of a prime double square in its standard factorization.

Proof. By contradiction, let us suppose that $u_1 = 1a\overline{1}b1$ is in the word P written as in Eq. (5.5). From Theorem 5.4.14, there exist $\alpha, \beta, \gamma \geq 0$ and $q \in \Sigma^+$ such that $k = (qu_1^\alpha)q$, $u_3 = (qu_1^\beta)q$ and $\overline{p} = (qu_1)^\gamma q$, namely,

$$P = u_1 \vdots (qu_1)^\alpha qu_1 \vdots (qu_1)^\beta q \vdots \overline{u}_1(qu_1)^\gamma q|\overline{u}_1 \vdots (\overline{qu}_1)^\alpha \overline{qu}_1 \vdots (\overline{qu}_1)^\beta \overline{q} \vdots u_1(\overline{qu}_1)^\gamma \overline{q}.$$

Furthermore, we recall that both u_1 and q are palindromes. As a consequence, the morphism $\varphi(0) = q$, $\varphi(1) = u_1$ is homologous, and maps the double square

$$Q = 1(01)^\alpha 01(01)^\beta 0\overline{1}(01)^\gamma 0|\overline{1}(\overline{01})^\alpha \overline{01}(\overline{01})^\beta \overline{0}1(\overline{01})^\gamma \overline{0}$$

in P. In case $\alpha = \beta = \gamma = 0$, Q is the cross. Moreover, from Remark 5.4.15, φ is not the identity, since u_1 has always length strictly greater than one. It follows that P is not prime, in contradiction with the hypothesis. □

All the results we achieved provide a proof of Theorem 5.4.6.

An alternative factorization for the boundary word

The result we reached concerning the factor u_1 is a very strong property, even if cannot be generalized in case of a higher number of occurrences of $\overline{1}$ (see Example 5.4.8). To make our studies easier, we now focus our attention on a special subclass of prime double squares, that is defined by adding the further constraint that $u_1 \in \{1, 0, \overline{0}\}^+$. We show that the previous hypothesis actually leads to the extension of the result we achieved for u_1 to all the other factors that appear in Eq. (5.5), thus leading to the final characterization of this subclass of tiles.

Looking at Theorem 5.4.14, we can argue that if any of the factors $u_1, u_3, k, \overline{p}$ contains all the letters of the alphabet Σ, then it is always possible to write all of them as the concatenation of two palindromes, thus leading to the definition of a non-trivial homologous morphism that makes the considered polyomino to be not prime.
To simplify the study of such properties, as well as the proofs of an analogue of Theorem 5.4.6 for u_3, k and $\overline{p}$, we define a second standard factorization for the boundary word P of a prime double square. The new factorization we propose can be seen as a dual of (5.5), where the factor u_3, instead of u_1, is highlighted and plays a relevant role. Our aim is to give more relevance to u_3 since we have no information concerning this factor, both in terms of its length (w.r.t. u_2 and u_4) and of forbidden letters, if any.
As a first result we show that the factors $u_2 = k\tilde{u}_1$ and $u_4 = \widehat{u}_1\overline{p}$ can be written in terms of u_3 too. This is possible due to the following theorem, where we underline that the absence of the letter $\overline{1}$ in u_1 is fundamental to carry on the proof.

Theorem 5.4.17 ([11]). If P is the boundary word of a prime double square in its standard factorization, (5.5), then $|u_3| < |u_2|, |u_4|$ holds.

Proof. Without loss of generality, we assume $n_1 = n_3 = 0$. By contradiction, let us suppose that $|u_3| > |u_2| = |k\tilde{u}_1|$, and consider the BN-factor $X = u_2u_3 = k\tilde{u}_1u_3$. By Theorem 5.1.4, X is a palindrome, meaning that there exists $x \in \Sigma^+$ such that $u_3 = xu_1k$, and x is palindrome too. We now move to the BN-factor $B = u_3u_4 = xu_1k\widehat{u}_1\overline{p}$, that is again a palindrome by Theorem 5.1.4. It is clear that $|x| \neq |\overline{p}|$, and we can suppose without loss of generality that $|x| < |\overline{p}|$ (the proof is similar in the other case).
By Theorem 5.4.6, $\overline{1} \notin u_1$ and $1 \notin \widehat{u}_1$, and so $\tilde{u}_1k$ must be a suffix of $\overline{p}$, namely, there exists $\overline{p}' \in \Sigma^+$ such that $\overline{p} = \overline{p}'\tilde{u}_1x$ and $\overline{p}$ and $k\widehat{u}_1\overline{p}'$ are both palindrome. So we have that

$$k, \quad x, \quad \overline{p}'\tilde{u}_1x, \quad k\widehat{u}_1\overline{p}'$$

are all palindrome at the same time.
We consider the third word in the list above: if $\overline{p}' = x$, then u_1 is palindrome and, by Lemma 5.4.2, $k = (au_1)^\alpha a$ and $x = (au_1)^\beta a$ for some $\alpha, \beta \geq 0$ and $a \in \Sigma^+$ palindrome. Then, it is possible to define the homologous morphism $\varphi(0) = a$, $\varphi(1) = u_1$ to show that P is not prime, contradiction.
So $\overline{p}' \neq x$, and $\overline{p} = xu_1\overline{p}''\tilde{u}_1x$ for some palindrome $\overline{p}'' \in \Sigma^+$. Replacing in the BN-factor $B = xu_1k\widehat{u}_1\overline{p}$ and imposing the palindromicity of its center, that is $k\widehat{u}_1xu_1\overline{p}''$, we get a palindrome having the same form of the BN-factor B previously analyzed. Then, iterating the same argument by imposing the palindromicity of X, B, k and $\overline{p}$, we finally obtain a proper morphism φ to show that P is not prime, contradiction. Then, $|u_3| < |u_2|$.
The same argument shows that $|u_3| < |u_4|$, by considering $u_3 = \overline{pu}_1x$ for a proper $x \in \Sigma^+$ and studying the palindrome $X = u_2u_3 = k\tilde{u}_1\overline{pu}_1x$. □

A direct consequence is that u_2 and u_4 can be written in terms of u_3, after imposing the palindromicity of the BN-factors of P, that we remind to be

$$\begin{aligned}
A &= w_1w_2 = (\tilde{u}_2\overline{u}_4)^{n_1}u_1(\tilde{u}_3u_1)^{n_2}u_2,\\
B &= w_3w_4 = (\tilde{u}_4u_2)^{n_3}u_3(\widehat{u}_1u_3)^{n_4}u_4,\\
X &= w_2w_3 = (\tilde{u}_3u_1)^{n_2}u_2(\tilde{u}_4u_2)^{n_3}u_3,\\
Y &= w_4\widehat{w}_1 = (\widehat{u}_1u_3)^{n_4}u_4(\widehat{u}_2u_4)^{n_1}\widehat{u}_1.
\end{aligned}$$

We have that $u_2 = \tilde{u}_3 t$ and $u_4 = \overline{s}\tilde{u}_3$, for proper non-empty palindromes $t, s \in \Sigma^+$.

Remark 5.4.18. The first (and last) letter of the palindromes t and $\overline{s}$ are 1 and $\overline{1}$, respectively. This is a direct consequence of Theorem 5.4.5 and of the double factorization of u_2 and u_4 as $u_2 = k\tilde{u}_1 = \tilde{u}_3 t$ and $u_4 = \widehat{u}_1\overline{p} = \overline{s}\tilde{u}_3$.

We conclude recalling here the standard factorization presented in (5.5),

$$P = (u_1 k\tilde{u}_1 p)^{n_1} u_1 \vdots k\tilde{u}_1 \vdots (\overline{pu}_1 k\tilde{u}_1)^{n_3} u_3 \vdots \widehat{u}_1\overline{p} | (\overline{u}_1\overline{k}\widehat{u}_1\overline{p})^{n_1}\overline{u}_1 \vdots \overline{k}\widehat{u}_1 \vdots (pu_1\overline{k}\widehat{u}_1)^{n_3}\overline{u}_3 \vdots \tilde{u}_1 p,$$

as well as the second standard factorization, where all the occurrences of u_3 are highlighted:

$$P = (tu_3 s\widehat{u}_3)^{n_1} u_1 \vdots \tilde{u}_3 t \vdots (u_3\overline{s}\tilde{u}_3 t)^{n_3} u_3 \vdots \overline{s}\tilde{u}_3 | (\overline{t}\overline{u}_3\overline{s}\tilde{u}_3)^{n_1}\overline{u}_1 \vdots \widehat{u}_3\overline{t} \vdots (\overline{u}_3 s\widehat{u}_3\overline{t})^{n_3}\overline{u}_3 \vdots s\widehat{u}_3. \quad (5.6)$$

Both Equations (5.5) and (5.6) will be crucial in the characterization of the alphabet on which the factors u_3, k, p, t and s are actually defined.

