Map Folding

by

Rahnuma Islam Nishat
B.Sc.Engg., Bangladesh University of Engineering and Technology, 2009

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Map Folding

by

Rahnuma Islam Nishat
B.Sc.Engg., Bangladesh University of Engineering and Technology, 2009

Supervisory Committee

Dr. Sue Whitesides, Supervisor
(Department of Computer Science)

Dr. Frank Ruskey, Departmental Member
(Department of Computer Science)

Supervisory Committee

Dr. Sue Whitesides, Supervisor
(Department of Computer Science)

Dr. Frank Ruskey, Departmental Member
(Department of Computer Science)

ABSTRACT

A crease pattern is an embedded planar graph on a piece of paper. An $m \times n$ map is a rectangular piece of paper with a crease pattern that partitions the paper into an $m \times n$ regular grid of unit squares. If a map has a configuration such that all the faces of the map are stacked on a unit square and the paper does not self-intersect, then it is flat foldable, and the linear ordering of the faces is called a valid linear ordering. Otherwise, the map is unfoldable. In this thesis, we show that, given a linear ordering of the faces of an $m \times n$ map, we can decide in linear time whether it is a valid linear ordering or not. We also define a class of unfoldable $2 \times n$ crease patterns for every $n \geq 5$.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

First I would like to thank my supervisor, Dr. Sue Whitesides for giving me an opportunity to work with her. She guided me throughout the Master's program, provided financial support and above all else been a good friend. I have learned a lot from her including the most effective way to teach a student. I have never hesitated to ask her anything and she has always given me a satisfactory answer patiently. Whenever I have expressed my opinion on something, she would consider it with importance. She has encouraged me to attend conferences and to get to know other researchers. In this way I have come to know some people who have created a positive influence on me and increased my determination to be a researcher.

I thank Dr. Frank Ruskey for the discussions we had on map folding. The two courses I took with him were very helpful and enjoyable. He was able to get me interested in combinatorics. I thank my undergraduate supervisor Dr. Saidur Rahman for inspiring me to be a researcher. He taught me how to dream big and how to make those dreams come true. I am grateful to have such a mentor at the beginning of my career.

I thank my family for being with me through my high and low points. I thank my friends for their support. I especially thank Jawaherul Alam for helping me at the beginning of my research career and Sudip Biswas for all his discussions and moral support. Thanks to my ex-roommate and friend Olga Sarycheva for making me feel like I was home. Thanks to my colleague and friend River for being supportive and helpful during the time I was writing my thesis.

Finally, I thank my best friend Jyoti, without whom I would not be here. I am a better person now than before because of him.

Dedicated to my father, Muhammad Muzahidul Islam; my mother, Nergis Shahina Akhter; my sister, Murshida Islam Rishta, and my brother, Waliuzzaman Jihad for always believing in me.

Chapter 1

Introduction

Origami or *paper folding* is a well-known art that is concerned with folding a piece of paper into various shapes, mostly shapes found in nature, such as birds or flowers, without cutting or tearing the paper. Although it is mainly a form of art, mathematicians have studied *paper folding* for over a century. In this thesis, we study folding a rectangular piece of paper called a *map*.

In this chapter, we first introduce *map folding*. We then give a formal statement of the map folding problem that we examine in this thesis and describe our main contributions. Finally, we give an overview of the organization of the chapters.

1.1 Map Folding

A *piece of paper* is a connected polygon in $\mathbb{R}^2$, with or without holes. Throughout this thesis, we assume that the thickness of a piece of paper is negligible. A common example of folding a piece of paper happens before putting a letter in an envelope. When writing a letter, we usually write on one side of the paper, which we call the *light side*. The other side of the paper is the *dark side*. After writing, we fold the letter and put it in an envelope. When the receiver unfolds the letter, he or she sees a pattern on the surface of the paper created by the folds. This pattern is called a *crease pattern*. More specifically, a *crease pattern* is an embedded planar graph on a piece of paper (for more details, please see Section 3.1). Each edge of a crease pattern that is not on the boundary of the paper is called a *crease*. By definition, two creases in a crease pattern do not intersect except possibly at their shared endpoint. The crease pattern divides the surface of the paper into a set of bounded regions called *faces*. Each face is bounded by a set of creases and possibly by part of the boundary of the paper. Each crease is incident to exactly two faces.

If a crease pattern partitions a rectangular piece of paper without holes into an $m \times n$ regular grid of unit squares, then the piece of paper is called an $m \times n$ *grid paper* or an $m \times n$ *map*. A crease pattern on an $m \times n$ map is called an $m \times n$ *crease pattern*. To differentiate the crease pattern on a map from a general crease pattern, we point out the difference between unfolding a letter and unfolding an envelope. Usually both of the unfolded papers are rectangular. The creases on the letter are parallel to the sides of the letter and all the faces are roughly of the same size and shape. On the other hand, the creases on the unfolded envelope are not necessarily parallel to the sides and the faces may not be of similar size. Both the patterns on the letter and the unfolded envelope are examples of crease patterns, but the letter becomes a *map* when each face of the letter is a unit square.

Folding a letter is easy when we do not have any pre-specified crease pattern on the piece of paper. Assume that a crease pattern is drawn on the paper prior to folding and we have to fold the letter respecting the given pattern. Then the paper might not be foldable. *Map Folding* is a branch of paper folding that is concerned with folding an $m \times n$ map with a given crease pattern, when the way the paper is folded about each crease may or may not be specified.

When we fold a piece of paper with a given crease pattern, we are restricted to fold the paper only along the creases. A crease can be folded either as a *mountain* or as a *valley*. A *mountain fold* folds the paper such that the two faces incident to the crease touch each other on the dark side after the fold. Similarly, a *valley fold* folds the paper such that the two faces incident to the crease touch each other on the light side after the fold.

A *mountain-valley assignment* is a many-to-one function from the creases in a crease pattern to a label set $\{M, V\}$. A *mountain-valley pattern* is a crease pattern together with a mountain-valley assignment. In other words, a mountain-valley pattern is a crease pattern, where each crease has been assigned either a mountain (M) or a valley (V) label. Figure 1.1 shows a mountain-valley pattern on a 3×3 map. The valley creases are denoted by triple-dot dashed (red) lines and the mountain creases are denoted by dashed (blue) lines. We call a crease pattern without a mountain-valley assignment an *unassigned crease pattern*.

In this thesis, we study folding maps with given mountain-valley patterns. From now on, unless otherwise specified, we assume that each crease pattern has a mountain-

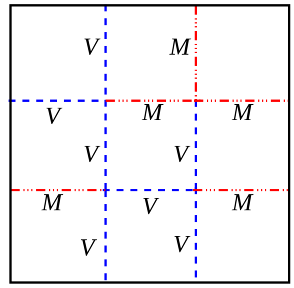


Figure 1.1: A mountain-valley pattern on a 3×3 map.

valley assignment.

1.2 Problem Statement

We examine the following open problem posed by Jack Edmonds in 1997 [6].

Open Problem 1: *What is the complexity of deciding whether an $m \times n$ map with a given mountain-valley pattern is* flat foldable *(i.e., has a* final flat-folded state*)?*

A *flat foldable* map and a *final flat-folded state* are defined as follows. Let P be a map with crease pattern $\mathbb{C}$. A *fragment* of P is a subset of faces of $\mathbb{C}$ that form a connected rectangular region (without a hole). A *flat folded state* of P is a stack of disjoint fragments of P that are parallel to each other, connected along the creases of $\mathbb{C}$, and such that the union of all the fragments is P. Each fragment in the stack is called a *layer*. For example, a folded letter inside an envelope is a flat folded state of the letter, where each layer consists of a single face of the letter and the layers are connected along the creases. Figure 1.2 shows an example of a flat folded state of a 6×8 map. A *final flat-folded state* of P is a flat folded state where each layer consists

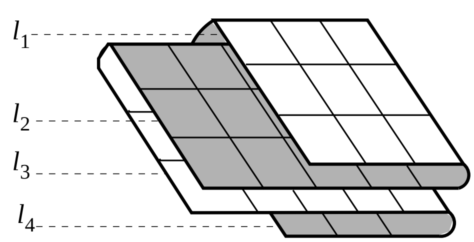


Figure 1.2: A flat folded state of P with four layers l_1, l_2, l_3 and l_4.

of exactly one face of $\mathbb{C}$. If P has a final flat-folded state, then P is *flat foldable*. A final flat-folded state of P is also called a final flat-folded state of $\mathbb{C}$. The terminology used is consistent with the literature [2, 1, 6].

Let P be a flat-foldable map. Then by definition, in a final flat-folded state of P, all the faces of P must be stacked on a unit square, say in the XY-plane. Let L be the ordering of the faces of P from bottom to top on a unit square in a final flat-folded state of P. Then L is called a *valid linear ordering* of the faces of P. Using these definitions, we can state the open problem above in a different way: *given an $m \times n$ map with a specified mountain-valley pattern, what is the complexity of deciding whether a valid linear ordering of the faces of the map exists?*

In this thesis, we investigate this problem and ask the following questions.

Problem 2: Given an arbitrary linear ordering L of the faces of a map, is it possible to decide whether L is a valid linear ordering? If it is possible, then what is the time complexity of this decision problem? Can we enumerate all possible valid linear orderings of a given crease pattern on a map?

Problem 3: What are the crease patterns on a $2 \times n$ map that do not have final flat-folded states? Let us call them the *unfoldable patterns*. Is there a way to characterize the unfoldable patterns?

1.3 Our Contributions

In this thesis, we investigate the combinatorial properties of an $m \times n$ crease pattern (assigned or unassigned).

We show that the adjacency relationships between the faces of a grid paper (the concept of *butterflies*) can be used to determine whether a given linear ordering of an $m \times n$ crease pattern (unassigned or assigned) is a valid linear ordering. We then show that the adjacency relationships determine whether the light side or the dark side of a face of the map faces up (the concept of *checkerboard pattern*) in a final flat folded state (if it exists). We also show that for a specific mountain-valley assignment of the input crease pattern, we get a *directed network* of the faces of the map, which any valid linear ordering of the mountain-valley pattern must satisfy.

In summary, we get the following results in this thesis.

1. We show that given a linear ordering of the faces of an $m \times n$ crease pattern (assigned or unassigned), we can decide in linear time whether it is a valid linear ordering or not.

2. We prove that there is an unfoldable $2 \times n$ mountain-valley pattern for each and every $n \geq 5$. We also define a class of unfoldable $2 \times n$ mountain-valley patterns for every $n > 5$.

1.4 Thesis Outline

Here is an outline of the rest of the thesis.

In Chapter 2, we present a brief history of origami. We then discuss related mathematical works on paper folding, emphasizing the works related to map folding.

In Chapter 3, we show that we can decide in linear time whether a given linear ordering of the faces of a map is a valid linear ordering. We also sketch an exponential time algorithm to enumerate all possible valid linear orderings of a given crease pattern. (solution of **Problem** 2).

In Chapter 4, we characterize a class of unfoldable $2 \times n$ patterns, for every $n > 5$ (solution of **Problem** 3).

In Chapter 5, we summarize our results, state some open problems and discuss directions for future research.

Chapter 2

Background and Related Work

In this chapter, we give a brief history of mathematical origami and mention some applications of origami. We then describe some previous results that are closely related to this thesis.

2.1 History of Mathematical Origami

Origami became popular among mathematicians at the end of the 19th century. Tandalam Sundara Rao [34] published a book in 1893, where he showed how to use paper folding to trisect an arbitrary angle. A few years later, in 1897, Felix Klein [23] wrote a book collecting some famous problems on geometry, which sparked an interest in solving mathematical problems using paper folding. In 1930, Giovanni Vacca [9] published an article demonstrating how paper folding can solve quadratic equations.

In 1867, Eduard Lill [8] showed how to solve a polynomial equation by drawing an orthogonal path on the plane. Let $f(x) = a_n x^n + \ldots + a_1 x + a_0$ be a polynomial with real coefficients. Then according to Lill's method, a real root of $f(x)$ can be found by creating a path on the plane based on the coefficients of $f(x)$. In 1936, Margharita P. Beloch [31] showed how to construct the path in Lill's method for a cubic equation using paper folding. She introduced a fold called the 'Beloch Fold' and gave a path construction using the Beloch fold to solve cubic equations, which refuted Rao's [34] claim that paper folding cannot be used to calculate the cube root of two.

The set of *Huzita-Hatori axioms* [18] is one of the most important contributions in mathematical paper folding. In this context, an *axiom* is an elementary folding move that makes a particular crease on the paper. The *Huzita-Hatori axioms* consist of all the elementary folding moves known by 1991. The axioms were reported by Humiaki Huzita at the First International Conference on Origami in Education and Therapy in 1991, although originally these axioms were formulated by Jacques Justin in 1989 [19]. Robert Lang [27] developed the first algorithm to construct crease patterns to achieve some specific shapes, which was a major contribution in combining art and mathematics in the field of origami.

An interesting branch of mathematical origami is *flat folding*, which seeks to characterize flat foldable papers. Thomas Hull published a series of papers on flat folding [13–16] and a book on mathematical origami in 2006 [17]. He not only published original research, but also brought back long forgotten works on origami. We discuss flat folding in detail later in this chapter.

A *vertex* of a crease pattern is an endpoint of a crease that is not on the boundary of the paper. Hull [16] gave upper and lower bounds on the number of flat foldable mountain-valley assignments on a single-vertex crease pattern on a disk. Researchers have also been interested in combinatorial problems in origami. Justin [19] enumerated a number of unfoldable mountain-valley patterns on 2×5, 2×6 and 2×7 maps.

The *permutations* of a $1 \times n$ map are the various $n!$ linear orderings of the faces of the map. Observe that these orderings include both valid linear orderings and orderings that are not valid. In his famous book [11], Gardner mentions that he was able to fold 60 *permutations* among the 720 for a 1×6 grid paper, without the support of any algorithm. Gardner considered a 1×6 crease pattern without a specific mountain-valley assignment. Therefore, each crease can be folded either as a mountain or as a valley. A $1 \times n$ grid paper without a mountain-valley assignment is often referred as a *strip of stamps of size* n in the literature. Many researchers concentrated on counting the number of valid linear orderings of a strip of stamps of size n [7, 26, 35, 29]. Lunnon considered folding $p \times q$ (2-dimensional) papers and multidimensional $(2 \times 2 \times \ldots \times 2)$ papers [30], but he assumed that distortions might be necessary for some folds.