Example 5.4.19. The polyomino in Figure 5.5 is a prime double square, with (half) boundary word

$$P = \underbrace{1}_{u_1} \vdots \underbrace{010\overline{1}0\overline{1}0101}_{u_2} \vdots \underbrace{0\overline{1}0\overline{1}010}_{u_3} \vdots \underbrace{\overline{1}0\overline{1}0\overline{1}010\overline{1}0\overline{1}0}_{u_4} |$$

Its two standard factorizations, highlighted in Fig. 5.6(a) and (b), respectively, are

$$u_2 \overset{k\tilde{u}_1}{=} (010\overline{1}0\overline{1}010)(1) \overset{\tilde{u}_3 t}{=} (010\overline{1}0\overline{1}0)(101),$$
$$u_4 \overset{\widehat{u}_1\overline{p}}{=} (\overline{1})(0\overline{1}0\overline{1}010\overline{1}0\overline{1}0) \overset{\overline{s}\tilde{u}_3}{=} (\overline{1}0\overline{1}0\overline{1})(010\overline{1}0\overline{1}0),$$

with the following involved factors:

$$\begin{aligned} u_1 &= 1 \\ u_3 &= 0\overline{1}0\overline{1}010 \\ k &= 010\overline{1}0\overline{1}010 \\ \overline{p} &= 0\overline{1}0\overline{1}010\overline{1}0\overline{1}0 \\ t &= 101 \\ \overline{s} &= \overline{1}0\overline{1}0\overline{1}. \end{aligned}$$

The alphabet of the other factors of a prime double square

The similarities between the two standard factorizations of the boundary word of a prime double square, described in Eq. (5.5) and (5.6), are evident. While in the former the factor u_1, together with $\tilde{u}_1, \overline{u}_1, \widehat{u}_1$, alternates with u_3, k and p, in the latter the same role is played by u_3. As a consequence, it is possible to carry on a study of the factor u_3 that is analogous to that of u_1, following the sequence of proofs described in Proposition 5.4.9, Lemmas 5.4.10, 5.4.11, 5.4.12, Theorem 5.4.14, and Corollary 5.4.16, but using P written as in (5.6) instead of (5.5). The final result is the following theorem, that is an analogous of Theorem 5.4.6 for the factor u_3:

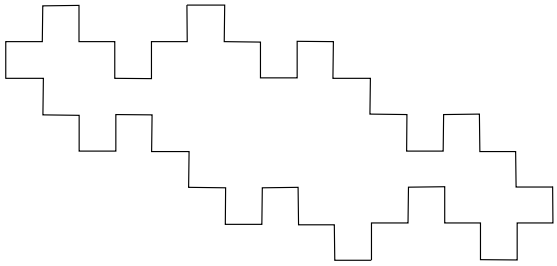


Figure 5.5: The prime double square described in Example 5.4.19.

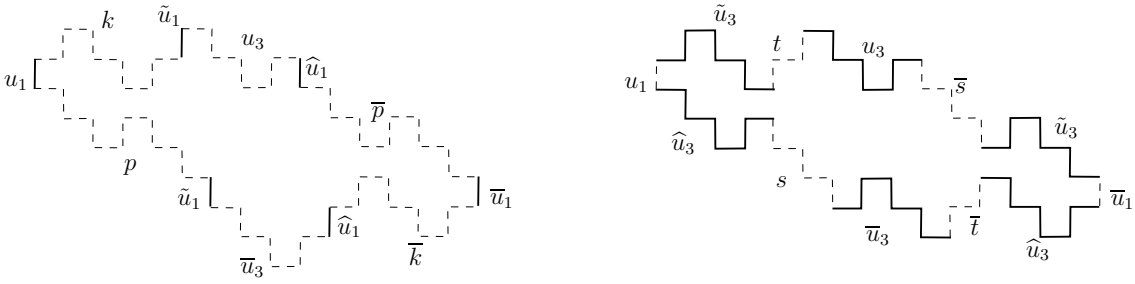


Figure 5.6: The prime double square described in Example 5.4.19 and the two standard factorizations of its boundary. In (a), the factors appearing in (5.5) are highlighted, while in (b) those ones concerning (5.6).

Theorem 5.4.20 ([11]). Given P the boundary word of a prime double square in its standard factorization, (5.6), the factor u_3 contains three letters of Σ only, specifically, $u_3 \in \{1, 0, \overline{1}\}^+$.

Remark 5.4.21. We underline that even in this case, the proof given for u_1 actually provides a dual result for a unique occurrence of $\overline{0}$ in u_3, while nothing can be stated in the general case. However, from now on we proceed under the hypothesis that $\overline{0} \notin u_3$, and focus on this special subclass of prime double squares. We will call it $\mathcal{P}^0$.

On the other hand, Eq. (5.5) and (5.6) can be exploited at the same time to reach similar results concerning the other factors that appear in the standard factorizations of the boundary word. We provide here the study of the palindrome k as an example.

Theorem 5.4.22 ([11]). Given P the boundary word of a prime double square in $\mathcal{P}^0$ in its standard factorization, (5.5), the factor k contains three letters of Σ only, specifically, $k \in \{1, 0, \overline{1}\}^+$.

Proof. By contradiction, let us suppose $\overline{0} \in k$. We consider the BN-factor $X = k\tilde{u}_1 u_3$ and study its palindromicity. Since $\overline{0} \notin u_3$, see Theorem 5.4.20, it must be $k = \tilde{u}_3 k'$, for a proper $k' \in \Sigma^+$. Notice that k' cannot be the empty word and also contains the letter $\overline{0}$. As a consequence, being k palindrome, we can write $k = \tilde{u}_3 k'' u_3$ for some palindrome $k'' \in \Sigma^+$ such that $\overline{0} \in k''$.

Moving to the boundary word P factorized as in (5.6), we have $X = \tilde{u}_3 t u_3 = \tilde{u}_3 k'' u_3 \tilde{u}_1 u_3$, if and only if $t = k'' u_3 \tilde{u}_1$. Being t palindrome, $\overline{0} \in k''$ and $\overline{0} \notin u_3$, there exist $t', k''' \in \Sigma^+$ such that $t = u_1 t'$ and $k'' = u_1 k'''$.
We now have $k = \tilde{u}_3 u_1 k''' u_3$ and $X = \tilde{u}_3 u_1 k''' u_3 \tilde{u}_1 u_3$ both palindrome, if and only if $t = u_1 k''' u_3 \tilde{u}_1$ is palindrome too. By iterating the same argument, we finally obtain that both k and t are a concatenation of the factors $u_1, \tilde{u}_1, u_3$ and $\tilde{u}_3$. Moreover, by imposing the palindromicity on k and t at the same time, we get that u_1 and u_3 are palindrome too. We also underline that $\overline{0} \in u_1$, and so $|u_1| \geq 3$ must hold.
We now consider the BN-factor $B = u_3 \widehat{u}_1 \overline{p}$, and two possible cases:

i) $\overline{0} \in \overline{p}$. Using the previous argument, it is possible to show that $\overline{p}$ is a concatenation of the factors u_3 and $\overline{u}_1$, both palindromes.

ii) $\overline{0} \notin \overline{p}$. We can have $|\overline{p}| \geq |u_3|$ or vice versa. Using again the previous argument, we obtain that $\overline{p}$ is a concatenation of the palindromes u_3 and $\overline{u}_1$ (first case), or that u_3 is a concatenation of the palindromes $\overline{p}$ and $\overline{u}_1$ (second case).

In both cases, it is possible to define the homologous morphism $\varphi(0) = u_3$ (or $\varphi(0) = \overline{p}$, depending on the mutual lengths of u_3 and $\overline{p}$), $\varphi(1) = u_1$, and show that P is not prime, contradiction. We remark that $|u_1| \geq 3$ ensures that φ is not the identity.

□

Similar results can be achieved for the factors p, t and s, see Theorem 5.4.23.

In this section, we have considered two standard factorizations for the boundary word of a prime double square in the class $\mathcal{P}^0$, described in Eq. (5.5) and (5.6), and carried on a study of the factors that appear in both of them, namely, the words u_1, u_3, k, p, t and s. For each of them we have shown that one of the letters of the alphabet Σ is forbidden, in other words, each factor is defined on an alphabet of three letters only. All the previous results are summarized in the following

Theorem 5.4.23 ([11]). Let P be the boundary word of a prime double square in the class $\mathcal{P}^0$. Considering the factorizations (5.5) and (5.6), the following results hold:

i) $u_1, t, s \in \{1, 0, \overline{0}\}^+$, so that $\overline{1} \notin u_1, t, s$,

ii) $u_3, k, \overline{p} \in \{1, 0, \overline{1}\}^+$, so that $\overline{0} \notin u_3, k, \overline{p}$.

We remark again that a complete proof has been provided for the factors u_1 and k, while in the other cases it is sufficient to follow the same arguments by exploiting the similarities between the two standard factorizations of P. We also recall that we impose $\overline{1} \notin u_1$ and $\overline{0} \notin u_3$, being P in the class $\mathcal{P}^0$.

As a final remark of the section, we remind again that all the proofs we have shown are still valid in the more general case of $n_1 \geq 0$, $n_3 \geq 0$, or both.

5.4.2 Characterization of the factors of prime double squares in $\mathcal{P}^0$

From now on, even if not explicitly written, we deal with prime double squares in the class $\mathcal{P}^0$ only, i.e., those ones characterized in Theorem 5.4.23.
Theorem 5.4.23 characterizes completely the alphabets of the factors involved in Eq. (5.5) and (5.6). Further properties concerning u_1 and u_3 can be detected, finally reaching their complete characterization.
We start with a result concerning their palindromicity:

Theorem 5.4.24 ([11]). Let us consider P the boundary word of a prime double square written as in (5.5) or (5.6). The following statements hold:

i) if $|u_1| > 1$, then u_1 is not palindrome, and

ii) if $|u_3| > 1$, then u_3 is not palindrome.

Proof. The proof proceeds by contradiction.

i) Let us suppose that u_1 is palindrome, and that its second letter is $\overline{0}$. Then, being $X = k\tilde{u}_1 u_3$ palindrome and $\overline{0} \notin u_3, k$, we have that $k = u_3$. Moving to the BN-factor $B = u_3\overline{u}_1\overline{p}$, we can use the same argument of Theorem 5.4.22 to show that $\overline{p}$ is a concatenation of factors $\overline{u}_1$ and u_3. As a consequence, the homologous morphism $\varphi(0) = u_3$, $\varphi(1) = u_1$ is well defined, and maps a polyomino different from the unit square in P, contradiction. If the second letter of u_1 is 0 the same conclusion can be reached, but studying B at first ($\overline{p} = u_3$) and X later (k concatenation of the palindromes u_1 and u_3). In both cases we reach a contradiction with the definition of prime double square, and so we conclude that u_1 is not palindrome.

ii) A proof analogous to case *i)* holds for the factor u_3.