2.2 Applications of Origami

Mathematicians have been interested in origami not only because of theoretical interests, but also because origami has important applications in various fields. In the following we mention a few of the numerous applications of origami.

- Origami has been used to create many puzzles and games. For example, we can write one letter on each stamp in a strip of stamps . Then folding the strip to a single square gives us an ordering of the stamps, in other words the letters, from top to bottom or from bottom to top. It is a really interesting game for children who are starting school to find all the anagrams with those letters that have meanings. There are many more puzzles that require folding maps [11]. In 1942, a puzzle based on flat folding was made commercially available. The goal of that puzzle was to put certain war criminals behind bars. That is, one has to fold the paper into a square in such a way that two of the faces appear behind the cut out jail windows, one at the bottom and the other on the top of the final folded square. For more exciting games and puzzles regarding folding, we refer the reader to [33].

- Folding techniques can be used to generate fractal curves that have applications in graphics, animation, and tiling [5]. A good example is the Heighway Dragon Curve [10, 3, 4], which is constructed in the following way. Take a rectangular piece of paper and fold it in half by aligning the left edge of the paper onto the right edge. Apply the same folding operation on it recursively. Then unfold the paper keeping a 90° angle at each crease. Let **R** and **L** refer to a clockwise turn by 90° and an anticlockwise turn by 90°, respectively. If we unfold after one iteration, we get the sequence: **R**. After two iterations, the sequence is: **RRL**. After three iterations, the sequence is: **RRLRRLL**. Therefore, after each iteration, each **R** in the previous iteration is replaced with the sequence **RRL**.

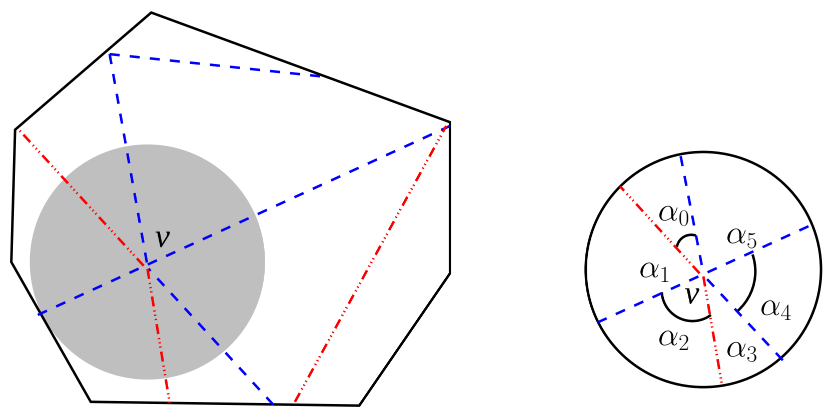
- Paper folding techniques have been used in packaging products on assembly lines [28] and in the sheet metal industry [37]. They have also been used in solving mathematical problems for centuries [34, 23, 31, 8]. A map folding technique, known as Miura Folding, has been used to deploy collapsible solar panels in space satellites [25].

2.3 Mathematics of Flat Folding

Flat Folding is a branch of paper folding that deals with flat foldability. In this section, we review several results on flat foldability that are closely related to our work. We first describe different folding models and then describe some complexity results in flat folding.

2.3.1 Single-Vertex Crease Patterns

Let P be a piece of paper, where P is not necessarily of rectangular shape. Let $\mathbb{C}$ be a crease pattern on P such that P is flat foldable with respect to $\mathbb{C}$. Let v be a vertex of $\mathbb{C}$. Suppose we draw any circle r around v such that no other vertex of $\mathbb{C}$ is on r or inside r (see Figure 2.1). Since P is flat foldable with respect to $\mathbb{C}$, the disk bounded by r is also flat foldable.



(a) (b)

Figure 2.1: (a) A flat foldable mountain-valley pattern on a paper P, and (b) the disk r cut out from P where v is the center of r.

The following results are known for a crease pattern on a disk with a single vertex v at the center of the disk.

Lemma 2.1.[21] *The difference between the number of creases with the label mountain and the number of creases with the label valley meeting at v is 2.*

It easily follows from Lemma 2.1 that the number of creases incident to v is even. In Figure 2.1(b), there are six creases around v, four of them are valleys (V) and two are mountains (M). Therefore, $||M| - |V|| = 2$, where $|M|$ and $|V|$ denote the number of creases labeled mountain and valley, respectively.

Lemma 2.2.[22, 19, 13] *The sum of either set of alternate angles around v is π.*

In Figure 2.1(b), $\alpha_0 + \alpha_2 + \alpha_4 = \pi$ and $\alpha_1 + \alpha_3 + \alpha_5 = \pi$. If the conditions stated in Lemma 2.1 and Lemma 2.2 are satisfied for all the vertices in $\mathbb{C}$, we say that the crease pattern $\mathbb{C}$ is *locally flat foldable.*

In this thesis, all crease patterns are locally flat foldable unless stated otherwise.

2.3.2 Simple Folding

In this section, we briefly describe simple folding techniques introduced by *Esther Arkin, Michael Bender, Erik Demaine, Martin Demaine, Joseph Mitchell, Saurabh Sethia, and Steven Skiena* [1]. We first review some terminology used in [1]. We next describe the general definition of simple folding that is applicable both in 1D and 2D. We then describe their algorithms to determine whether a 1D paper is flat foldable using simple folding and a 2D paper is flat foldable using simple folding, respectively.

A *1D piece of paper* is a line segment along the X-axis and a *2D piece of paper* is a connected simple polygon on the XY-plane. *Creases* have one less dimension than the paper. Thus, a crease on a 1D paper is a point and a crease on a 2D paper is a line segment. The *top side of a flat folded 1D paper* consists of the parts of the paper that are visible from the positive Y-axis and the *bottom side of a flat folded 1D paper* consists of the parts of the paper that are visible from the negative Y-axis. Similarly, the *top side of a flat folded 2D paper* consists of the parts of the paper that are visible from the positive Z-axis and the *bottom side of a flat folded 2D paper* consists of the parts of the paper that are visible from the negative Z-axis. Then the top side and the bottom side of the unfolded paper are the light side and the dark side, respectively. Note that in a flat folded state, the top side and the bottom side may consist of both light pieces and dark pieces.

Let P be a paper (1D or 2D) with crease pattern $\mathbb{C}$. A *flat folding operation* on P maps one flat folded state of P to another flat folded state of P. A *simple fold* on P is a flat folding operation on P that is performed along a crease on the top side of a flat folded state of P. Let l_1 be a layer of a flat folded state of P and let c be a crease on l_1 that is visible on the top side. Arkin et al. [1] distinguished three types of simple folds about the crease c as follows.

1. **One-layer simple fold:** A *one-layer simple fold* along c corresponds to continuously rotating a portion of layer l_1 that is visible on the top side and incident to the crease c, by 180° about c. The rotation can be either clockwise or counterclockwise depending on the label of c (mountain or valley). During this

rotation, P remains rigid everywhere except at c and P does not self-intersect. Figure 2.2(a) and (b) show a one-layer simple fold on a flat folded state of a 1D paper.

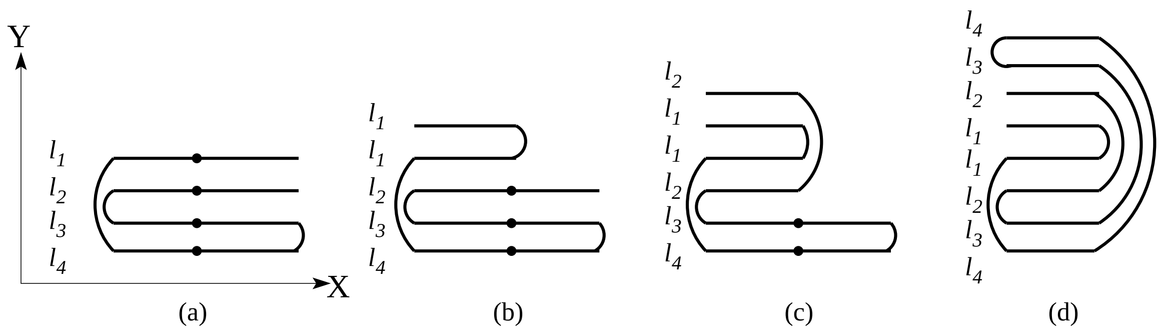


Figure 2.2: (a) A flat folded state of a 1D paper with four layers l_1, l_2, l_3, l_4. The creases are shown in black dots. We assume that the creases are labeled valley when seen from above (i.e., positive Y-axis), (b) a one-layer simple fold of the topmost layer l_1, (c) a some-layers simple fold of the top two layers l_1, l_2, and (d) an all-layers simple fold.

2. **Some-layers simple fold:** A *some-layers simple fold* of k layers of P along c is the operation of continuously rotating a portion of layer l that is visible on the top side and incident to the crease c along with $k-1$ layers of the paper immediately below c. The rotation is by 180° about c in clockwise or counterclockwise direction depending on the label of c. P does not self-intersect, and remains rigid everywhere except at c on l_1 and at the creases on the $k-1$ layers immediately below c. See Figure 2.2(a) and (c). Note that a one-layer simple fold is the special case of a some-layers simple fold when $k = 1$, i.e., only the topmost layer is folded along a crease.

3. **All-layers simple fold:** An *all-layers simple fold* is the special case of the some-layers simple fold when k is equal to the number of layers coinciding at the specified crease c. See Figure 2.2(a) and (d).

Arkin et al. [1] defined *simple folding* as a sequence of simple folds. A *simply folded state* of a paper P (1D or 2D) is a flat folded state of P reached using simple folding. P is *simply flat foldable* if there exists a final flat folded state of P that is a simply folded state.

An *orthogonal crease pattern* is a crease pattern where the creases are parallel to the sides of the paper. Notice that the creases that are parallel to the same side of the paper are not necessarily equidistantly spaced. Since a 1D paper is a line segment and the creases are points on the segment, a 1D crease pattern is orthogonal by definition. In the next two sections, we describe the algorithms given by Arkin et al. [1] to determine whether an orthogonal mountain-valley pattern $\mathbb{C}$ (1D or 2D) is simply flat foldable or not.

1D Folding

In this section, we show an algorithm given by Arkin et al. [1] to determine whether a 1D mountain-valley pattern is flat foldable using simple folding.

Let P be a 1D piece of paper and let $\mathbb{C} = c_1, c_2, \ldots, c_n$ be the mountain-valley pattern on P, where c_1 denotes the leftmost crease. We denote by $P(t)$ the flat folded state of P after t flat folding operations, and by $\mathbb{C}(t) = c_1^t, c_2^t, \ldots, c_{n_t}^t$ the mountain-valley pattern on $P(t)$, where n_t is the number of creases in $\mathbb{C}(t)$. Then $P(0) = P$ and $\mathbb{C}(0) = \mathbb{C}$. For any t, c_0^t and $c_{n_t+1}^t$ denote the left end and right end of the paper $P(t)$, respectively. Figures 2.4(a) and (b) show a 1D paper P with a crease pattern and a flat folded state $P(2)$ of P after two simple folds, respectively.

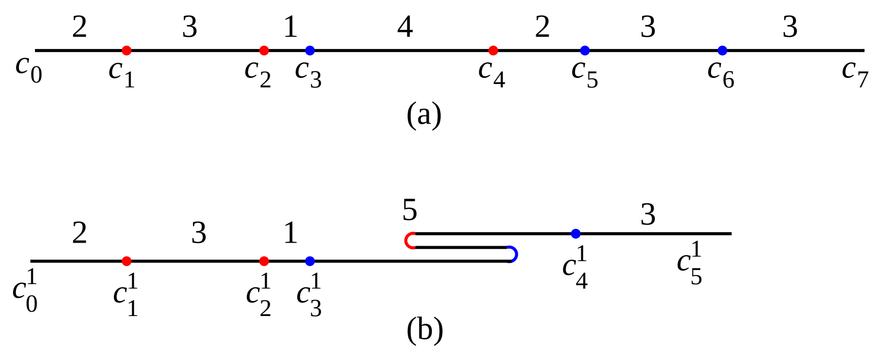


Figure 2.3: (a) A 1D piece of paper P with a crease pattern $\mathbb{C} = c_1, c_2, \ldots, c_6$. The creases with the labels mountain and valley are shown as blue and red dots, respectively. (b) A flat folded state $P(2)$ of P with the crease pattern $\mathbb{C}(2) = c_1^1, c_2^1, \ldots, c_5^1$ that is obtained by a pair of simple folds (called a "crimp") along the pair of creases c_4, c_5 in $\mathbb{C}$.

For any two creases c_i^t and c_j^t in $\mathbb{C}(t)$, c_i^t occurs to the left of c_j^t when $i < j$. The *distance between two creases* c_i^t and c_j^t is the distance along the X-axis between their corresponding points and is denoted by $|c_i^t - c_j^t|$. Figures 2.3(a)–(b) show the distance between any two consecutive creases visible on the top side and between the endpoints of the paper and the respective creases closest to them.

Arkin et al. [1] defined the following two operations on 1D crease patterns; each of the operations is a combination of one or two simple folds.

(a) **Crimp:** Let c_i^t, c_{i+1}^t be two consecutive creases in $\mathbb{C}$, where $1 \leq i < n$, such that the labels (mountain or valley) of c_i^t and c_{i+1}^t are opposite. The pair of creases c_i, c_{i+1} is *crimpable* if $|c_{i-1} - c_i| \geq |c_i - c_{i+1}|$ and $|c_{i+1} - c_{i+2}| \geq |c_i - c_{i+1}|$. A *crimp* operation on the pair c_i^t, c_{i+1}^t corresponds to the application of a one-layer simple fold on c_i^t and then a one-layer simple fold on c_{i+1}^t. Figure 2.3 shows an example of a crimp operation.

(b) **End Fold:** The end $c_{n_t+1}^t$ is a *foldable end* if $|c_{n_t-1}^t - c_{n_t}^t| \geq |c_{n_t}^t - c_{n_t+1}^t|$ and the end c_0^t is a *foldable end* if $|c_1^t - c_2^t| \geq |c_0^t - c_1^t|$. An *end fold* operation on c_0^t or $c_{n_t+1}^t$ corresponds to a one-layer simple fold on c_1^t or $c_{n_t}^t$, respectively. Figure 2.4(b) shows the end fold operation on the left end c_0 of the paper in Figure 2.4(a).

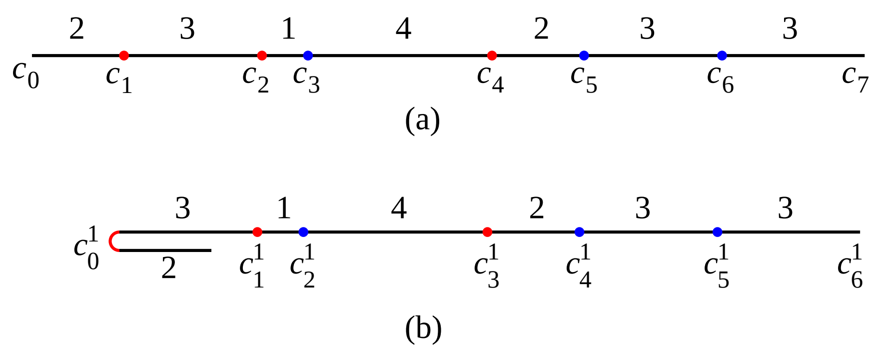


Figure 2.4: (a) A 1D piece of paper P with a crease pattern $\mathbb{C} = c_1, c_2, \ldots, c_6$. The creases with the labels mountain and valley are shown as blue and red dots, respectively. (b) A flat folded state $P(1)$ of P with the crease pattern $\mathbb{C}(1) = c_1^1, c_2^1, \ldots, c_5^1$ that is obtained from P by an end fold operation on the leftmost crease c_1 in $\mathbb{C}$.

Let $P(t)$ be a flat folded state of a 1D paper P. We can obtain a flat folded state $P(t+1)$ by applying an end fold operation on a foldable end of $P(t)$ (if it exists). Similarly, we can obtain a flat folded state $P(t+2)$ by applying a crimp on $P(t)$ if a crimpable pair exists in $\mathbb{C}(t)$.

Arkin et al. [1] gave the following algorithm to determine whether a given 1D mountain-valley pattern is foldable using simple folding.

They showed that any flat-foldable 1D mountain-valley pattern is foldable by Algorithm 1 using only one-layer simple folds or some-layer simple folds in $O(n)$ time. The following theorem summarizes the results of Arkin et al. [1] on 1D mountain-valley patterns.

Algorithm 1: 1Dsimplefolding(ℂ)

```
ℂ = c_1, ..., c_n is the input crease pattern
if n == 0 then
    return true
else
    if there is a crimpable pair or foldable end in ℂ then
        Perform crimp or end fold and obtain the reduced crease pattern ℂ′
        1Dsimplefolding(ℂ′)
    else
        return false
```

Theorem 2.1. *[1] Let P be a 1D mountain-valley pattern. Then the following are equivalent:*

1. P is flat foldable.

2. P is foldable by a sequence of some-layers simple folds.

3. P is foldable by a sequence of one-layer simple folds.

Moreover, "The 1D one-layer and some-layers simple-fold problems can be solved in $O(n)$ worst-case time on a machine supporting arithmetic on the input lengths."

Universality of Simple Folding

Let P be a $1 \times n$ map with a mountain-valley pattern $\mathbb{C}$, where the creases are equidistantly spaced. Uehara [36] showed that simple folding is strong enough to get the following results on P.

Lemma 2.3. *[36] Let P be a $1 \times n$ map with the mountain-valley pattern $\mathbb{C}$, where the creases in $\mathbb{C}$ are equidistantly spaced. Then any flat folded state of P can be reached by a sequence of end folds.*

By Lemma 2.3, any $1 \times n$ mountain-valley pattern is flat foldable using simple folding.

2D Folding

In this section, we give a brief overview of the algorithm given by Arkin *et al.* [1] to decide whether a 2D orthogonal crease pattern is simply flat foldable (i.e., flat foldable using simple folding). Since we focus on map folding in our thesis, we describe their algorithm for $m \times n$ maps.

A *crease line* is a maximal set of creases that are collinear and visible on the top side of a flat folded state of a map such that the line extends all the way across the top side. An unfolded $m \times n$ map has exactly $m-1$ horizontal crease lines and $n-1$ vertical crease lines. A crease line is *homogeneous* if all the creases on the line have the same label. Figure 2.5 shows a 4×4 map with two horizontal homogeneous crease lines.

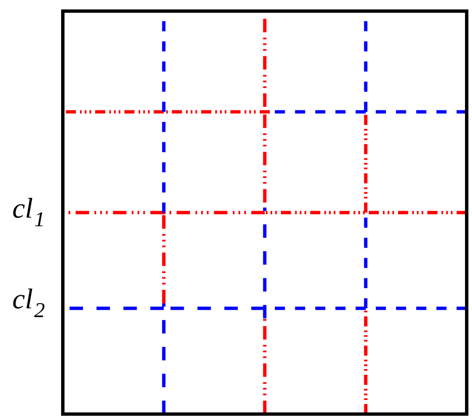


Figure 2.5: An $m \times n$ map with two homogeneous crease lines.

Let P be an $m \times n$ map with the crease pattern $\mathbb{C}$. Let S be a flat folded state of P. S is a *valid simply flat folded state* if all the pairs of creases that lie on one another have the same label (mountain or valley) when looking down from the positive Z axis. Note that a crease with a mountain label on the light side of the paper has the label valley on the dark side of the paper and vice versa. If S is a valid simply flat folded state, we say that all the creases in $\mathbb{C}$ *match up* in S.

Let $\mathcal{H}$ be the set of all the homogeneous crease lines in S. Since $\mathbb{C}$ is locally flat foldable, $\mathcal{H}$ cannot have both horizontal and vertical crease lines. This is because in that case, the vertex v, where a vertical homogeneous crease line and a horizontal homogeneous crease line intersect, cannot have three mountain creases and one valley crease or three valley creases and one mountain crease. Otherwise, Lemma 2.1 would not hold at v.

We now describe the algorithm given by Arkin et al. [1]. The input of the algorithm is an $m \times n$ map P with a crease pattern $\mathbb{C}$. By definition, the initial unfolded state of P is a valid simply flat folded state. The first step of the algorithm is to find the set $\mathcal{H}$ of homogeneous crease lines. Without loss of generality we assume that $\mathcal{H}$ contains horizontal crease lines. If $\mathcal{H}$ is empty, then no simple fold can be made and P is not flat foldable using simple folding. Otherwise, there is at least one crease line in $\mathcal{H}$. Arkin et al. [1] proved that the creases in $\mathcal{H}$ must be folded before any other crease using the following lemma.

Lemma 2.4. *[1] Let P be an $m \times n$ map with the mountain-valley pattern $\mathbb{C}$. Let $\mathcal{H}$ be the set of all the horizontal homogeneous crease lines in any flat folded state S of P. Then all the creases in $\mathcal{H}$ must be folded before any other crease in $\mathbb{C}$ in a simple folding.*

Proof. Since simple folds can be applied only to the creases on the top side of a flat folded state, it suffices to show that the creases in $\mathcal{H}$ must be folded before any crease on the top side of S. Suppose for a contradiction that some other crease can be folded before the crease lines in $\mathcal{H}$. By definition, a simple fold maps a flat folded state of P to another flat folded state of P. Therefore, we have to fold the paper along a crease line. But there are no other vertical or horizontal crease lines in S that are homogeneous and hence, we cannot fold the paper along any other crease line, a contradiction. Therefore, no other crease can be folded before the creases in $\mathcal{H}$. □

The second step of the algorithm is to apply simple folds along the crease lines in $\mathcal{H}$. To determine whether the crease lines in $\mathcal{H}$ can be folded using simple folding, they considered the crease pattern $\mathbb{C}'$ obtained from $\mathbb{C}$ by removing all the creases that are not in $\mathcal{H}$ and merging all the creases along each crease line in $\mathcal{H}$ into one crease. Since $\mathcal{H}$ contains only horizontal crease lines, $\mathbb{C}'$ is a 1D crease pattern. Therefore, Algorithm 1 can be applied on $\mathbb{C}'$ to obtain a flat folded state S' of P if it exists and we proceed to the next step. If such a flat folded state does not exists, then P is not simply flat foldable.

The third step of the algorithm is to check whether S' is a valid simply flat folded state. If all the creases in $\mathbb{C}$ match up in S', then the three steps of the algorithm are applied on S' recursively. Otherwise, P is not simply flat foldable.

The formal description of the algorithm is given below.

Algorithm 2: 2Dsimplefolding(ℂ)

```
ℂ is the input crease pattern
if there are no unfolded creases in ℂ then
    return true
Find the set of homogeneous crease lines H
if H == φ then
    return false
else
    Let the 1D crease pattern obtained by merging the creases on each crease
    line in H into one crease and removing all other creases of ℂ be ℂ′.
    Apply Algorithm 1 on ℂ′
    if ℂ′ is flat foldable then
        Let the flat folded state obtained after folding the creases in ℂ′ be S′.
        if S′ is a valid simply flat folded state then
            Let the crease pattern on the top side of S′ be ℂ″
            2Dsimplefolding(ℂ″)
        else
            return False
    else
        return false
```

The following theorem states the correctness of the algorithm.

Theorem 2.2. *[1] Some-layers simple folding of an orthogonal crease pattern on a rectangular piece of paper can be done in linear time using Algorithm 2.*

Complexity Results

In this section, we briefly mention the computational complexity of various foldability problems.

Bern and Hayes [2] proved that both the *flat foldability* and the *assigned flat foldability* problems are NP-complete. The *flat foldability* problem asks whether a paper with a given crease pattern has a final flat folded state, where the creases are not necessarily labeled and the crease pattern satisfies Lemma 2.2. The *assigned flat foldability* problem asks whether a paper with a given locally flat foldable crease pattern has a final flat folded state, where each crease is labeled either mountain or valley. The hardness proofs of these two problems are based on a reduction from the

Not-All-Equal-3-SAT problem. For details on the theory of computational complexity including the Not-All-Equal-3-SAT problem, we refer the reader to Garey and Johnson [12].

Arkin et al. [1] proved NP-completeness for some variations of the flat foldability problem. They showed that it is weakly NP-hard (i.e., there exists a pseudo-polynomial time algorithm) to decide whether a 2D axis-parallel mountain-valley pattern is simply flat foldable, where the initial paper is allowed to be an arbitrary orthogonal polygon. They also showed that it is weakly NP-hard to decide whether a mountain-valley pattern on a square piece of paper can be folded by some-layers simple folds, where the creases are allowed to be axis-parallel and at a 45° angle. The decision problems are weakly NP-hard because both of the hardness proofs reduce the *Partition* problem, which is weakly NP-hard according to Garey and Johnson [12]. They also proved that the problem of deciding whether an orthogonal piece of paper (a piece of paper with axis-parallel edges that is not necessarily rectangular) with an orthogonal crease pattern (without a mountain-valley assignment) is simply flat foldable is weakly NP-complete, for both all-layers and some-layers simple folds.

2.4 Conclusion

In this chapter, we described a brief history of mathematical origami. We then sketched some results from the literature on flat folding that are closely related to our work. We finally discussed the complexity results on this topic. In the next chapter, we focus on *valid linear orderings*. We show that we can decide whether a given linear ordering of the faces of an $m \times n$ crease pattern is a *valid linear ordering* in time linear to the size of the input linear ordering.

Chapter 3

Valid Linear Orderings

In this chapter, we show that given any linear ordering L of the faces of an $m \times n$ crease pattern $\mathbb{C}$ (assigned or unassigned), we can decide in linear time whether L is a valid linear ordering. In Section 3.1, we define the necessary terminology that we use in this chapter. Then in Section 3.2, we give linear time algorithms for deciding whether a given linear ordering of an unassigned crease pattern is a valid linear ordering. In Section 3.3, we solve the same decision problem as in Section 3.2 for mountain-valley patterns. Finally, in Section 3.4, we briefly sketch an exponential-time algorithm to enumerate all the valid linear orderings of a crease pattern on a map.

3.1 Preliminaries

Let P be an $m \times n$ map with crease pattern $\mathbb{C}$. A *vertex of* $\mathbb{C}$ is an endpoint of a crease of $\mathbb{C}$ that is not on the boundary of the paper. A *vertex of* P is either an endpoint of a crease or a corner of the boundary of P. We call a vertex of P that is not a vertex of $\mathbb{C}$ a *dummy vertex*. Figure 3.1(a) shows the vertices of a 3×3 map, where the dummy vertices are shown as red disks.