We conclude that, unless of length one, u_1 and u_3 are never palindrome. □

We continue the analysis of the words u_1 and u_3, providing further results that lead to their complete characterization.

Corollary 5.4.25 ([11]). The following statements hold:

i) Unless $|u_1| = 1$, the factor u_1 contains at least one occurrence of the letter 0 and one occurrence of the letter $\overline{0}$.

ii) Unless $|u_3| = 1$, the factor u_3 contains at least one occurrence of the letter 1 and one occurrence of the letter $\overline{1}$.

The result is a straightforward consequence of Theorems 5.4.23 and 5.4.24.

Theorem 5.4.26 ([11]). The following statements hold:

i) In u_1, the factor defined by the first and last occurrence of the letter 0 (respectively, $\overline{0}$) is palindrome.

ii) In u_3, the factor defined by the first and last occurrence of the letter 1 (respectively, $\overline{1}$) is palindrome.

Proof. It is sufficient to exploit the palindromicity of the BN-factors of a prime double square in its standard factorization(s).

i) We consider the boundary word P factorized as in (5.5), and the BN-factors $X = k\tilde{u}_1 u_3$ and $B = u_3\widehat{u}_1\overline{p}$. By Theorem 5.4.23 and Corollary 5.4.25, point *i*), when considering X, the letter $\overline{0}$ appears in the factor u_1 only, so that the palindrome at the center of X is defined by its first and last occurrence. It follows that the factor between the first and last $\overline{0}$ in u_1 has to be a palindrome too.

Similarly, when considering $B = u_3\widehat{u}_1\overline{p}$, the first and last occurrence of $\overline{0}$ in $\overline{u}_1$ define the palindrome that is the center of the BN-factor. Moving to the complement, u_1, it directly follows that the factor between the first and last occurrence of the letter 0 in it is a palindrome, thus reaching the thesis.

ii) The proof is analogous, by considering P factorized as in (5.6) and the property $\overline{1} \notin t, s$.

We underline that Corollary 5.4.25 is fundamental to reach the proof, since it ensures the presence of the letter $\overline{0}$ both in u_1 and $\widehat{u}_1$, as well as $\overline{1}$ in u_3 and $\overline{u}_3$ in point *ii*). □

The result we achieved in Theorem 5.4.26 is actually a characterization of the factors u_1 and u_3, that can be explicitly expressed in terms of the alphabet Σ.

Theorem 5.4.27 ([11]). The factors u_1 and u_3 are characterized as follows:

i) The factor u_1 either is equal to 1 or it has one of the following forms

$$u_1 = \Big((10)^\alpha(1\overline{0})^\beta\Big)^\gamma 1$$
$$u_1 = \Big((1\overline{0})^\alpha(10)^\beta\Big)^\gamma 1$$

for some $\alpha, \beta, \gamma \geq 1$.

ii) The factor u_3 either is equal to 0 or it has one of the following forms

$$u_3 = \Big((0\overline{1})^\alpha(01)^\beta\Big)^\gamma 0$$
$$u_3 = \Big((01)^\alpha(0\overline{1})^\beta\Big)^\gamma 0$$

for some $\alpha, \beta, \gamma \geq 1$.

Remark 5.4.28. The exponents α, β and γ that appear in Theorem 5.4.27 vary independently each other.

Although u_1 and u_3 are explicitly characterized, a result similar to Theorem 5.4.27 cannot be reached for the factors k, p, t and s in (5.5) and (5.6). This is due to the fact that they strictly depend on the choice of u_1 and u_3, as well as on the length of the boundary word P. However, once all these parameters are fixed, they are all uniquely determined, as shown in the next section.

5.5 Generation of $\mathcal{P}^0$

Starting from the study of the couple free conjecture, we have defined two standard forms (factorizations) for the boundary word of a prime double square, thus reducing its study to the study of six factors only. Basing on the characteristics of the factors u_1, u_3, k, p, t and s, now we are ready to provide the complete generation of prime double square tiles that belong to the class $\mathcal{P}^0$, also presented in our work [11]. We generate prime double squares with respect to the half-length of the boundary word, that is, the semi-perimeter of the corresponding polyomino. This is possible due to the fact that the BN-factorization(s) divide the boundary word of the polyomino in two symmetric parts, so that we can reduce our analysis to the fist half only.
Differently from the algorithm presented by the authors in [21], our generation is free from any repetition and outlier, thus naturally resulting in the enumeration of this class of tiles, with respect to the semi-perimeter.
We dedicate this section to the description of the generating strategy, while we will sketch the enumeration in Section 5.6.

For the sake of simplicity, since we work with the semi-perimeter, from now on we will write the first half of the boundary word only. Moreover, we will partition prime double squares in three mutual disjoint subclasses, to simplify the generating process. The classification will be with respect to the lengths of u_1 and u_3. We remind that $|u_1| \leq |u_3|$ always holds, by hypothesis.

5.5.1 Class $\mathcal{A}$

We put in the first class all those prime double squares having $u_1 = 1$ and $u_3 = 0$. For the generation of this class, we consider the boundary word factorized as in (5.5), so we have

$$P = (1k1p)^{n_1}1 \vdots k1 \vdots (\overline{p}\overline{1}k1)^{n_3}0 \vdots \overline{1}\overline{p}|$$

We need to determine the factors k and $\overline{p}$, that can vary according to the length of the word. We recall that they are both palindrome and start (and end) with the letter 0.

Basic case $\mathcal{A}_0$

We first consider the case $n_1 = n_3 = 0$.
The BN-factors of the tile are then $X = k10$ and $B = 0\overline{1}\overline{p}$, meaning that k and $\overline{p}$ can vary independently of each other. We observe that:

i) Unless of length equal to one, the second letter of k must be 1, to avoid boundary intersections and to ensure X to be palindrome.

ii) For the same reason, the second letter of $\overline{p}$, unless the factor has length equal to one, must be $\overline{1}$.

So, the two factors are of type

$$k = \begin{cases} 0 \\ 0k'0, \quad \text{with } k' \in \{1, 0, \overline{1}\}^+ \text{ a palindrome starting with the letter 1,} \end{cases}$$

$$\overline{p} = \begin{cases} 0 \\ 0\overline{p}'0, \quad \text{with } \overline{p}' \in \{1, 0, \overline{1}\}^+ \text{ a palindrome starting with the letter } \overline{1}, \end{cases}$$

and the (first half of the) boundary word of a tile in this class is

$$P_{\mathcal{A}_0} = 1 \vdots k1 \vdots 0 \vdots \overline{1}\overline{p}.$$

The smallest polyomino in $\mathcal{A}_0$ is the cross, obtained by choosing $k = \overline{p} = 0$. We underline that the cross is also the prime double square with the smallest perimeter, all over this class of tiles.

Example 5.5.1. If we choose the palindromes $k = 010\overline{1}0\overline{1}010$ and $\overline{p} = 0\overline{1}0\overline{1}0\overline{1}0$, we get the boundary word of the polyomino in Figure 5.7,

$$P = 1 \vdots 010\overline{1}0\overline{1}0101 \vdots 0 \vdots \overline{1}0\overline{1}0\overline{1}0\overline{1}0 | \overline{1} \vdots \overline{010}\overline{1}0\overline{1}0\overline{1010\overline{1}} \vdots \overline{0} \vdots 1\overline{0}1\overline{0}1\overline{0}1\overline{0}.$$

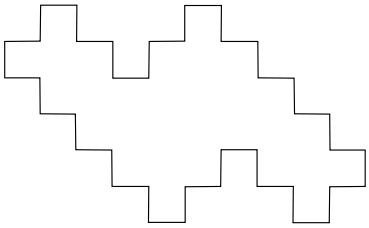


Figure 5.7: The prime double square in $\mathcal{A}_0$ obtained by choosing the palindromes $k = 010\overline{1}0\overline{1}010$ and $\overline{p} = 0\overline{1}0\overline{1}0\overline{1}0$.

General case $\mathcal{A}$

To obtain the boundary word in the general case of the class $\mathcal{A}$, it is sufficient to let the values n_1 and n_3 vary, thus getting

$$P_{\mathcal{A}} = (1k1p)^{n_1}1 \vdots k1 \vdots (\overline{p}\overline{1}k1)^{n_3}0 \vdots \overline{1}0.$$

The smallest tiles in the class $\mathcal{A} \setminus \mathcal{A}_0$, having semi-perimeter $\ell = 10$, are obtained by choosing $k = \overline{p} = 0$ and then $n_1 = 1$ and $n_3 = 0$, or vice versa. They are depicted in Fig. 5.8. Notice that both tiles can be interpreted as a slightly variation of the cross, that is somehow doubled in the vertical and horizontal direction, respectively.