An *edge* of P is either a crease or a straight line segment between two consecutive vertices on the boundary of P. A *face* of $\mathbb{C}$ is a unit square bounded by four edges of P. For example, the crease pattern in Figure 3.1(a) has nine faces. We denote by $f_{i,j}$ a face of $\mathbb{C}$ that has the vertices $v_{i,j}, v_{i+1,j}, v_{i,j+1}, v_{i+1,j+1}$ on its boundary. Figure 3.1(b) shows all the faces of the crease pattern in Figure 3.1(a). For each face $f_{i,j}$ of $\mathbb{C}$, we associate the creases $(v_{i,j}, v_{i+1,j})$ (left side of the unit square), where $0 < j < n$, and $(v_{i,j}, v_{i,j+1})$ (top of the unit square), where $0 < i < m$, to $f_{i,j}$. The creases associated

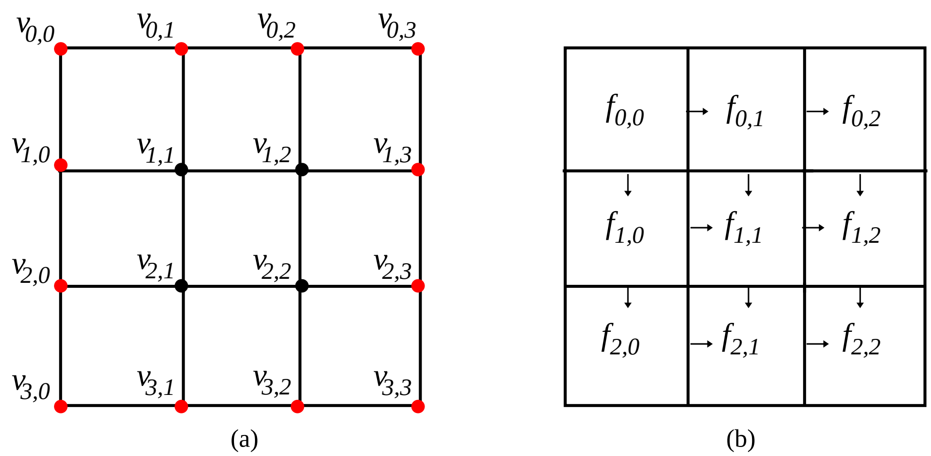


Figure 3.1: (a) A 3×3 map with crease pattern $\mathbb{C}$, where the vertices of $\mathbb{C}$ are shown as black disks and the dummy vertices of P are shown as red disks. (b) The faces of $\mathbb{C}$.

with each face are shown in Figure 3.1(b).

A *column* c_j of $\mathbb{C}$ is a set of m faces $f_{0,j}, f_{1,j}, \ldots, f_{m-1,j}$, where $0 \leq j \leq n-1$. A *row* r_i of $\mathbb{C}$ is a set of n faces $f_{i,0}, f_{i,1}, \ldots, f_{i,n-1}$, where $0 \leq i \leq m-1$. The creases associated with a column c_j are the creases associated with the faces in c_j. Similarly, the creases associated with a row r_i are the creases associated with the faces in r_i.

3.2 Unassigned Crease Patterns

Let P be an $m \times n$ map and let $\mathbb{C}$ be the unassigned crease pattern. In this section, we show that we can decide whether a given linear ordering of the faces of $\mathbb{C}$ is a valid linear ordering. We first introduce the concepts of "checkerboard pattern" and "butterflies". We then use those concepts to recognize a valid linear ordering in $O(mn)$ time.

3.2.1 Checkerboard Pattern

Let us assume that P is flat foldable and let L be the linear ordering of the faces of $\mathbb{C}$ in a final flat folded state S_f of P. Without loss of generality we assume that the face $f_{0,0}$ is facing light side up in S_f and the vertex $v_{0,0}$ is incident to the top-left corner of the unit square on which the faces are stacked. It is easy to observe that the faces that share an edge with $f_{0,0}$ must face dark side up. In a similar way, if a face $f_{i,j}$, $0 \leq i \leq m-1$ and $0 \leq j \leq n-1$, is facing light side (respectively, dark side) up, then all the faces that share an edge with $f_{i,j}$ must face dark side (respectively, light side) up. So the faces of $\mathbb{C}$ form a *checkerboard pattern*, where the color of a face f depends on which side of f must face up in any final flat folded state of P (under our assumption), as shown in Figure 3.2.

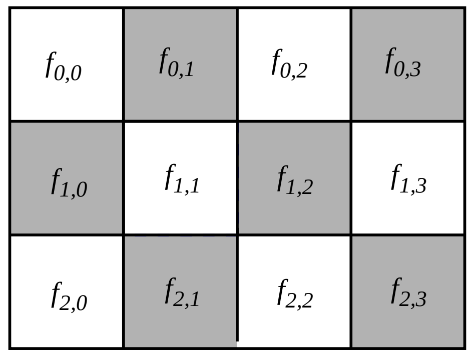


Figure 3.2: A 3×4 crease pattern $\mathbb{C}$. The faces pointing light side up in the final flat folded state are shown in white and the other faces are shown in gray.

From now on, we assume that the face $f_{0,0}$ faces light side up in any final flat folded state of P (if one exists), and the vertex $v_{0,0}$ is incident to the top-left corner of the unit square on which the faces are stacked.

3.2.2 Butterflies

A *butterfly* B is a pair of faces f, f' of $\mathbb{C}$ incident to the same crease e. We call f and f' the *wings* of B and the crease e the *hinge* of B. A *pair of butterflies* is a set of two butterflies B_1 and B_2 that have no wing in common.

Let S be any flat folded state of P and let B_1, B_2 be a pair of butterflies such that the wings of B_1, B_2 lie above the same unit square u on the XY-plane and the hinges of B_1, B_2 lie above the same edge of u (see Figure 3.3).

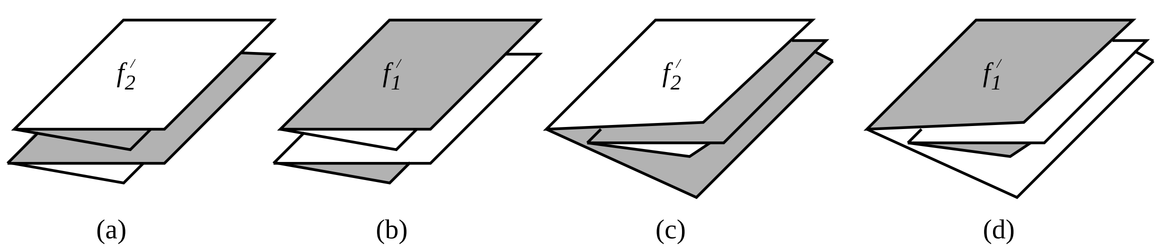


Figure 3.3: A pair of butterflies B_1, B_2, where (a) B_2 is stacked on B_1, (b) B_1 is stacked on B_2, (c) B_1 nests in B_2, and (d) B_2 nests in B_1.

Let the wings of B_1 and B_2 be f_1, f_1' and f_2, f_2', respectively. Here, f_1 and f_2 denote the lower wings and f_1' and f_2' denote the upper wings of their respective butterflies. Then the ordering of the four wings from bottom to top must be one of the following: (f_1, f_1', f_2, f_2'), (f_2, f_2', f_1, f_1'), (f_2, f_1, f_1', f_2') or (f_1, f_2, f_2', f_1'), as shown in Figure 3.3(a)–(d), respectively. Note that the ordering of the wings cannot be (f_1, f_2, f_1', f_2') or (f_2, f_1, f_2', f_1') as P would self-intersect. If the order of the wings is as in Figure 3.3(a) or (b), we say that B_1 and B_2 *stack*. Otherwise, we say that B_1 and B_2 *nest*.

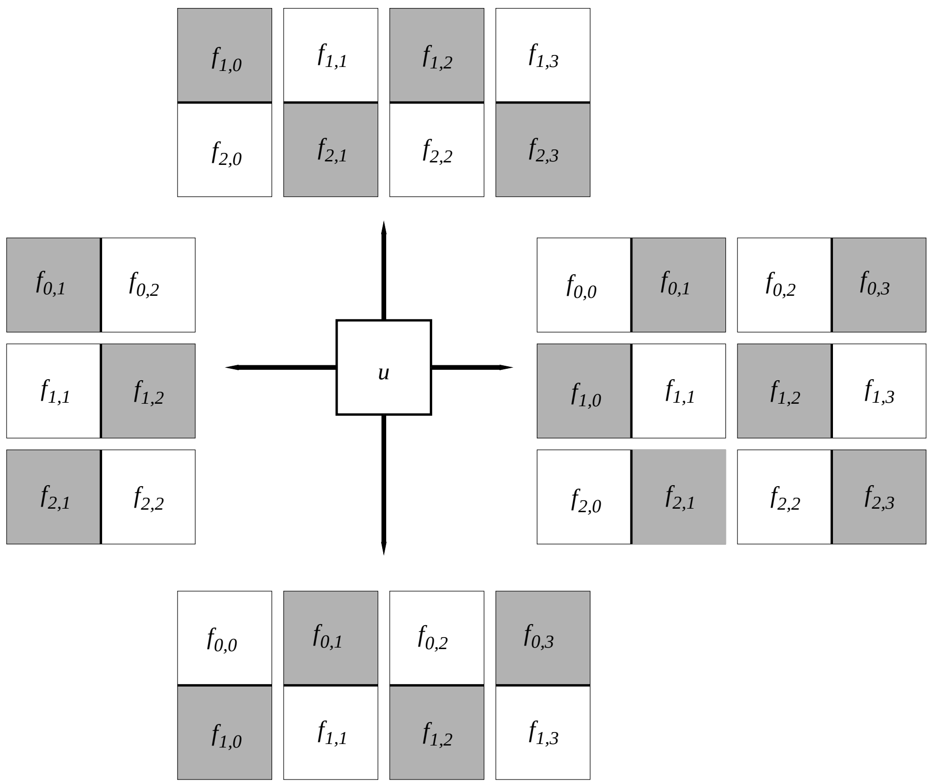


Figure 3.4: The unit square u on the XY-plane. The 3 west butterflies of $\mathbb{C}$ in Figure 3.2 are shown to the left of u. The 6 east butterflies of $\mathbb{C}$ are shown to the right of u. The 4 north butterflies of $\mathbb{C}$ are shown above u and the 4 south butterflies of $\mathbb{C}$ are shown below u.

If P is flat foldable, then there exists a final flat folded state of P where all the faces of $\mathbb{C}$ are lie above a unit square u on the XY-plane. In any final flat folded state (if one exists) of P, $v_{0,0}$ is incident to the top-left corner of u. Therefore, the horizontal creases in row r_i, $0 \leq i \leq m-1$, lie above the top edge of u when i is even. We call the butterflies that have these creases as hinges the *north butterflies*. Similarly, the horizontal creases in row r_i, $0 \leq i \leq m-1$, lie above the bottom edge of u when i is odd. We call butterflies with these hinges the *south butterflies*. The vertical edges in column c_j, $0 \le j \le n-1$, lie above the left edge of u when j is even and they lie above the right edge of u when j is odd. We call butterflies with those hinges the *west butterflies* and the *east butterflies*, respectively. Figures 3.4 shows all the north butterflies, east butterflies, south butterflies and west butterflies of the crease pattern in Figure 3.2 in clockwise order around the unit square u starting from the top.

A pair of butterflies B_1 and B_2 is called a pair of *twin butterflies* if both of them are north butterflies or south butterflies or east butterflies or west butterflies.

3.2.3 Recognizing Valid Linear Orderings

In this section, we describe our algorithm to decide whether a given linear ordering of the faces of $\mathbb{C}$ is a valid linear ordering.

Let L be any arbitrary linear ordering of the faces of $\mathbb{C}$. For each pair of twin butterflies B_1, B_2 in $\mathbb{C}$, we check whether B_1, B_2 nest, stack or intersect in L. If each pair of twin butterflies in $\mathbb{C}$ either stacks or nests in L, then L is a valid linear ordering. Otherwise, it is not a valid linear ordering. The formal description of the algorithm is given below.

Algorithm 3: RecogUnassignVLO(L, m, n)

L is the input linear ordering of the faces of an $m \times n$ map
if *there are more or less than* mn *faces in* L **then**
  return *false*
for *each pair of twin butterflies* B_1, B_2 **do**
  if B_1 *and* B_2 *intersect* **then**
    return *false*

The following theorem proves the correctness of Algorithm 3.

Theorem 3.1. ***(Butterfly Condition)*** *Let P be a map with unassigned crease pattern $\mathbb{C}$. Let L be a linear ordering of the faces of $\mathbb{C}$. Then L is a valid linear ordering if and only if every pair of twin butterflies either stacks or nests in L.*

Proof. We first assume that L is a valid linear ordering. Then by definition, L gives a final flat folded state of P, where no pair of twin butterflies intersect each other. Therefore, each pair of twin butterflies either nests or stacks.

We now assume that each pair of twin butterflies either stacks or nests in L. We first decompose P into $m \times n$ distinct unit squares, where each square is a face of $\mathbb{C}$. Each of these squares has a light side and a dark side. We stack these squares on a unit square u according to the linear ordering L. The checkerboard pattern of $\mathbb{C}$ decides for each face whether it faces dark or light side up. For each north butterfly B in P, we join its two wings (along the hinge of B) such that its hinge lies above the top edge of u. Since any two north butterflies either nest or stack, there will be no intersection of butterflies. We join the wings of the south, east and west butterflies along the bottom, right and left edge of u in a similar way. In this way, we construct a final flat folded state S_f of P and L is the linear ordering of the faces of $\mathbb{C}$ in S_f. Therefore, L is a valid linear ordering. □

We now calculate the running time of Algorithm 3 and show that with careful implementation, the running time can be reduced to be linear in the size of the input.

Theorem 3.2. *The running time of Algorithm 3 is $O(m^2n^2)$. With a careful implementation, the running time can be reduced to $O(mn)$ which is linear in the size of the input.*

Proof. The number of north butterflies is $\lfloor \frac{m-1}{2} \rfloor \times n$. Similarly, the number of south butterflies is $\lceil \frac{m-1}{2} \rceil \times n$. The number of west butterflies is $\lfloor \frac{n-1}{2} \rfloor \times m$ and the number of east butterflies is $\lceil \frac{n-1}{2} \rceil \times m$. Therefore, there are $O(m^2n^2)$ pairs of twin north butterflies, $O(m^2n^2)$ pairs of twin south butterflies, $O(m^2n^2)$ pairs of twin west butterflies and $O(m^2n^2)$ pairs of twin east butterflies. Therefore, Algorithm 3 has to check $O(m^2n^2)$ pairs of butterflies. Since it takes $O(1)$ time to check whether a pair of butterflies stacks or nests, the total time for the checking is $O(m^2n^2) \times O(1) = O(m^2n^2)$.

We can reduce the time complexity from $O(m^2n^2)$ to $O(mn)$ by reducing the number of comparisons. Suppose that B_1, B_2 is a pair of twin butterflies such that B_1 nests in B_2 in L. Let B_1, B_3 be another pair of twin butterflies. If B_3 nests in B_1 in L, then it must also nest in B_2. Therefore, we do not need to check the pair of twin butterflies B_2, B_3 for intersection. We now apply this idea, which is similar to balanced parentheses checking, to give an improved implementation of our algorithm as follows.

We first show how to check the north butterflies for intersection. Let B be a north butterfly. Then its two wings must be the faces $f_{i,j}$ and $f_{i+1,j}$, where $1 \leq i \leq m-2$, $0 \leq j \leq n-1$ and i is odd. We now take a two dimensional array $M[0 \ldots m-1][0 \ldots n-1]$ and a stack $S[1 \ldots mn]$. At first the stack is empty and each of the entries in M is 0. We take the faces in the order given by L and process them as follows. Let the current face be $f_{x,y}$. We consider the following steps.

- If $x < 1$, or $x > m-2$ and m is even, then $f_{x,y}$ is not a wing of a north butterfly. Therefore, we proceed to the next face in L. Otherwise, we examine $M[x, y]$.
  - (a) If the entry $M[x, y]$ contains 0, then it is the wing of a butterfly that occurs before the other wing. We push the face $f_{x,y}$ to S. If x is odd, then we set the entry $M[x+1, y] = 1$, and if x is even, then we set $M[x-1, y] = 1$.
  - (b) If $M[x, y]$ contains 1, then the other wing of the corresponding butterfly is already in the stack. In this case, we check the top of the stack. If the topmost face in the stack is $f_{x+1,y}$ (x is odd) or $f_{x-1,y}$ (x is even), then we pop the topmost face and proceed to the next face. Otherwise, there is an intersection, and hence L is not a valid linear ordering.

We stop when either we detect an intersection or we reach the end of L. Therefore, checking for intersection between any pair of north butterflies takes $O(mn)$ time.

We next check the south, west and east butterflies in a similar way. For a south butterfly, the two wings are the faces $f_{i,j}$ and $f_{i+1,j}$, where $0 \le i \le m-2$, $0 \le j \le n-1$ and i is even. For a west butterfly, the two wings are the faces $f_{i,j}$ and $f_{i,j+1}$, where $0 \le i \le m-1$, $1 \le j \le n-2$ and j is odd. For an east butterfly, the two wings are the faces $f_{i,j}$ and $f_{i,j+1}$, where $0 \le i \le m-1$, $0 \le j \le n-2$ and j is even. The total running time of the algorithm is $4 \times O(mn) = O(mn)$, which is linear in the size of the input linear ordering L since L is a linear ordering of mn faces. □

The following corollary easily follows from Theorem 3.2.

Corollary 3.1. *Let $\mathbb{C}$ be a $2 \times n$ unassigned crease pattern and let L be a linear ordering of the faces of $\mathbb{C}$. Then Algorithm 3 decides whether L is a valid linear ordering in $O(n)$ time.*

We now give an example to show how the algorithm works.

Assume that for the 3×4 crease pattern in Figure 3.2, we are given a linear ordering $L = (f_{0,0}, f_{1,0}, f_{0,1}, f_{2,1}, f_{1,1}, f_{1,2}, f_{2,0}, f_{2,2}, f_{0,2}, f_{0,3}, f_{1,3}, f_{2,3})$. We want to check whether L is a valid linear ordering. We will show how to check for intersection for the north butterflies shown in Figure 3.5(a).

In Figure 3.5, we show the states of array M and stack S when we check the north butterflies for intersection. We start from face $f_{0,0}$, which we ignore since it is not a wing of a north butterfly (see Figure 3.5(b)). We then go to the next face $f_{1,0}$, which is the wing of a north butterfly that has occurred before the other wing $f_{2,0}$. Therefore, we push $f_{1,0}$ in the stack and set $M[2,0] = 1$, as shown in Figure 3.5(c). The algorithm skips $f_{0,1}$ (Figure 3.5(d)) and then pushes $f_{2,1}$ in the stack (Figure 3.5(e)). The next face is $f_{1,1}$, where the other wing $f_{2,1}$ of the same butterfly is already on the top of the stack. Therefore, we pop $f_{2,1}$ as in Figure 3.5(f). When we are at the face $f_{2,0}$ (Figure 3.5(h)), the other wing $f_{1,0}$ is not the topmost face in the stack. Here, we have detected an intersection between the north butterflies with wings $f_{1,0}, f_{2,0}$ and $f_{1,2}, f_{2,2}$. Therefore, L is not a valid linear ordering.

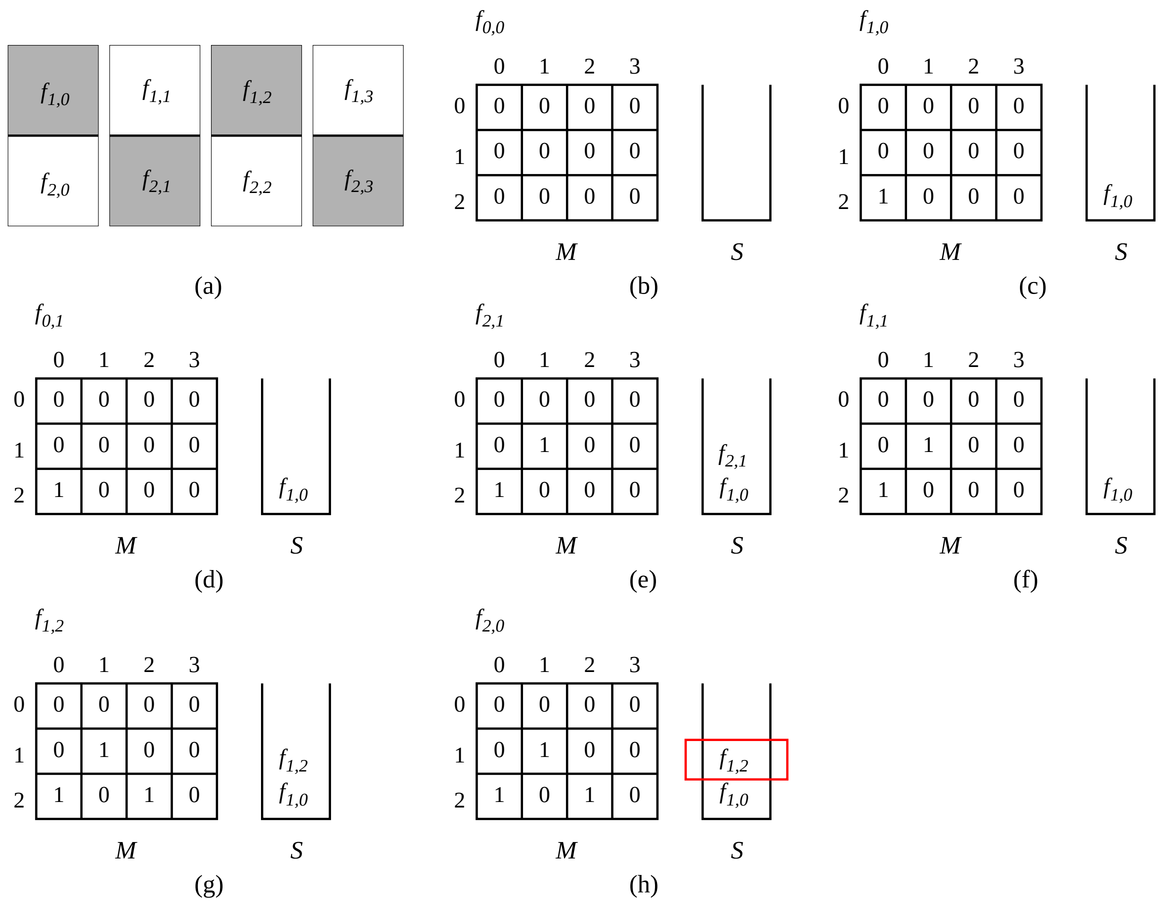


Figure 3.5: (a) The north butterflies of the crease pattern in Figure 3.2. (b)–(h) The steps of Algorithm 3.

3.3 Mountain-Valley Patterns

Let P be an $m \times n$ map and let $\mathbb{C}$ be the mountain-valley pattern on P. In this section, we show that we can decide whether a given linear ordering of the faces of $\mathbb{C}$ is a valid linear ordering in $O(mn)$ time. We first introduce the concept of "directed networks". We then use this concept to recognize valid linear orderings of $\mathbb{C}$ by modifying Algorithm 3 from the previous section.

3.3.1 Directed Networks

Let B be a butterfly of P with hinge e and wings f, f'. Since f and f' are adjacent faces, exactly one of them has light side up in the checkerboard pattern. Without loss of generality, we assume that f has the light side up. Then the label of e (mountain or valley) determines the *ordering* of f and f'. If e has the label mountain, then f (the face with the light side up) comes above f' (the face with dark side up). We denote the ordering by $f \prec f'$, where $\prec$ means 'comes above'. On the other hand, if e has the label valley, then the ordering is $f' \prec f$. The labels of all the creases give a *directed network* (of the faces) of $\mathbb{C}$. For example, Figure 3.6(a) shows a 2×2 mountain-valley pattern $\mathbb{C}$ and Figure 3.6(b) shows the checkerboard pattern of the faces of $\mathbb{C}$. The creases e_1, e_2, e_3 and e_4 impose the following directed network. (See

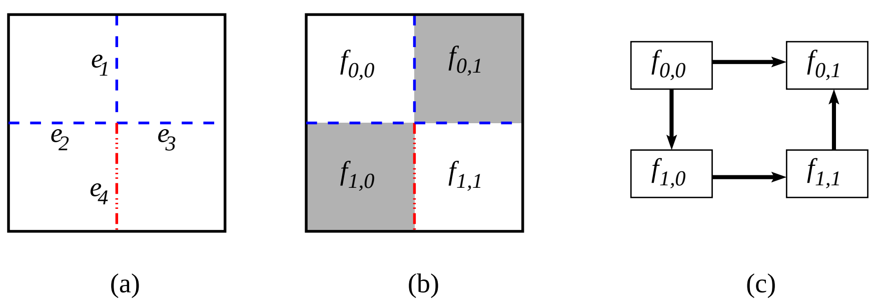


Figure 3.6: (a) A 2×2 mountain-valley pattern, (b) the checkerboard pattern, and (c) the directed network.

Figure 3.6(c).)

1. $f_{0,1}$ is facing dark side up and $f_{0,0}$ is facing light side up. Since e_1 has the label mountain, then $f_{0,0} \prec f_{0,1}$.

2. $f_{0,1}$ is facing dark side up and $f_{1,1}$ is facing light side up. Since e_3 has the label mountain, then $f_{1,1} \prec f_{0,1}$.

3. $f_{1,1}$ is facing light side up and $f_{1,0}$ is facing dark side up. Since e_4 has the label valley, then $f_{1,0} \prec f_{1,1}$.

4. $f_{0,0}$ is facing light side up and $f_{1,0}$ is facing dark side up. Since e_2 has the label mountain, then $f_{0,0} \prec f_{1,0}$.

Since $f_{0,0}$ comes above all other faces, it must be the topmost face. Similarly, the bottommost face must be $f_{0,1}$. In fact, in this particular example, the directed network gives a unique candidate for a valid linear ordering of the faces of $\mathbb{C}$, which is $L = (f_{0,0}, f_{1,0}, f_{1,1}, f_{0,1})$ from top to bottom. Notice that the directed network in Figure 3.6(c) is a directed acyclic graph (DAG). But such a directed network may contain cycles, as shown in Figure 3.7. We now prove that the directed network of

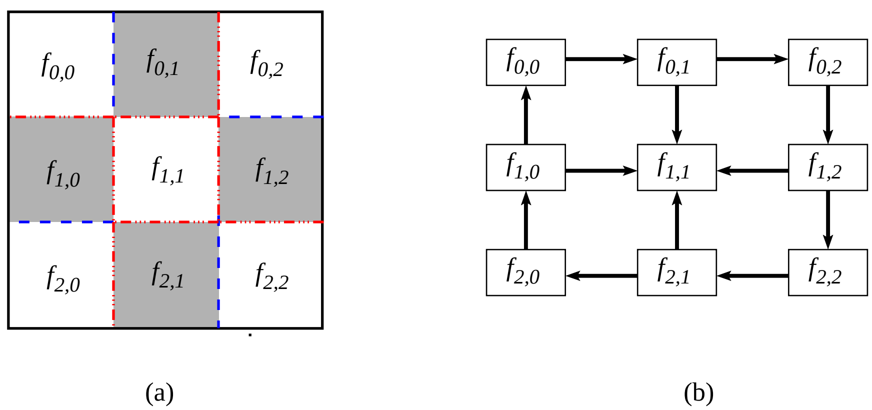


Figure 3.7: (a) A 3×3 mountain-valley pattern, and (b) its directed network that contains a cycle $f_{0,0} \prec f_{0,1} \prec f_{0,2} \prec f_{1,2} \prec f_{2,2} \prec f_{2,1} \prec f_{2,0} \prec f_{1,0} \prec f_{0,0}$.

any flat foldable $m \times n$ mountain-valley pattern must be a DAG.

Lemma 3.1. *Let $\mathbb{C}$ be an $m \times n$ mountain-valley pattern. If $\mathbb{C}$ is flat foldable, then its directed network $\mathcal{N}$ is a directed acyclic graph.*

Proof. Suppose for a contradiction that $\mathcal{N}$ contains a cycle $f_1 \prec f_2 \prec f_3 \prec \ldots \prec f_k \prec f_1$. Since $\mathbb{C}$ is flat foldable, there is a final flat folded state that gives a valid linear ordering L of the faces of $\mathbb{C}$. Let L' be the ordering of the faces $f_1, f_2, \ldots, f_k$ within L. Since L' is a linear ordering of faces, it cannot satisfy all the constraints imposed by a cycle. Consequently, the ordering of the faces in L' cannot satisfy $f_1 \prec f_2 \prec f_3 \prec \ldots \prec f_k \prec f_1$. □

3.3.2 Recognizing Valid Linear Orderings

In this section, we give an algorithm to decide whether a given linear ordering of the faces of a mountain-valley pattern $\mathbb{C}$ is a valid linear ordering of $\mathbb{C}$.