Example 5.5.2. With reference to Example 5.5.1, choosing again the palindromes $k = 010\overline{1}0\overline{1}010$ and $\overline{p} = 0\overline{1}0\overline{1}0\overline{1}0$, and the exponents $n_1 = 1$ and $n_3 = 2$, we get a prime double square in the class $\mathcal{A} \setminus \mathcal{A}_0$, see Fig. 5.9.
Its boundary word is

$$\begin{aligned} P \;=\; & 1010\overline{1}0\overline{1}0101\overline{0}1\overline{0}1\overline{0}1\overline{0}1 \vdots 010\overline{1}0\overline{1}0101 \vdots (0\overline{1}0\overline{1}0\overline{1}0\overline{1}010\overline{1}0\overline{1}0101)^2 0 \vdots \overline{1}0\overline{1}0\overline{1}0\overline{1}0 | \\ & \overline{1010\overline{1}0\overline{1}010\overline{1}0\overline{1}0\overline{1}0\overline{1}} \vdots \overline{010\overline{1}0\overline{1}010\overline{101}} \vdots (\overline{0}1\overline{0}1\overline{0}1\overline{0}1\overline{0}1\overline{0}1\overline{0}1\overline{0}1\overline{0}1\overline{0}1)^2 \overline{0} \vdots 1\overline{0}1\overline{0}1\overline{0}1\overline{0}. \end{aligned}$$

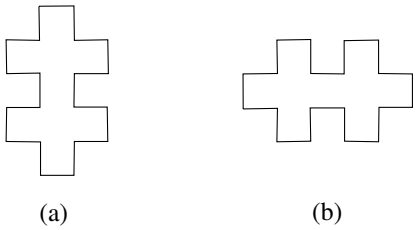


Figure 5.8: The smallest (w.r.t. semi-perimeter) prime double squares in the class $\mathcal{A} \setminus \mathcal{A}_0$. The polyomino in (a) is obtained by choosing $n_1 = 1$ and $n_3 = 0$, while the polyomino in (b) with $n_1 = 0$ and $n_3 = 1$.

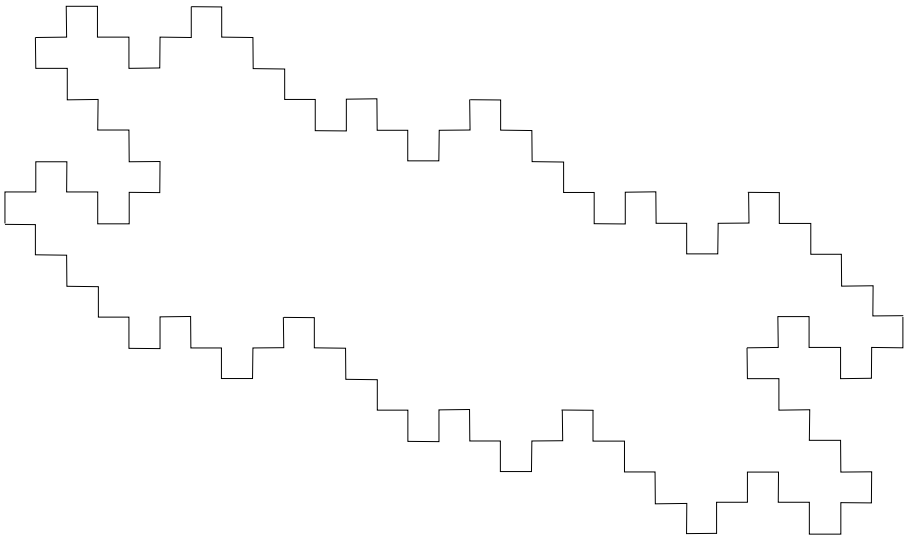


Figure 5.9: The prime double square in $\mathcal{A}$ obtained by choosing the palindromes $k = 010\overline{1}0\overline{1}010$ and $\overline{p} = 0\overline{1}0\overline{1}0\overline{1}0$, and the exponents $n_1 = 1$ and $n_3 = 2$.

5.5.2 Class $\mathcal{B}$

The second class is given by all those prime double squares having factor $u_1 = 1$ and such that $|u_3| > 1$. In this case, we consider the boundary word P factorized as in (5.6), where u_3 is highlighted and so easier to study:

$$P = (tu_3s\widehat{u}_3)^{n_1}1 \vdots \tilde{u}_3t \vdots (u_3\overline{s}\tilde{u}_3t)^{n_3}u_3 \vdots \overline{s}\tilde{u}_3|$$

Similarly to case $\mathcal{A}$, we want to find the possible factors t and $\overline{s}$ that give rise to a prime double square having boundary word P.

Basic case $\mathcal{B}_0$

We first consider the case $n_1 = n_3 = 0$.
The BN-factors of the tile are $A = 1\tilde{u}_3t$ and $Y = \overline{s}\tilde{u}_3\overline{1}$. Moreover, the boundary word contains $\tilde{u}_3\overline{1}\widehat{u}_3$ as a factor, meaning that the second letter of u_3 must be $\overline{1}$, to avoid boundary intersections. With reference to Theorem 5.4.27, we find out that the factor is equal to $u_3 = \Big((0\overline{1})^\alpha(01)^\beta\Big)^\gamma 0$, for some $\alpha, \beta, \gamma \geq 1$, and so that the (half) boundary word of the polyomino is of type

$$P = 1 \vdots 0\Big((10)^\beta(\overline{1}0)^\alpha\Big)^\gamma t \vdots \Big((0\overline{1})^\alpha(01)^\beta\Big)^\gamma 0 \vdots \overline{s}0\Big((10)^\beta(\overline{1}0)^\alpha\Big)^\gamma.$$

By Theorem 5.4.23, $\overline{1} \notin t, s$, and so we immediately argue, by imposing the palindromicity of A and Y, that $t = (10)^\beta 1$ and $\overline{s} = (\overline{1}0)^\alpha\overline{1}$. Notice that the exponents α and β are the same appearing in the definition of u_3. By varying their values, together with γ, all the tiles in this class are generated. Their (half) boundary word is, in general,

$$P_{\mathcal{B}_0} = 1 \vdots 0\Big((10)^\beta(\overline{1}0)^\alpha\Big)^\gamma(10)^\beta 1 \vdots \Big((0\overline{1})^\alpha(01)^\beta\Big)^\gamma 0 \vdots (\overline{1}0)^\alpha\overline{1}0\Big((10)^\beta(\overline{1}0)^\alpha\Big)^\gamma.$$

The smallest polyomino in this class, in Figure 5.10, has semi-perimeter $\ell = 22$, and it is given by the choice of $\alpha = \beta = \gamma = 1$. Its boundary word is

$$P = 1 \vdots 010\overline{1}0101 \vdots 0\overline{1}010 \vdots \overline{1}0\overline{1}010\overline{1}0|\overline{1} \vdots \overline{010}1\overline{010}\overline{1} \vdots \overline{0}1\overline{010} \vdots 1\overline{0}1\overline{010}1\overline{0}.$$

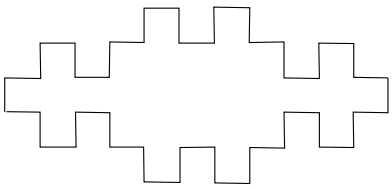


Figure 5.10: The smallest (w.r.t. semi-perimeter) prime double square in the class $\mathcal{B}_0$.

General case $\mathcal{B}$

To obtain the boundary word in the general case, it is sufficient to let the exponents n_1 and n_3 vary, thus getting

$$\begin{aligned} P_{\mathcal{B}} \;=\; & \Big[(10)^\beta 1\Big((0\overline{1})^\alpha(01)^\beta\Big)^\gamma 0(1\overline{0})^\alpha 1\overline{0}\Big((\overline{1}0)^\beta(1\overline{0})^\alpha\Big)^\gamma\Big]^{n_1} 1\vdots \\ & 0\Big((10)^\beta(\overline{1}0)^\alpha\Big)^\gamma(10)^\beta 1\vdots \\ & \Big[\Big((0\overline{1})^\alpha(01)^\beta\Big)^\gamma 0(\overline{1}0)^\alpha\overline{1}0\Big((10)^\beta(\overline{1}0)^\alpha\Big)^\gamma(10)^\beta 1\Big]^{n_3}\Big((0\overline{1})^\alpha(01)^\beta\Big)^\gamma 0\vdots \\ & (\overline{1}0)^\alpha\overline{1}0\Big((10)^\beta(\overline{1}0)^\alpha\Big)^\gamma. \end{aligned}$$

The smallest tiles in the class $\mathcal{B}\setminus\mathcal{B}_0$, both of semi-perimeter $\ell = 38$, are obtained by choosing $\alpha = \beta = \gamma = 1$ and then $n_1 = 1$ and $n_3 = 0$, or vice versa. They are both depicted in Fig. 5.11. Notice that both tiles can be interpreted as a slightly variation of the smallest tile in $\mathcal{B}_0$, that is somehow doubled in the vertical and horizontal direction, respectively.

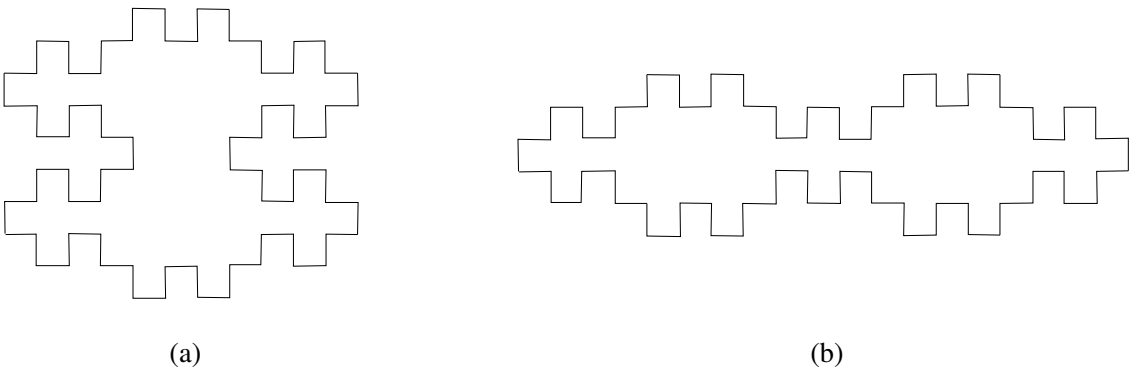


Figure 5.11: The smallest (w.r.t. semi-perimeter) prime double squares in the class $\mathcal{B}\setminus\mathcal{B}_0$. The polyomino in (a) is obtained by choosing $n_1 = 1$ and $n_3 = 0$, while the polyomino in (b) with $n_1 = 0$ and $n_3 = 1$.