Let L be any linear ordering of the faces of $\mathbb{C}$. For each pair of twin butterflies B_1, B_2 in $\mathbb{C}$, we check whether B_1, B_2 nest, stack or intersect in L. If they either stack or nest, then we check whether the ordering of the wings of B_1 and B_2 satisfies the ordering in the directed network. If each pair of twin butterflies satisfies the ordering in the directed network and does not intersect, then L is a valid linear ordering. Otherwise, it is not a valid linear ordering. The formal description of the algorithm is given below.

Algorithm 4: RecogAssignVLO($L, m, n, \mathcal{N}$)

L is the input linear ordering of the faces of an $m \times n$ mountain-valley pattern
$\mathcal{N}$ is the directed network of the mountain-valley pattern
if *there are more or less than mn faces in L* **then**
    return *false*
for *each pair of twin butterflies B_1, B_2* **do**
    if *B_1 and B_2 intersect* **then**
        return *false*
    else
        if *the ordering of the wings of B_1 and B_2 does not satisfy $\mathcal{N}$* **then**
            return *false*

The following theorem proves the correctness of Algorithm 4.

Theorem 3.3. *Let P be a map with the mountain-valley pattern $\mathbb{C}$. Let L be a linear ordering of the faces of $\mathbb{C}$. Then L is a valid linear ordering if and only if (a)–(b) hold: (a) every pair of twin butterflies either stacks or nests in L (i.e., satisfies the Butterfly Condition), and (b) L satisfies the directed network $\mathcal{N}$ of $\mathbb{C}$.*

Proof. We first assume that L is a valid linear ordering of $\mathbb{C}$. Then Conditions (a) and (b) must hold. Therefore, we assume that Conditions (a) and (b) hold. Then in a similar way as in the proof of Theorem 3.1, we can decompose P into unit squares and construct a final flat folded state using the linear ordering L. Since no pair of twin butterflies intersect in L, we can always construct such a final flat folded state. Therefore, L is a valid linear ordering. □

We now calculate the running time of Algorithm 4 and show that with careful implementation, the running time can be linear in the size of the input.

Theorem 3.4. *The running time of Algorithm 4 is $O(m^2n^2)$. With a careful implementation, the running time can be reduced to $O(mn)$ which is linear in the size of the input.*

Proof. Since there are $O(m^2n^2)$ pairs of twin butterflies, and it takes $O(1)$ time to check whether a pair of twin butterflies intersect and whether the order of the wings of each butterfly satisfies the ordering given by the directed network, the total running time of Algorithm 4 is $O(m^2n^2) \times O(1) = O(m^2n^2)$.

We now show a careful implementation to reduce the time complexity. We first check for each pair of north butterflies whether they intersect or not. We take a two dimensional array $M[0 \dots m-1][0 \dots n-1]$ and a stack $S[1 \dots mn]$. At first the stack is empty and each of the entries in M is 0. In the previous section we updated M during the processing of each face in L. Here we preprocess M based on the directed network of $\mathbb{C}$, and M remains unchanged during the processing of the faces. Here are the rules for preprocessing M.

For each $1 \leq i \leq m-2$, where i is odd, and for each $0 \leq j \leq n-1$, we do the following:

- If $f_{i,j}$ faces light side up in the checkerboard pattern of $\mathbb{C}$ and the crease between $f_{i,j}, f_{i+1,j}$ is labeled mountain, then set $M[i+1, j] = 1$. This means that the face $f_{i,j}$ must occur in L before the face $f_{i+1,j}$.

- If $f_{i,j}$ faces dark side up in the checkerboard pattern of $\mathbb{C}$ and the crease between $f_{i,j}, f_{i+1,j}$ is labeled mountain, then set $M[i, j] = 1$.

- If $f_{i,j}$ faces light side up in the checkerboard pattern of $\mathbb{C}$ and the crease between $f_{i,j}, f_{i+1,j}$ is labeled valley, then set $M[i, j] = 1$.

- If $f_{i,j}$ faces dark side up in the checkerboard pattern of $\mathbb{C}$ and the crease between $f_{i,j}, f_{i+1,j}$ is labeled valley, then set $M[i+1, j] = 1$.

We now take the faces in the order given by L and process them as follows. Let the current face be $f_{x,y}$.

- If $x < 1$, or $x > m-2$ and m is even, then it is not a wing of a north butterfly. Therefore, we proceed to the next face in L. Otherwise, it is the wing of a north butterfly and we examine $M[x, y]$.

– If $M[x, y] = 0$, then it is the wing of a butterfly that occurs before the other wing. We push the face $f_{x,y}$ to S.

– If $M[x, y] = 1$, the other wing of the corresponding butterfly is already in the stack. In this case, we check the top of the stack. If the topmost face in the stack is $f_{x+1,y}$ (x is odd) or $f_{x-1,y}$ (x is even), then we pop the topmost face and proceed to the next face. Otherwise, there is an intersection, and hence L is not a valid linear ordering.

We stop when either we detect an intersection or we reach the end of L. Therefore, checking for intersection among the north butterflies takes $O(mn)$ time. We next check the south, west and east butterflies in a similar way. The total running time of the algorithm is $4 \times O(mn) = O(mn)$ which is linear in the size of the input linear ordering L. □

3.4 Enumerating Valid Linear Orderings

In this section, we sketch the outline of an exponential-time exact algorithm to enumerate all the valid linear orderings of an $m \times n$ crease pattern $\mathbb{C}$.

We first assume that $\mathbb{C}$ is a mountain-valley pattern. Then there is a unique directed network $\mathcal{N}$ of $\mathbb{C}$. We assume that $\mathcal{N}$ is a directed acyclic graph; otherwise $\mathbb{C}$ is not flat foldable by Lemma 3.1. We now enumerate all the linear orderings of the faces of $\mathbb{C}$ using the algorithm of [32], which takes constant amortized time; i.e., the total running time of the algorithm is $O(e(\mathcal{N}))$, where $e(\mathcal{N})$ is the number of linear orderings generated by the algorithm from the partial order $\mathcal{N}$. We can decide whether a linear ordering is a valid linear ordering in $O(mn)$ time. From Theorem 1.1 of [20], we know that $e(\mathcal{N}) \leq 2^{mn(log(mn)-H(\mathcal{N}))} \leq 2^{mnlog(mn)}$, where $H(\mathcal{N}) \leq \log mn$ is the entropy function of $\mathcal{N}$. Therefore, enumerating all valid linear orderings takes $O(mn) \times O(2^{mn\log(mn)}) = O(mn \cdot 2^{mn\log(mn)})$ time.

We demonstrate how the algorithm works on a mountain-valley pattern for a small example. Let $\mathbb{C}$ be the 2×3 mountain-valley pattern shown in Figure 3.8(a). From its partial order diagram in Figure 3.8(b), we get two partial orderings $f_{0,0} \prec f_{1,0} \prec f_{1,1} \prec f_{0,1}$ and $f_{0,2} \prec f_{1,2} \prec f_{1,1} \prec f_{0,1}$. Combining these two partial ordering diagrams, we get the following linear orderings of the faces.

(a) $f_{0,2} \prec f_{1,2} \prec f_{0,0} \prec f_{1,0} \prec f_{1,1} \prec f_{0,1}$ (a valid linear ordering)

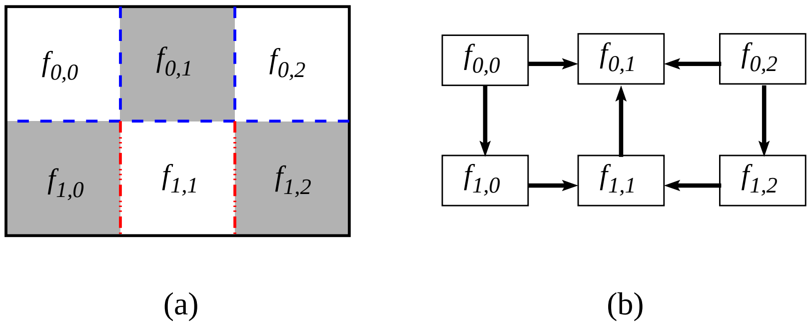


Figure 3.8: (a) A 2×3 mountain-valley pattern, and (b) its directed network.

(b) $f_{0,2} \prec f_{0,0} \prec f_{1,2} \prec f_{1,0} \prec f_{1,1} \prec f_{0,1}$ (not a valid linear ordering)

(c) $f_{0,2} \prec f_{0,0} \prec f_{1,0} \prec f_{1,2} \prec f_{1,1} \prec f_{0,1}$ (a valid linear ordering)

(d) $f_{0,0} \prec f_{0,2} \prec f_{1,2} \prec f_{1,0} \prec f_{1,1} \prec f_{0,1}$ (a valid linear ordering)

(e) $f_{0,0} \prec f_{0,2} \prec f_{1,0} \prec f_{1,2} \prec f_{1,1} \prec f_{0,1}$ (not a valid linear ordering)

(f) $f_{0,0} \prec f_{1,0} \prec f_{0,2} \prec f_{1,2} \prec f_{1,1} \prec f_{0,1}$ (a valid linear ordering)

Therefore, there are four valid linear orderings of $\mathbb{C}$.

We now enumerate all the valid linear orderings of $\mathbb{C}$ when $\mathbb{C}$ is an unassigned crease pattern. Without loss of generality, we assume that $n \geq m$. One can enumerate all the mountain-valley assignments of $\mathbb{C}$ using the local flat foldability property. There are $m(n-1)$ vertical creases and $(m-1)n$ horizontal creases in $\mathbb{C}$. Without loss of generality we first assign labels (mountain or valley) to all the horizontal creases and to the $n-1$ vertical creases of row r_0. For all the vertical creases in the other rows, we have to choose the appropriate labels to keep the pattern locally flat foldable. Therefore, there can be $2^{(m-1)n+n-1} = 2^{mn-1}$ locally flat foldable mountain-valley assignments for $\mathbb{C}$. The corresponding directed networks may not be acyclic. We can check whether a directed network is a directed acyclic graph (DAG) in $O(mn)$ time [24]. Recall that for each of the mountain-valley assignments, we can enumerate all valid linear orderings in $O(mn \cdot 2^{mn \log(mn)})$ time. Therefore, it takes $O(mn \cdot 2^{(mn \log(mn))} \cdot 2^{mn-1})$ time to enumerate all valid linear orderings of $\mathbb{C}$.

3.5 Conclusion

In this chapter, we gave Algorithm 3 to decide whether a linear ordering L of the faces of an $m \times n$ unassigned crease pattern is a valid linear ordering, and Algorithm 4 to decide whether a linear ordering L of the faces of an $m \times n$ mountain-valley pattern is a valid linear ordering. We proved that, if L is a valid linear ordering, then it satisfies the Butterfly Condition (Theorem 3.1 and Theorem 3.3). We also gave an exponential-time exact algorithm to enumerate all valid linear orderings of a given crease pattern (assigned or unassigned).

As a final remark for this chapter, we now discuss generalization of the Butterfly Condition for any flat folded state of a map P.

Butterfly Condition Generalization: Let u_1, u_2 be two unit squares on the XY-plane adjacent along an edge e. Let S be any flat folded state (not necessarily a final flat folded state) of P. Let B_1 and B_2 be any two butterflies of P such that their hinges lie above e. Let the wings of B_1 and B_2 be f_1, f_1' and f_2, f_2', respectively. Figures 3.9(a)–(c) show B_1, B_2 and e on the XY-plane, respectively. Since S is a flat

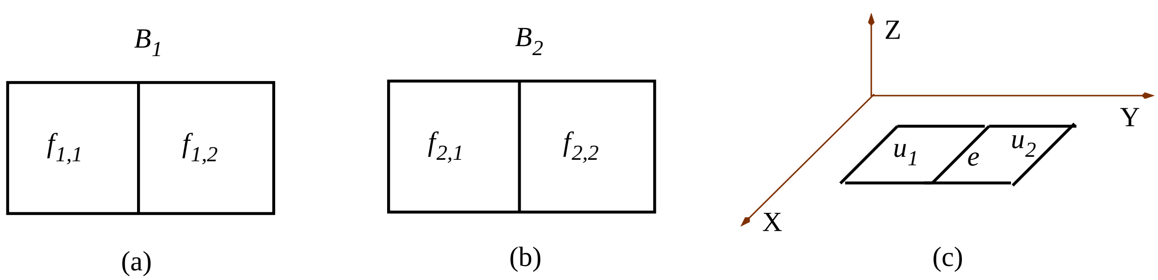


Figure 3.9: (a) B_1, (b) B_2, (c) e and u_1, u_2, (d)–(f) invalid orderings of faces in a flat folded state.

folded state, the following conditions must be satisfied to ensure that B_1 and B_2 do not intersect each other. We call these conditions the *generalized butterfly conditions.*

$f_{1,1}, f_{2,1}$ **lie above the unit square** u_1 **and** $f_{1,2}, f_{2,2}$ **lie above** u_2. In this case, if the ordering of the faces $f_{1,1}, f_{2,1}$ from top to bottom is $f_{1,1}, f_{2,1}$ then the ordering of the faces $f_{1,2}, f_{2,2}$ from top to bottom must be $f_{1,2}, f_{2,2}$. Otherwise, the butterflies intersect as shown in Figure 3.10(a).