5.5.3 Class $\mathcal{C}$

Finally, we consider all those prime double squares such that the factor u_1 has length strictly greater than one, and denote this class of tiles as $\mathcal{C}$. In such a case, both the standard factorizations can be used to study the boundary word. We decide to write it as in (5.5).

As we did for the other cases, we first arbitrarily choose the factor u_1, then consider the possible factors u_3 in accordance with the previous choice (Theorem 5.4.27), and finally determine k and $\overline{p}$ uniquely. We remark that we have to impose the condition $|u_1| \leq |u_3|$.

Basic case $\mathcal{C}_0$

We first consider the case $n_1 = n_3 = 0$.
In accordance to Theorem 5.4.27, we can choose the second letter of the factor u_1, that may be equal to 0 or $\overline{0}$, indifferently. So, we further divide the class $\mathcal{C}$ in the disjoint subclasses $\mathcal{C}_1$ and $\mathcal{C}_2$, as well as $\mathcal{C}_{01}$ and $\mathcal{C}_{02}$ for the basic cases inside each of them.

Class $\mathcal{C}_{01}$. Let be $u_1 = \Big((10)^\alpha(1\overline{0})^\beta\Big)^\gamma 1$, with $\alpha, \beta, \gamma \geq 1$.

We have $X = k\tilde{u}_1 u_3 = k\Big((1\overline{\mathbf{0}})^\beta(10)^\alpha\Big)^\gamma 1 u_3$ palindrome (Theorem 5.1.4), k palindrome, and $\overline{0} \notin u_3, k$ (Theorem 5.4.23). As a consequence, we have to choose u_3 of type $u_3 = \Big((0\overline{1})^\beta(01)^\alpha\Big)^\delta 0$, for some $\delta \geq \gamma$. Notice that the exponents $\alpha, \beta \geq 1$ both define u_1 and u_3.

At this point, it is sufficient to impose $X = k\tilde{u}_1 u_3$ and $B = u_3\widehat{u}_1\overline{p}$ to be palindrome to obtain $k = (01)^\alpha u_3$ and $\overline{p} = u_3(\overline{1}0)^\beta$, and so the (half) boundary word of a prime double square in this class:

$$\begin{aligned} P_{\mathcal{C}_{01}} &= \Big((10)^\alpha(1\overline{0})^\beta\Big)^\gamma 1 \vdots (01)^\alpha\Big((0\overline{1})^\beta(01)^\alpha\Big)^\delta 01\Big((\overline{0}1)^\beta(01)^\alpha\Big)^\gamma \vdots \\ &\quad \Big((0\overline{1})^\beta(01)^\alpha\Big)^\delta 0 \vdots \overline{1}\Big((0\overline{1})^\beta(\overline{01})^\alpha\Big)^\gamma\Big((0\overline{1})^\beta(01)^\alpha\Big)^\delta 0(\overline{1}0)^\beta. \end{aligned}$$

The tile with shortest semi-perimeter in this class, $\ell = 34$, is obtained by choosing $\alpha = \beta = \gamma = \delta = 1$, that is,

$$\begin{aligned} u_1 &= 101\overline{0}1 \\ u_3 &= 0\overline{1}010 \\ k &= 010\overline{1}010 \\ \overline{p} &= 0\overline{1}010\overline{1}0. \end{aligned}$$

Its boundary word is

$$\begin{aligned} P &= 101\overline{0}1 \vdots 010\overline{1}0101\overline{0}101 \vdots 0\overline{1}010 \vdots \overline{1}0\overline{101}0\overline{1}010\overline{1}0| \\ &\quad \overline{101}0\overline{1} \vdots \overline{010}1\overline{010}1\overline{010}\overline{101} \vdots \overline{0}1\overline{010} \vdots 1\overline{0}101\overline{0}1\overline{010}1\overline{0}, \end{aligned}$$

and it is depicted in Fig. 5.12(a).

Class $\mathcal{C}_{02}$. Let be $u_1 = \Big((1\overline{0})^\alpha(10)^\beta\Big)^\gamma 1$, with $\alpha, \beta, \gamma \geq 1$.

Following the same reasoning as in the previous case, we obtain that u_3 is of type $u_3 = \Big((01)^\beta(0\overline{1})^\alpha\Big)^\delta 0$, for some $\delta \geq \gamma$, and $k = (01)^{-\beta}u_3$, $\overline{p} = u_3(\overline{1}0)^{-\alpha}$, with α, β the same exponents in u_1 and u_3[2].

[2]We use the notation $k = (01)^{-\beta}u_3$ to point out that k is obtained after removing the prefix $(01)^\beta$ from u_3. Since $\beta \geq 1$ by hypothesis, this notation is always well defined. The same holds for the word $\overline{p}$.

Finally, the (half) boundary word of a prime double square in this class is

$$\begin{aligned} P_{\mathcal{C}_{02}} &= \Big((1\overline{0})^{\alpha}(10)^{\beta}\Big)^{\gamma}1\,\vdots\,(01)^{-\beta}\Big((01)^{\beta}(0\overline{1})^{\alpha}\Big)^{\delta}01\Big((01)^{\beta}(\overline{0}1)^{\alpha}\Big)^{\gamma}\,\vdots \\ &\quad \Big((01)^{\beta}(0\overline{1})^{\alpha}\Big)^{\delta}0\,\vdots\,\overline{1}\Big((\overline{01})^{\beta}(0\overline{1})^{\alpha}\Big)^{\gamma}\Big((01)^{\beta}(0\overline{1})^{\alpha}\Big)^{\delta}0(\overline{1}0)^{-\alpha}. \end{aligned}$$

The tile with shortest semi-perimeter in this class, $\ell = 26$, is obtained by choosing $\alpha = \beta = \gamma = \delta = 1$, that is,

$$\begin{aligned} u_1 &= 1\overline{0}101 \\ u_3 &= 010\overline{1}0 \\ k &= 0\overline{1}0 \\ \overline{p} &= 010. \end{aligned}$$

Its boundary word is

$$P = 1\overline{0}101\,\vdots\,0\overline{1}010\overline{1}01\,\vdots\,010\overline{1}0\,\vdots\,\overline{101}0\overline{1}010|\overline{1}0\overline{101}\,\vdots\,\overline{0}1\overline{010}\overline{1}01\,\vdots\,\overline{010}1\overline{0}\,\vdots\,101\overline{0}1\overline{010},$$

and it is depicted in Fig. 5.12(b).

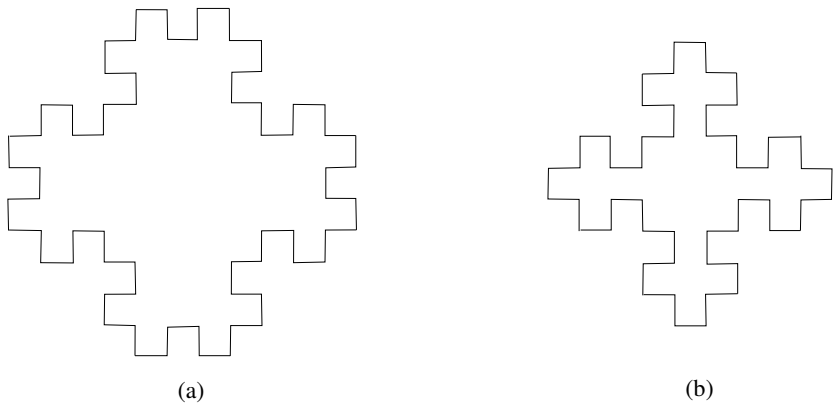


Figure 5.12: The smallest (w.r.t. semi-perimeter) prime double squares in the class $\mathcal{C}_{01}$, on the left, and $\mathcal{C}_{02}$, on the right.

General case $\mathcal{C}$

We conclude by providing the (half) boundary word in the general case, that is, allowing the values $n_1, n_3 \geq 0$ to vary.

Class $\mathcal{C}_1$. We have

$$\begin{aligned}
P_{\mathcal{C}_1} \;=\; & \Big[\Big((10)^\alpha(1\bar{0})^\beta\Big)^\gamma 1(01)^\alpha\Big((0\bar{1})^\beta(01)^\alpha\Big)^\delta 01\Big((\bar{0}1)^\beta(01)^\alpha\Big)^\gamma\Big((\bar{0}1)^\beta(\overline{01})^\alpha\Big)^\delta\bar{0}(1\bar{0})^\beta\Big]^{n_1}\Big((10)^\alpha(1\bar{0})^\beta\Big)^\gamma 1\,\vdots\\
& (01)^\alpha\Big((0\bar{1})^\beta(01)^\alpha\Big)^\delta 01\Big((\bar{0}1)^\beta(01)^\alpha\Big)^\gamma\,\vdots\\
& \Big[\Big((0\bar{1})^\beta(01)^\alpha\Big)^\delta 0(\bar{1}0)^\beta\Big((\overline{10})^\alpha(\bar{1}0)^\beta\Big)^\gamma\bar{1}(01)^\alpha\Big((0\bar{1})^\beta(01)^\alpha\Big)^\delta 01\Big((\bar{0}1)^\beta(01)^\alpha\Big)^\gamma\Big]^{n_3}\Big((0\bar{1})^\beta(01)^\alpha\Big)^\delta 0\,\vdots\\
& \bar{1}\Big((0\bar{1})^\beta(\overline{01})^\alpha\Big)^\gamma\Big((0\bar{1})^\beta(01)^\alpha\Big)^\delta 0(\bar{1}0)^\beta,
\end{aligned}$$

and the smallest polyominoes in the class $\mathcal{C}_1 \setminus \mathcal{C}_{01}$, depicted in Fig. 5.13, have boundary words

$$\begin{aligned}
P_{(a)} \;=\; & 101\bar{0}1010\bar{1}0101\bar{0}101\bar{0}1\overline{010}1\bar{0}101\bar{0}1\,\vdots\,010\bar{1}0101\bar{0}101\,\vdots\,0\bar{1}010\,\vdots\,\bar{1}0\overline{101}0\bar{1}010\bar{1}0|\\
& \overline{101}0\overline{101}0\overline{101}0\overline{101}0\overline{101}0\bar{1}010\bar{1}0\overline{101}0\bar{1}\,\vdots\,\overline{010}1\overline{010}1\overline{010}\overline{101}\,\vdots\,\bar{0}1\overline{010}\,\vdots\,1\bar{0}101\bar{0}1\overline{010}1\bar{0}
\end{aligned}$$

and

$$\begin{aligned}
P_{(b)} \;=\; & 101\bar{0}1\,\vdots\,010\bar{1}0101\bar{0}101\,\vdots\,0\bar{1}010\bar{1}0\overline{101}0\bar{1}010\bar{1}0101\bar{0}1010\bar{1}010\,\vdots\,\bar{1}0\overline{101}0\bar{1}010\bar{1}0|\\
& \overline{101}0\bar{1}\,\vdots\,\overline{010}1\overline{010}1\overline{010}\overline{101}\,\vdots\,\bar{0}1\overline{010}1\bar{0}101\bar{0}1\overline{010}1\overline{010}1\overline{010}1\overline{010}1\overline{010}\,\vdots\,1\bar{0}101\bar{0}1\overline{010}1\bar{0}.
\end{aligned}$$

They both have semi-perimeter $\ell = 58$.

Class $\mathcal{C}_2$. We have

$$\begin{aligned}
P_{\mathcal{C}_2} \;=\; & \Big[\Big((1\bar{0})^\alpha(10)^\beta\Big)^\gamma 1(01)^{-\beta}\Big((01)^\beta(0\bar{1})^\alpha\Big)^\delta 01\Big((01)^\beta(\bar{0}1)^\alpha\Big)^\gamma\Big((\bar{0}1)^\beta(\bar{0}1)^\alpha\Big)^\delta\bar{0}(1\bar{0})^{-\alpha}\Big]^{n_1}\Big((1\bar{0})^\alpha(10)^\beta\Big)^\gamma 1\,\vdots\\
& (01)^{-\beta}\Big((01)^\beta(0\bar{1})^\alpha\Big)^\delta 01\Big((01)^\beta(\bar{0}1)^\alpha\Big)^\gamma\,\vdots\\
& \Big[\Big((01)^\beta(0\bar{1})^\alpha)\Big)^\delta 0(\bar{1}0)^{-\alpha}\Big((\bar{1}0)^\alpha(\overline{10})^\beta\Big)^\gamma\bar{1}(01)^{-\beta}\Big((01)^\beta(0\bar{1})^\alpha\Big)^\delta 01\Big((01)^\beta(\bar{0}1)^\alpha\Big)^\gamma\Big]^{n_3}\Big((01)^\beta(0\bar{1})^\alpha\Big)^\delta 0\,\vdots\\
& \bar{1}\Big((\overline{01})^\beta(0\bar{1})^\alpha\Big)^\gamma\Big((01)^\beta(0\bar{1})^\alpha\Big)^\delta 0(\bar{1}0)^{-\alpha},
\end{aligned}$$

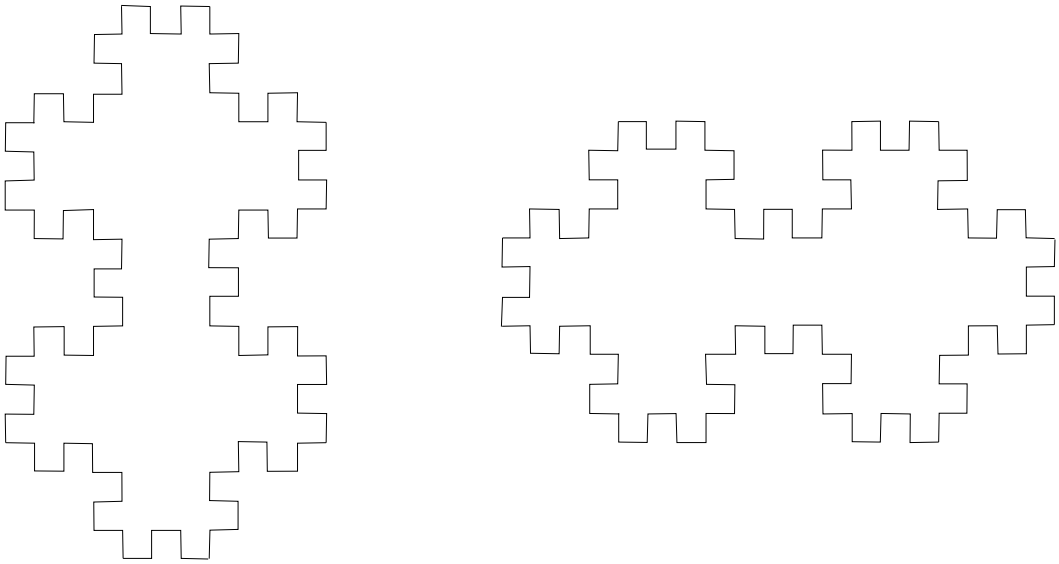
and the smallest polyominoes in the class $\mathcal{C}_2 \setminus \mathcal{C}_{02}$, depicted in Fig. 5.14, have boundary words

$$\begin{aligned}
P_{(a)} \;=\; & 1\bar{0}1010\bar{1}0101\bar{0}1\overline{010}1\bar{0}101\,\vdots\,0\bar{1}0101\bar{0}1\,\vdots\,010\bar{1}0\,\vdots\,\overline{101}0\bar{1}010|\\
& \bar{1}0\overline{101}0\overline{101}0\overline{101}0\bar{1}010\bar{1}0\overline{101}\,\vdots\,\bar{0}1\overline{010}1\overline{010}\bar{1}\,\vdots\,\overline{010}1\bar{0}\,\vdots\,101\bar{0}1\overline{010}
\end{aligned}$$

and

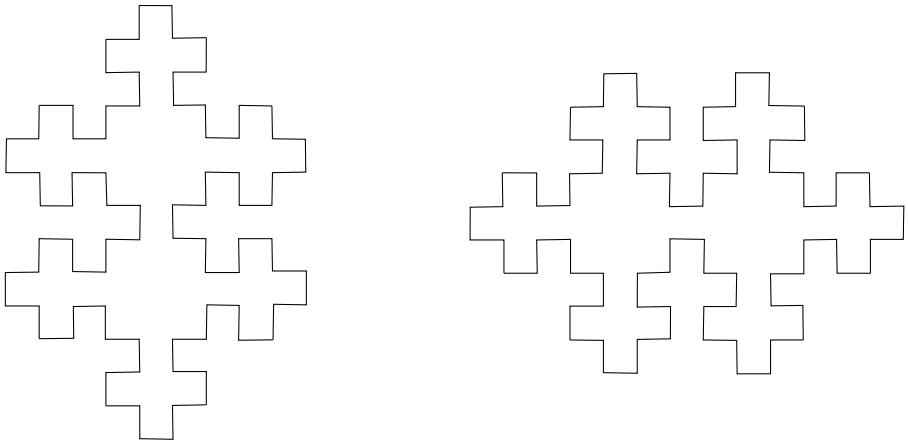
$$\begin{aligned}
P_{(b)} \;=\; & 1\bar{0}101\,\vdots\,0\bar{1}0101\bar{0}1\,\vdots\,010\bar{1}0\overline{101}0\bar{1}0101\bar{0}1010\bar{1}0\,\vdots\,\overline{101}0\bar{1}010|\\
& \bar{1}0\overline{101}\,\vdots\,\bar{0}1\overline{010}1\overline{010}\bar{1}\,\vdots\,\overline{010}1\bar{0}101\bar{0}1\overline{010}1\overline{010}1\overline{010}1\bar{0}\,\vdots\,101\bar{0}1\overline{010}.
\end{aligned}$$

They both have semi-perimeter $\ell = 42$.



(a) (b)

Figure 5.13: The smallest (w.r.t. semi-perimeter) prime double squares in the class $\mathcal{C}_1 \setminus \mathcal{C}_{01}$. The polyomino in (a) is obtained by choosing $n_1 = 1$ and $n_3 = 0$, while the polyomino in (b) with $n_1 = 0$ and $n_3 = 1$.



(a) (b)

Figure 5.14: The smallest (w.r.t. semi-perimeter) prime double squares in the class $\mathcal{C}_2 \setminus \mathcal{C}_{02}$. The polyomino in (a) is obtained by choosing $n_1 = 1$ and $n_3 = 0$, while the polyomino in (b) with $n_1 = 0$ and $n_3 = 1$.