$f_{1,1}$ **lies above the unit square** u_1 **and** $f_{2,1}, f_{1,2}, f_{2,2}$ **lie above** u_2**.** In this case, the ordering of the faces $f_{2,1}, f_{1,2}, f_{2,2}$ from top to bottom must be one of the

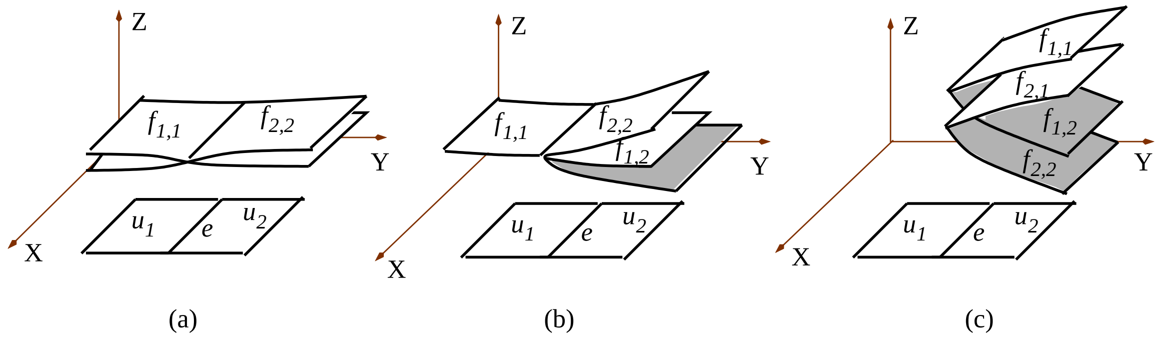


Figure 3.10: (a) B_1, (b) B_2, (c) e and u_1, u_2, (d)–(f) invalid orderings of faces in a flat folded state.

following: $(f_{1,2}, f_{2,1}, f_{2,2})$, $(f_{1,2}, f_{2,2}, f_{2,1})$, $(f_{2,1}, f_{2,2}, f_{1,2})$ or $(f_{2,2}, f_{2,1}, f_{1,2})$. An invalid ordering of the faces in this case is shown in Figure 3.10(b).

All the faces $f_{1,1}, f_{2,1}, f_{1,2}, f_{2,2}$ lie above the same unit square u_2. In this case, either B_1, B_2 stack or they nest, satisfying the Butterfly Condition (Theorem 3.1 and Theorem 3.3). An invalid ordering of the faces in this case is shown in Figure 3.10(c).

Chapter 4

Unfoldable Maps

An *unfoldable pattern* is a mountain-valley pattern that does not have a final flat folded state. In this chapter, we show all the 2×5 mountain-valley patterns that are not flat foldable, i.e., that are *unfoldable*. We then generalize those unfoldable patterns to define a class of $2 \times n$ unfoldable mountain-valley patterns, for every $n > 5$. We note that Justin [19] enumerated all the unfoldable $2 \times n$ mountain-valley patterns for $5 \leq n \leq 7$.

4.1 Preliminaries

In this section, we introduce the terminology that we use in this chapter.

4.1.1 $2 \times n$ Maps

Let P be a $2 \times n$ map with a mountain-valley pattern $\mathbb{C}$. By definition, there are n horizontal creases. We call each of these creases a *spinal crease* and collectively we call these creases the *spine*. We call the $n-1$ vertical creases above the spine the *upper ribs* and the remaining creases the *lower ribs*. We denote the upper ribs in $\mathbb{C}$ by $u_1, u_2, \ldots, u_{n-1}$ from left to right as shown in Figure 4.1. Similarly, the lower ribs are denoted by $l_1, l_2, \ldots, l_{n-1}$ from left to right, and the spinal creases are denoted by $s_1, s_2, \ldots, s_n$ from left to right,

Assume that $\mathbb{C}$ is flat foldable. Suppose some u_i and l_i, where $1 \leq i \leq n-1$, have the same label (mountain or valley). Then by Lemma 2.1, the label of s_i must be opposite of the label of s_{i+1}. Let $\mathcal{H}$ be the set of all the vertical homogeneous crease lines. We call $\mathcal{H}$ the *set of pre-spine folds*. For the crease pattern in Figure 4.1,

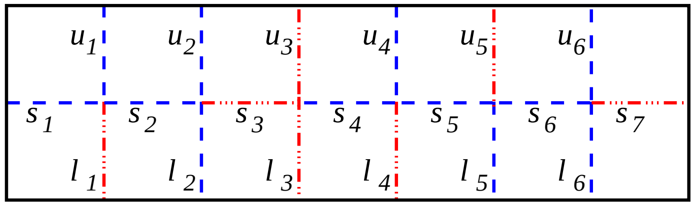


Figure 4.1: A 2×7 mountain-valley pattern.

$\mathcal{H} = \{\{u_2, l_2\}, \{u_3, l_3\}, \{u_6, l_6\}\}$, where $\{u_2, l_2\}, \{u_3, l_3\}$ and $\{u_6, l_6\}$ are the vertical homogeneous crease lines or the *pre-spine folds*.

4.1.2 Symmetric Closure

Let $\mathbb{C}$ be an $m \times n$ mountain-valley pattern on an $m \times n$ grid paper P. The *symmetric closure* [1] of $\mathbb{C}$ is the set of all the $m \times n$ mountain-valley patterns that can be obtained by applying zero or more of the following operations on $\mathbb{C}$.

1. **MirrorX:** Taking the mirror image of $\mathbb{C}$ with respect to the X axis.
2. **MirrorY:** Taking the mirror image of $\mathbb{C}$ with respect to the Y axis.
3. **Switch:** Switching the mountain labels to valley and the valley labels to mountain.

Figure 4.2 shows the symmetric closure of a 2×3 mountain-valley pattern. Note that since we are looking down from the positive Z-axis, the X-axis is parallel to the left side of the paper and the Y-axis is parallel to the top of the paper as shown in Figure 4.2(a).

We now have the following observation.

Observation 4.1. *Let $\mathbb{C}$ be a mountain-valley pattern on an $m \times n$ grid paper P and let $C_s(\mathbb{C})$ be the symmetric closure of $\mathbb{C}$. Then the following (a)–(d) hold for $C_s(\mathbb{C})$.*

(a) When applying multiple operations on P, changing the order of operations does not give a new mountain-valley pattern.

(b) The symmetric closure of any mountain-valley pattern $\mathbb{C}_1 \in C_s(\mathbb{C})$ is $C_s(\mathbb{C}_1) = C_s(\mathbb{C})$.

[1] The term is inspired by the term used for binary relations.

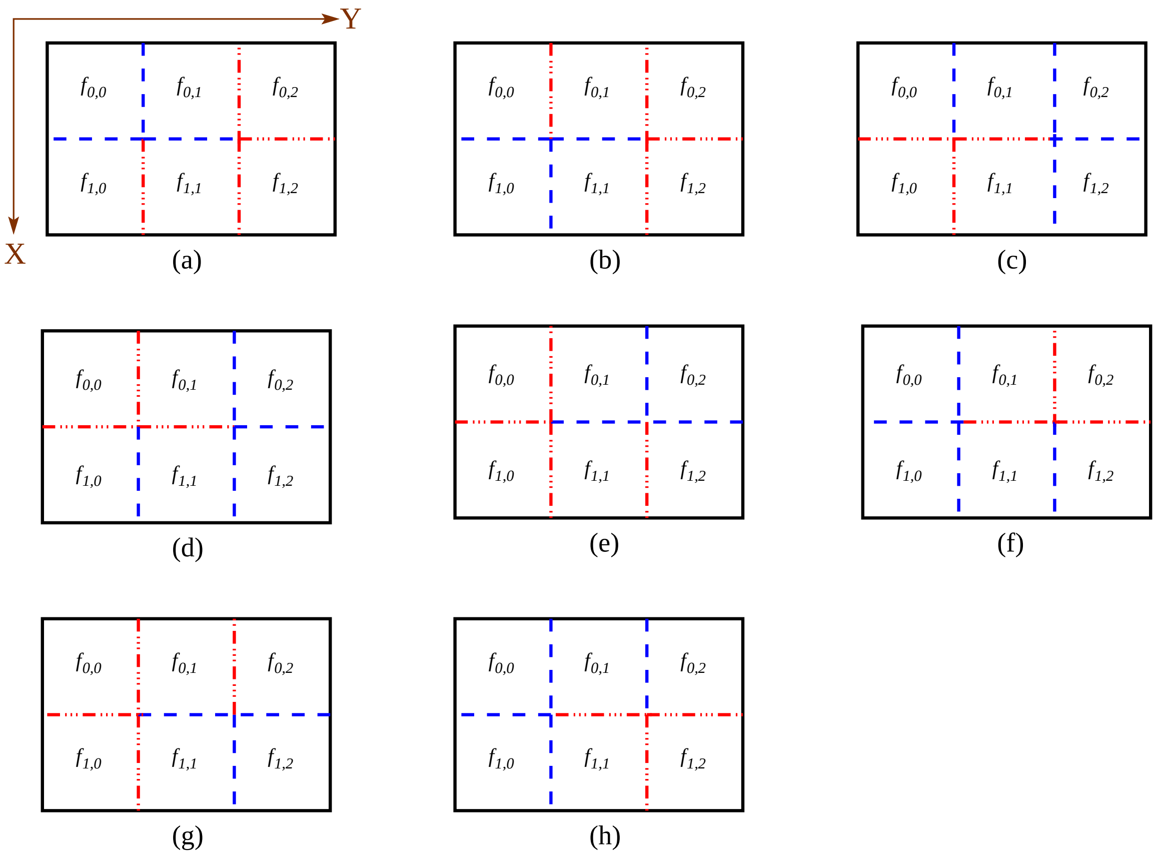


Figure 4.2: (a) A 2×3 mountain-valley pattern $\mathbb{C}$, (b) after applying MirrorX on $\mathbb{C}$, (c) after applying MirrorX and Switch on $\mathbb{C}$, (d) after applying switch on $\mathbb{C}$, (e) after applying MirrorY on $\mathbb{C}$, (f) after applying MirrorY and Switch on $\mathbb{C}$, (g) after applying MirrorX and MirrorY on $\mathbb{C}$, (h) after applying MirrorX, MirrorY and Switch on $\mathbb{C}$.

(c) If $\mathbb{C}$ is flat foldable, then all the mountain-valley patterns in $C_s(\mathbb{C})$ are flat foldable.

(d) If $\mathbb{C}$ is unfoldable, then all the mountain-valley patterns in $C_s(\mathbb{C})$ are unfoldable.

4.2 Folding Small Patterns

In this section, we prove that any $2 \times n$ locally flat foldable mountain-valley pattern $\mathbb{C}$ is flat foldable when $n \leq 4$. *Note that all the mountain-valley patterns in the rest of this chapter are locally flat foldable.* We first prove that any 2×2 mountain-valley pattern is flat foldable. In fact, the following lemma shows that any 2×2 mountain-valley pattern is simply flat foldable.

Lemma 4.1. *Let $\mathbb{C}$ be a 2×2 (locally flat foldable) mountain-valley pattern. Then $\mathbb{C}$ is simply flat foldable.*

Proof. If $\mathbb{C}$ is locally flat foldable, then it satisfies Lemma 2.1. Therefore, exactly three of the four creases in $\mathbb{C}$ must be labeled either mountain or valley and the remaining crease must be labeled either valley or mountain, respectively. If the spinal creases have the same labels, then we first apply a one-layer simple fold along the spine and then an all-layers simple fold along the ribs. Otherwise, we apply a one-layer simple fold along the ribs and then an all-layers simple fold along the spine. Therefore, $\mathbb{C}$ is simply flat foldable. □

We now show that 2×3 mountain-valley patterns are simply flat foldable.

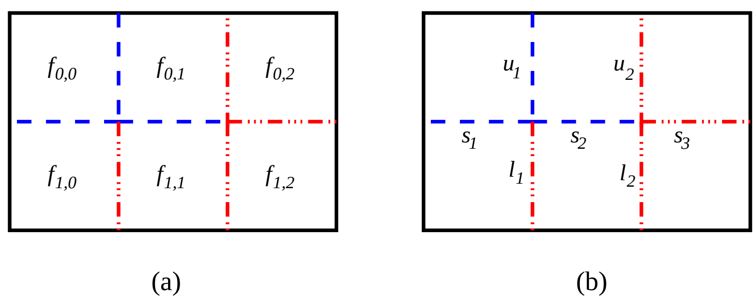


Figure 4.3: (a) A 2×3 mountain-valley pattern, where the faces are labeled, (b) the same mountain-valley pattern, where the ribs and spinal creases are labeled.

Lemma 4.2. *Let $\mathbb{C}$ be a 2×3 mountain-valley pattern. Then $\mathbb{C}$ is simply flat foldable.*

Proof. Depending on the number of pre-spine folds, there are two cases to consider.

(a) There is no pre-spine fold, i.e., $\mathcal{H} = \emptyset$. By Lemma 2.1, every vertex of $\mathbb{C}$ must have either three mountain creases and one valley crease or three valley creases and one mountain crease. Since at any vertex v of $\mathbb{C}$, the ribs have opposite labels, the spinal creases incident to v must have the same label. Therefore, all the spinal creases in $\mathbb{C}$ must have the same label (mountain or valley). We first apply a one-layer simple fold along the spine. Since $\mathbb{C}$ is locally flat foldable, the label of u_1 must be opposite to the label of l_1 and the label of u_2 must be opposite to the label of l_2. Therefore, after applying the simple fold along the spine all the creases match up (i.e., all the ribs that coincide have the same mountain or valley label when looking from the positive Z-axis, see Section 2.3.2), and the 2×3 mountain-valley pattern $\mathbb{C}$ reduces to a 1×3 mountain-valley pattern $\mathbb{C}'$.

Since the creases are equidistantly spaced in $\mathbb{C}'$, then $\mathbb{C}'$ is simply flat foldable by Lemma 2.3. Therefore, $\mathbb{C}$ is simply flat foldable.

(b) There is at least one pre-spine fold. Let $\{u_1, l_1\}$ be a pre-spine fold. Since $\mathbb{C}$ is locally flat foldable and u_1, l_1 have the same label, it follows from Lemma 2.1 that the label of s_1 must be opposite to the label of s_2. We first apply a one-layer simple fold along the crease line $\{u_1, l_1\}$. After applying the simple fold, s_1 matches up with s_2 (i.e., s_1 and s_2 lie on one another and have the same label when looking from the positive Z-axis). Therefore, $\mathbb{C}$ is reduced to a 2×2 mountain-valley pattern $\mathbb{C}''$ which is simply flat foldable by Lemma 4.1. Therefore, $\mathbb{C}$ is simply flat foldable.