5.6 Enumeration of the class $\mathcal{P}^0$

We conclude the chapter with some comments about the enumeration of prime double square tiles with respect to the semi-perimeter, that are a direct consequence of the generation procedure we presented in Section 5.5. We underline that the procedure generates exactly each prime double square in the class $\mathcal{P}^0$, and only once. We sketch the main enumeration results, and we address the reader to our work [11] for a more detailed description.

The whole class of prime double squares is given by the union of the four distinct subclasses we described in the previous section, $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}_1$ and $\mathcal{C}_2$. For each of them, we consider the generating function according to the semi-length of the boundary word, addressed as

$$\begin{aligned} A(x) &= \textstyle\sum_{n\geq 0} a_n x^n, \\ B(x) &= \textstyle\sum_{n\geq 0} b_n x^n, \\ C_1(x) &= \textstyle\sum_{n\geq 0} c_n^1 x^n, \\ C_2(x) &= \textstyle\sum_{n\geq 0} c_n^2 x^n, \end{aligned}$$

respectively, where a_n (b_n, c_n^1, c_n^2, respectively) is the number of prime double squares in the class $\mathcal{A}$ ($\mathcal{B}$, $\mathcal{C}_1$, $\mathcal{C}_2$, respectively) having semi-perimeter equal to n. The generating function of the whole class $\mathcal{P}^0$ will be simply obtained as their sum, that is,

$$P(x) = A(x) + B(x) + C_1(x) + C_2(x).$$

An explicit expression for the generating function $P(x) = \sum_{n\geq 0} p_n x^n$ is hard to determine, but we are able to provide the first terms, as well as the asymptotic behavior of the sequence p_{2m+6}, for $m \geq 2$.

$$\begin{aligned} P(x) \quad = \quad & x^6 + 2x^8 + 5x^{10} + 6x^{12} + 16x^{14} + 16x^{16} + 34x^{18} + 46x^{20} + 78x^{22} + \\ & 96x^{24} + 186x^{26} + 224x^{28} + 377x^{30} + 540x^{32} + 838x^{34} + 1154x^{36} + \\ & 1862x^{38} + 2560x^{40} + 4011x^{42} + 5694x^{44} + 8660x^{46} + 12292x^{48} + \mathrm{O}\big(x^{50}\big)\,. \end{aligned}$$

Example 5.6.1. There is a unique prime double square having semi-perimeter equal to $\ell = 6$, represented by the term x^6 in $P(x)$, that is the cross (class $\mathcal{A}_0$).
Two prime double squares of semi-perimeter $\ell = 8$ exist, both in $\mathcal{A}_0$ (see Fig. 5.15).

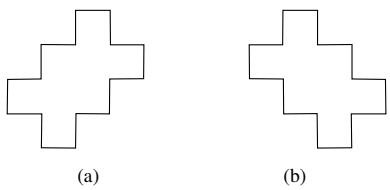


Figure 5.15: The prime double squares of semi-perimeter $\ell = 8$, all in the class $\mathcal{A}_0$. The tile in (a) is obtained by choosing $k = 010$ and $\overline{p} = 0$, while the tile in (b) with $k = 0$ and $\overline{p} = 0\overline{1}0$.

The five prime double squares of semi-perimeter $\ell = 10$ are described in Fig. 5.16, and they all belong to the class $\mathcal{A}$. More in detail:

a) $P_{(a)} = 10101010\overline{1}0 \in \mathcal{A}_0$, choosing $k = 01010$ and $\overline{p} = 0$.

b) $P_{(b)} = 101010\overline{1}0\overline{1}0 \in \mathcal{A}_0$, choosing $k = 010$ and $\overline{p} = 0\overline{1}0$.

c) $P_{(c)} = 1010\overline{1}0\overline{1}0\overline{1}0 \in \mathcal{A}_0$, choosing $k = 0$ and $\overline{p} = 0\overline{1}0\overline{1}0$.

d) $P_{(d)} = (101\overline{0})^1 1010\overline{1}0 \in \mathcal{A}$, choosing $k = \overline{p} = 0$, $n_1 = 1$ and $n_3 = 0$.

e) $P_{(e)} = 101(0\overline{1}01)^1 0\overline{1}0 \in \mathcal{A}$, choosing $k = \overline{p} = 0$, $n_1 = 0$ and $n_3 = 1$.

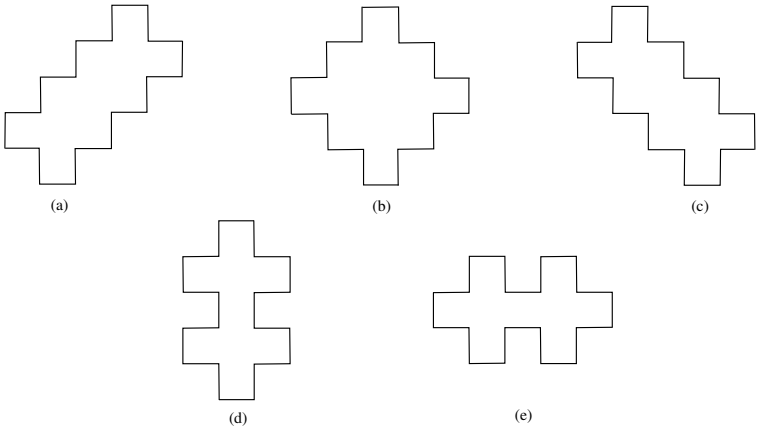


Figure 5.16: The prime double squares of semi-perimeter $\ell = 10$, all belonging to the class $\mathcal{A}$. The ones in the first row also belong to the class $\mathcal{A}_0$.

Theorem 5.6.2 ([11]). For $m \geq 2$, we have:

i) $p_{2m+6} \in \mathcal{O}(m2^{\frac{m}{2}})$.

ii) $\lim_{m\to\infty} \frac{b_m + c_m}{a_m} = 0$, where $c_m = c_m^1 + c_m^2$.

It is noticeable to observe that, with the semi-perimeter growing, the subclass $\mathcal{A}$ acquires a predominant role on the other subclasses, whose size becomes infinitely smaller.

Conclusion and final remarks

As a conclusion of the chapter, we propose a short list of possible research lines that arise from our work.

- ▷ We completely characterized the boundary words of prime double square tiles, that now can be quickly detected just looking for a factorization of the word as in (5.5), and then verifying if the involved factors u_1, u_3, k and p satisfy the required

conditions. As a consequence, we can look for special classes of tiles that could be subclasses of prime double squares, such as Christoffel and Fibonacci tiles ([18]).

Problem 1. Prove that Christoffel and Fibonacci tiles are prime double squares, as conjectured by the authors in [18].

▷ When looking at prime double squares in the classes $\mathcal{A} \setminus \mathcal{A}_0$, $\mathcal{B} \setminus \mathcal{B}_0$ and $\mathcal{C} \setminus \mathcal{C}_0$, we observe from the examples that these tiles are somehow obtained by gluing copies of tiles in the classes $\mathcal{A}_0$, $\mathcal{B}_0$ and $\mathcal{C}_0$, vertically or horizontally, according to the (strictly positive) values of the exponents n_1 and n_3 that appear in their boundary word.

Problem 2. Study the role of the parameters n_1 and n_3 in the boundary word of a prime double square. Provide a geometrical interpretation of them, in terms of the vertical and horizontal development of the tiles in $\mathcal{A}$ ($\mathcal{B}, \mathcal{C}$, respectively) w.r.t. the corresponding tile in $\mathcal{A}_0$ ($\mathcal{B}_0, \mathcal{C}_0$, respectively).

▷ We worked under the strong hypothesis that no occurrences of the letter $\overline{1}$ are present in the factor u_1, as well as no $\overline{0}s$ appear in the factor u_3. We completely generated and enumerated the prime double squares related to this class of tiles, but a general result is still missing.

Problem 3. Characterize the factors u_1 and u_3 in terms of the possible number of occurrences and positions of the letters $\overline{1}$ and $\overline{0}$, respectively, and complete the generation of the whole class of prime double square polyominoes.

▷ After the complete characterization of prime double squares, the whole class of double squares can be studied (and characterized) by exploiting our results. Indeed, prime double squares can be interpreted as the seed of the generation of all double square tiles, that can always be obtained from them using homologous morphisms.

Problem 4. Characterize and enumerate all double squares by exploiting the notion of homologous morphism and the results we reached for prime double squares, as well as possible bijections with well-known combinatorial structures.

Bibliography

[1] N. Alon, J. Spencer, *The Probabilistic Method, 4th edition*, John Wiley & Sons, Hoboken, NJ (2016)

[2] M. Ascolese, P. Dulio, S.M.C. Pagani, *Uniqueness and reconstruction of finite lattice sets from their line sums*, Discrete Appl. Math. 356, 293-306 (2024)

[3] M. Ascolese, P. Dulio, S.M.C. Pagani, *Some geometric and tomographic results on gray-scale images*, Lecture Notes in Computer Science Vol 14605 (2024)

[4] M. Ascolese, A. Frosini, *On the decomposability of homogeneous binary planar configurations with respect to a given exact polyomino*, Lecture Notes in Comput. Sci. 13493, 139-152 (2022)

[5] M. Ascolese, A. Frosini, *Characterization and reconstruction of hypergraphic pattern sequences*, Combinatorial image analysis, 303-316, Lecture Notes in Comput. Sci. 13348 (2023)

[6] M. Ascolese, A. Frosini, *Proving a conjecture on prime double square tiles*, Discrete Appl. Math. 350, 71-83 (2024)

[7] M. Ascolese, A. Frosini, *Setting the path to the combinatorial characterization of prime double square polyominoes*, In: Italian Conference on Theoretical Computer Science 2023, CEUR Workshop Proceedings, Vol. 3587, 157-168 (2023)

[8] M. Ascolese, A. Frosini, W.L. Kocay, L. Tarsissi, *Properties of unique degree sequences of 3-uniform hypergraphs*, Discrete geometry and mathematical morphology, 312-324, Lecture Notes in Comput. Sci. 12708 (2021)

[9] M. Ascolese, A. Frosini, E. Pergola, S. Rinaldi, *A Heuristic for the P-time Reconstruction of Unique 3-Uniform Hypergraphs from their Degree Sequences*, In: Italian Conference on Theoretical Computer Science 2023, CEUR Workshop Proceedings, Vol-3587, 77-91 (2023)

[10] M. Ascolese, A. Frosini, E. Pergola, S. Rinaldi, L. Vuillon, *An algebraic approach to the reconstruction of uniform hypergraphs from their degree sequence*, Theor. Comput. Sci. 1020 (2024)

[11] M. Ascolese, A. Frosini, S. Rinaldi, *Generation and enumeration of prime double square polyominoes* (submitted to Theoretical Computer Science)

[12] M. Ascolese, M. Lienau, M. Schulte, A. Taraz, *Randomized algorithms to generate hypergraphs with given degree sequences* (submitted to Electronic Journal of Combinatorics)

[13] K.J. Batenburg, W. Fortes, L. Hajdu, R. Tijdeman, *Bounds on the quality of reconstructed images in binary tomography*, Discrete Appl. Math. 161(15), 2236–2251 (2013)

[14] D. Battaglino, A. Frosini, S. Rinaldi, *A decomposition theorem for homogeneous sets with respect to diamond probes*, Comput. Vis. Image Underst. 17, 319-325 (2013)

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

[16] S. Behrens, C. Erbes, M. Ferrara, S.G. Hartke, B. Reiniger, H. Spinoza, C. Tomlinson, *New results on degree sequences of uniform hypergraphs*, Electron. J. Combin. 20(4) (2013)

[17] D. Billington, *Lattices and degree sequences of uniform hypergraphs*, Ars Combin. 21A, 9-19 (1986)

[18] A. Blondin Massé, S. Brlek, A. Garon, S. Labbé, *Two infinite families of polyominoes that tile the plane by translation in two distinct ways*, Theor. Comput. Sci. 412, 4778-4786 (2011)

[19] A. Blondin Massé, S. Brlek, S. Labbé, *Palindromic lacunas of the Thue-Morse word*, in: GASCom, Int. Conf. on Random Generation of Combinatorial Structures, 53-67 (2008)

[20] A. Blondin Massé, S. Brlek, S. Labbé, *A parallelogram tile fills the plane by translation in at most two distinct ways*, Discret. Appl. Math. 170(7-8), 1011-1018 (2021)

[21] A. Blondin Massé, A. Garon, S. Labbé, *Combinatorial properties of double square tiles*, Theor. Comput. Sci. 502, 98-117 (2013)

[22] B. Bollobás, *Random Graphs, 2nd edition*, Cambridge Studies in Advanced Mathematics 73, Cambridge University Press (2001)

[23] S. Brlek, A. Frosini, *A tomographical interpretation of a sufficient condition on h-graphical sequences*, Discrete geometry for computer imagery, 95-104, Lecture Notes in Comput. Sci. 9647 (2016)

[24] S. Brlek, X. Provençal, J.M. Fédou, *On the tiling by translation problem*, Discrete Applied Mathematics 157, 464-475 (2009)

[25] S. Brunetti, P. Dulio, C. Peri, *Characterization of* $\{-1, 0, +1\}$ *valued functions in discrete tomography under sets of four directions*, Lecture Notes in Comput. Sci. 6607, 394–405 (2011)

[26] S. Brunetti, P. Dulio, C. Peri, *Discrete tomography determination of bounded lattice sets from four X-rays*, Discrete Appl. Math. 161(15), 2281–2292 (2013)

[27] S. Brunetti, P. Dulio, C. Peri, *Discrete tomography determination of bounded sets in* $\mathbb{Z}^n$, Discrete Appl. Math. 183, 20–30 (2015)

[28] S. Brunetti, P. Dulio, C. Peri, *Uniqueness results for grey scale digital images*, Fund. Inform. 172(2), 221–238 (2020)

[29] C. Cooper, *The cores of random hypergraphs with a given degree sequence*, Random Structures Algorithms 25(4), 353-375 (2004)

[30] C. Cooper, A. Frieze, M. Molloy, B. Reed, *Perfect matchings in random r-regular, s-uniform hypergraphs*, Combin. Probab. Comput. 5(1), 1-14 (1996)

[31] S. Choudum, *On graphic and* 3*-graphic sequences*, Discrete Math. 87(1), 91-95 (1991)

[32] B. van Dalen, L. Hajdu, R. Tijdeman, *Bounds for discrete tomography solutions*, Indagationes Mathematicae 24, 391-402 (2013)

[33] A. Deza, A. Levin, S.M. Meesum, S. Onn, *Optimization over degree sequences*, SIAM J. Discrete Math. 32(3), 2067-2079 (2018)

[34] P. Dulio, S.M.C. Pagani, *A rounding theorem for unique binary tomographic reconstruction*, Discrete Appl. Math. 268, 54-69 (2019)

[35] P. Dulio, S.M.C. Pagani, Erratum to *"A rounding theorem for unique binary tomographic reconstruction"*, Discrete Appl. Math. (https://doi.org/10.1016/j.dam.2024.05.031)

[36] A.K. Dewdney, *Degree sequences in complexes and hypergraphs*, Proc. Amer. Math. Soc. 53(2), 535-540 (1975)

[37] M. Dyer, C. Greenhill, P. Kleer, J. Ross, L. Stougie, *Sampling hypergraphs with given degree sequence*, Discrete Math. 344(11), 112566 (2021)

[38] P. Erdős, T. Gallai, *Graphs with prescribed degrees of vertices* (in Hungarian), Mat. Lapok 11, 264-274 (1960)

[39] P. Erdős, A. Rényi, V.T. Sós, *On a problem of graph theory*, Studia Sci. Math. Hungar. 1, 215–235 (1966)

[40] H. Freeman, *Boundary encoding and processing*, B. Lipkin, A. Rosenfeld (Eds.), Picture Processing and Psychopictorics, Academic Press, New York, 241-266 (1970)

[41] H. Freeman, *On the encoding of arbitrary geometric configurations*, IRE Trans. Electron. Comput. 10, 260-268 (1961)

[42] A. Frosini, M. Nivat, *On a tomographic equivalence between* $(0,1)$*-matrices*, Pure Mathematics and Applications 16, 251-265 (2005)

[43] A. Frosini, G. Palma, S. Rinaldi, *Combinatorial properties of degree sequences of 3-uniform hypergraphs arising from Saind arrays*, Beyond the horizon of computability, Lecture Notes in Comput. Sci. 12098, 228-238 (2020)

[44] A. Frosini, C. Picouleau, S. Rinaldi, *New sufficient conditions on the degree sequences of uniform hypergraphs*, Theoretical Computer Science 868, 97-111 (2021)

[45] M.R. Garey, D.S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman, New York (1979)

[46] J. Guédon, *The Mojette transform: theory and applications*, Wiley-ISTE (2009)

[47] L. Hajdu, *Unique reconstruction of bounded sets in discrete tomography*, Proceedings of the Workshop on Discrete Tomography and its Applications, Electron. Notes Discrete Math. 20, 15–25 (2005)

[48] L. Hajdu, R. Tijdeman, *Algebraic aspects of discrete tomography*, J. Reine Angew. Math. 534, 119–128 (2001)

[49] S.L. Hakimi, *On realizability of a set of integers as degrees of the vertices of a linear graph I*, J. Soc. Indust. Appl. Math. 10, 496-506 (1962)

[50] V. Havel, *A remark on the existence of finite graphs* (in Czech), Časopis Pěst. Mat. 80, 477-480 (1955)

[51] R. van der Hofstad, *Random Graphs and Complex Networks Vol 1*, Cambridge University Press (2017)

[52] S. Janson, *The probability that a random multigraph is simple*, Combin. Probab. Comput. 18(1-2), 205-225 (2009)

[53] M. Katz, *Question of uniqueness and resolution in reconstruction from projections*, Lect. Notes in Biomath., Springer Verlag (1977)

[54] J. Kleinberg, E. Tardos, *Algorithm Design*, Pearson (2013)

[55] M. Lothaire, *Combinatorics on Words*, Cambridge University Press, Cambridge (1997)

[56] M. Nivat, *Sous-ensembles homogènes de* $\mathbb{Z}^2$ *et pavages du plan* (in French), C.R. Acad. Sci. Paris Ser. I 335, 83-86 (2002)

[57] J. Radon, *Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten* (in German), Computed Tomography (Cincinnati, Ohio, 1982), Proc. Sympos. Appl. Math. 27, 71-86, Amer. Math. Soc., Providence, R.I. (1982)

[58] H.J. Ryser, *Combinatorial properties of matrices of zeros and ones*, Canadian Journal of Mathematics 9, 269-275 (1957)

[59] G. Sierksma, H. Hoogeveen, *Seven criteria for integer sequences being graphic*, J. Graph Theory 15(2), 223-231 (1991)

[60] A. Stolk, *Discrete tomography for integer-valued functions*, available at https://scholarlypublications.universiteitleiden.nl/access/item%3A2895501/view (2011)

[61] D.B. West, *Combinatorial Mathematics*, Cambridge University Press (2021)

[62] D.B. West, *Introduction to Graph Theory, Second Edition*, Prentice Hall (2001)

[63] *The On-Line Encyclopedia of Integer Sequences*, published electronically at http://oeis.org