□

We now prove that any 2×4 mountain-valley pattern is flat foldable.

Lemma 4.3. *Let $\mathbb{C}$ be a 2×4 mountain-valley pattern. Then $\mathbb{C}$ is flat foldable.*

Proof. We consider the following three cases.

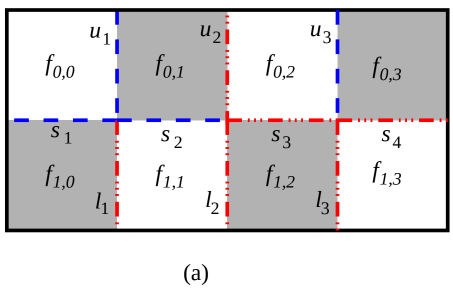
(a) **No pre-spine fold i.e., $\mathcal{H} = \emptyset$:** In a similar way as in Case 1 of the proof of Lemma 4.2, we apply a one-layer simple fold along the spine. This reduces $\mathbb{C}$ to a 1×4 mountain-valley pattern $\mathbb{C}_1$. Since $\mathbb{C}_1$ is simply flat foldable by Lemma 2.3, $\mathbb{C}$ is simply flat foldable.

(b) **At least two pre-spine folds:** In this case, $\mathcal{H}$ must contain at least one of of the crease lines $\{u_1, l_1\}$ and $\{u_3, l_3\}$. Without loss of generality we assume that $\{u_1, l_1\} \in H$. Then the label of s_1 must be opposite to the label of s_2 to preserve local flat foldability. We apply a one-layer simple fold along $\{u_1, l_1\}$, and then s_1 matches up with s_2. This fold reduces $\mathbb{C}$ to a 2×3 mountain-valley pattern $\mathbb{C}_2$. Since $\mathbb{C}_2$ is simply flat foldable by Lemma 4.2, $\mathbb{C}$ is simply flat foldable.

(c) **Exactly one pre-spine fold:** Assume first that the pre-spine fold is either $\{u_1, l_1\}$ or $\{u_3, l_3\}$. We then apply a one-layer simple fold along the pre-spine fold and reduce $\mathbb{C}$ to a 2×3 mountain-valley pattern $\mathbb{C}_3$. Since $\mathbb{C}_3$ is simply flat foldable by Lemma 4.2, $\mathbb{C}$ is also simply flat foldable.

We now assume that the pre-spine fold is $\{u_2, l_2\}$. If the label of u_1 is opposite to the label of u_3, then the label of l_1 must be opposite to the label of l_3 to preserve local flat foldability. Furthermore, s_1, s_2 must have the same labels which are opposite to the labels of s_3, s_4. We first apply a one-layer simple fold along $\{u_2, l_2\}$. After applying the simple fold, all the creases in $\mathbb{C}$ match up in the flat folded state. Then $\mathbb{C}$ is reduced to a 2×2 mountain-valley pattern, which is simple flat foldable by Lemma 4.1. Therefore, $\mathbb{C}$ is simply flat foldable.

We now show that even if u_1 has the same label as u_3, $\mathbb{C}$ has a final flat folded state. Without loss of generality, we assume that $\mathbb{C}$ is the 2×4 mountain-valley pattern shown in Figure 4.4(a). Any other 2×4 mountain-valley pattern such that $\{u_2, l_2\}$ is the only pre-spine fold and u_1 has the same label as u_3 must be in the symmetric closure of $\mathbb{C}$. We get a directed network of $\mathbb{C}$ as follows. The



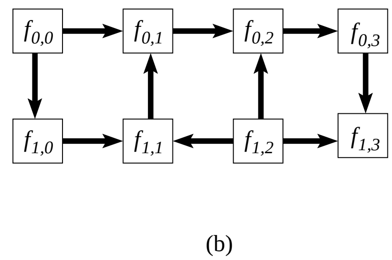


Figure 4.4: (a) A 2×4 mountain-valley pattern $\mathbb{C}$. $\mathbb{C}$ is flat foldable but the final flat folded state cannot be achieved by simple folding, (b) the directed network of $\mathbb{C}$.

crease u_1 has label mountain, which implies that $f_{0,0} \prec f_{0,1}$. On the other hand u_2 has label valley, which implies that $f_{0,1} \prec f_{0,2}$. In a similar way, we process each crease and the pair of faces incident to that crease. The resulting directed network is shown in Figure 4.4(b).

From Figure 4.4(b), we observe that the directed network is acyclic. From the directed network, we get the partial ordering $f_{0,0} \prec f_{1,0} \prec f_{1,1} \prec f_{0,1} \prec f_{0,2} \prec f_{0,3} \prec f_{1,3}$ of all the faces of $\mathbb{C}$ except $f_{1,2}$. Since $f_{1,2} \prec f_{1,1}$, we can place $f_{1,2}$ just before $f_{1,1}$ or just before $f_{1,0}$ or before $f_{0,0}$. If we place $f_{1,2}$ just before $f_{1,1}$ (i.e., between $f_{1,0}$ and $f_{1,1}$), then the linear ordering $f_{0,0} \prec f_{1,0} \prec f_{1,2} \prec f_{1,1} \prec f_{0,1} \prec f_{0,2} \prec f_{0,3} \prec f_{1,3}$ has intersection between the east butterflies l_1 and l_3 (i.e., butterflies with hinges at l_1 and l_3). If we we place $f_{1,2}$ just before $f_{1,0}$ (i.e., between $f_{0,0}$ and $f_{1,0}$), then the linear ordering $f_{0,0} \prec f_{1,2} \prec f_{1,0} \prec f_{1,1} \prec f_{0,1} \prec f_{0,2} \prec f_{0,3} \prec f_{1,3}$ has intersection between the east butterflies u_1 and l_3. Therefore, we place $f_{1,2}$ before $f_{0,0}$ (as the topmost face). We now check whether $L = f_{1,2} \prec f_{0,0} \prec f_{1,0} \prec f_{1,1} \prec f_{0,1} \prec f_{0,2} \prec f_{0,3} \prec f_{1,3}$ has any intersecting pair of twin butterflies. The west butterflies in $\mathbb{C}$ are u_1, u_3, l_1, l_3 (see Figure 4.5(a)).

The east butterflies are u_2, l_2 (see Figure 4.5(b)) and the south butterflies are s_1, s_2, s_3, s_4 (see Figure 4.5(c)). As Figure 4.5 shows, any pair of twin butterflies either stack or nest in L and hence, there is no intersection of butterflies in L. All the arcs in the figure is pointing from left to right, which proves that L satisfies the ordering of the wings of each butterfly in the directed network. Therefore, L

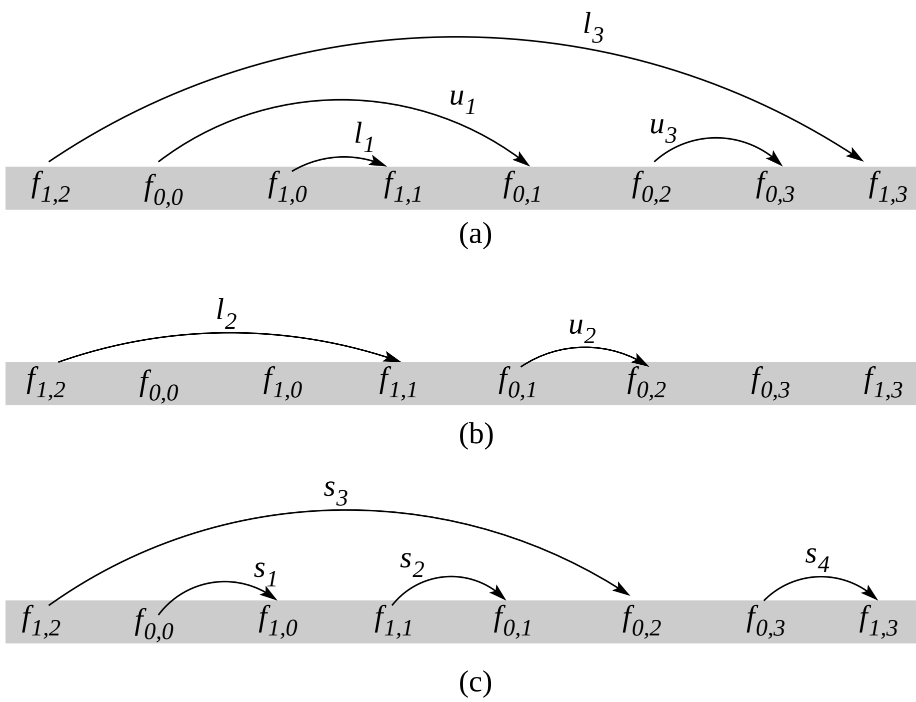


Figure 4.5: (a) The west butterflies, where the labeled arcs represent hinges, (b) the east butterflies, and (c) the south butterflies of the 2×4 mountain valley pattern in Figure 4.4(a).

is a valid linear ordering of $\mathbb{C}$ and $\mathbb{C}$ is flat foldable.

□

In the proofs of Lemmas 4.2 and 4.3, we observed that any 2×2, 2×3 and 2×4 locally flat foldable mountain-valley pattern is simply flat foldable when there is no pre-spine fold. In the following we prove that this observation holds for all $n \geq 2$.

Lemma 4.4. *Let $\mathbb{C}$ be a $2 \times n$ locally flat foldable mountain-valley pattern, where $n \geq 2$. If $\mathbb{C}$ does not have any pre-spine fold, then it is simply flat foldable.*

Proof. Since $\mathbb{C}$ does not have any pre-spine fold, all the spinal creases must have the same labels. Since $\mathbb{C}$ is locally flat foldable, the label of u_i must be opposite to the label of l_i for $1 \leq i \leq n-1$. We first apply a one-layer simple fold along the spine. After applying the fold, each u_i matches up with l_i, $1 \leq i \leq n-1$ and $\mathbb{C}$ is reduced to a $1 \times n$ mountain-valley pattern $\mathbb{C}'$. Since $\mathbb{C}'$ is simply flat foldable by Lemma 2.3, $\mathbb{C}$ is simply flat foldable. □

4.3 2×5 Mountain-Valley Patterns

In this section, we show that there exist eight 2×5 mountain-valley patterns that are not flat foldable. We first find some 2×5 mountain-valley patterns that are flat foldable in Section 4.3.1. We then prove in Section 4.3.2 that all the remaining 2×5 patterns are unfoldable. This gives a complete characterization of the unfoldable $2 \times n$ maps, where $n = 5$.

4.3.1 Flat Foldable 2×5 Patterns

Let $\mathbb{C}$ be a mountain-valley pattern on a 2×5 map P. Unless otherwise stated, we always assume that the mountain-valley patterns are locally flat foldable. We now prove that if at least one of $\{u_1, l_1\}$ and $\{u_4, l_4\}$ is a pre-spine fold in $\mathbb{C}$ then P is flat foldable.

Lemma 4.5. *Let $\mathbb{C}$ be a 2×5 mountain-valley pattern. If at least one of $\{u_1, l_1\}$ and $\{u_4, l_4\}$ is a pre-spine fold, then $\mathbb{C}$ is flat foldable.*

Proof. We first assume that $\{u_1, l_1\}$ is a pre-spine fold. Since $\mathbb{C}$ is locally flat foldable, the label of s_1 must be opposite to the label of s_2. Therefore, we can apply a one-layer simple fold along the crease line $\{u_1, l_1\}$. After applying the fold, the crease s_1 matches up with s_2. Then $\mathbb{C}$ is reduced to a 2×4 mountain-valley pattern $\mathbb{C}'$. Since $\mathbb{C}'$ is flat foldable by Lemma 4.3, $\mathbb{C}$ is flat foldable. Assume now that $\{u_1, l_1\}$ is not a pre-spine fold. Then $\{u_4, l_4\}$ must be a pre-spine fold. In a similar way as described above, we can prove that $\mathbb{C}$ is flat foldable in this case. Therefore, in both cases, $\mathbb{C}$ is flat foldable. □

In the following two lemmas, we prove that $\mathbb{C}$ is flat foldable if the number of pre-spine folds in $\mathbb{C}$ is either less than two or greater than two.

Lemma 4.6. *Let $\mathbb{C}$ be a 2×5 mountain-valley pattern. If $\mathbb{C}$ has at least three pre-spine folds then $\mathbb{C}$ is flat foldable.*

Proof. If there are at least three pre-spine folds, then at least one of $\{u_1, l_1\}$ and $\{u_4, l_4\}$ must be a pre-spine fold. Then by Lemma 4.5, $\mathbb{C}$ is flat foldable. $\square$

Lemma 4.7. *Let $\mathbb{C}$ be a 2×5 mountain-valley pattern. If $\mathbb{C}$ has at most one pre-spine fold then $\mathbb{C}$ is flat foldable.*

Proof. If $\mathbb{C}$ does not have any pre-spine fold i.e., $\mathcal{H} = \emptyset$, then $\mathbb{C}$ is simply flat foldable by Lemma 4.4. Therefore, we assume that $\mathbb{C}$ has exactly one pre-spine fold. If $\{u_1, l_1\}$ or $\{u_4, l_4\}$ is a pre-spine fold, then $\mathbb{C}$ is flat foldable by Lemma 4.5. Therefore, we also assume that the pre-spine fold is either $\{u_2, l_2\}$ or $\{u_3, l_3\}$. Assume without loss of generality that it is $\{u_2, l_2\}$. We now need to consider only the two cases shown in Figure 4.6(a)–(b), where the creases shown in black straight lines do not have any labels assigned to them. All the other cases belong to the symmetric closures of these partially labeled patterns. We now prove that the partially labeled patterns are always flat foldable regardless of how we label the remaining ribs (respecting local flat foldability).

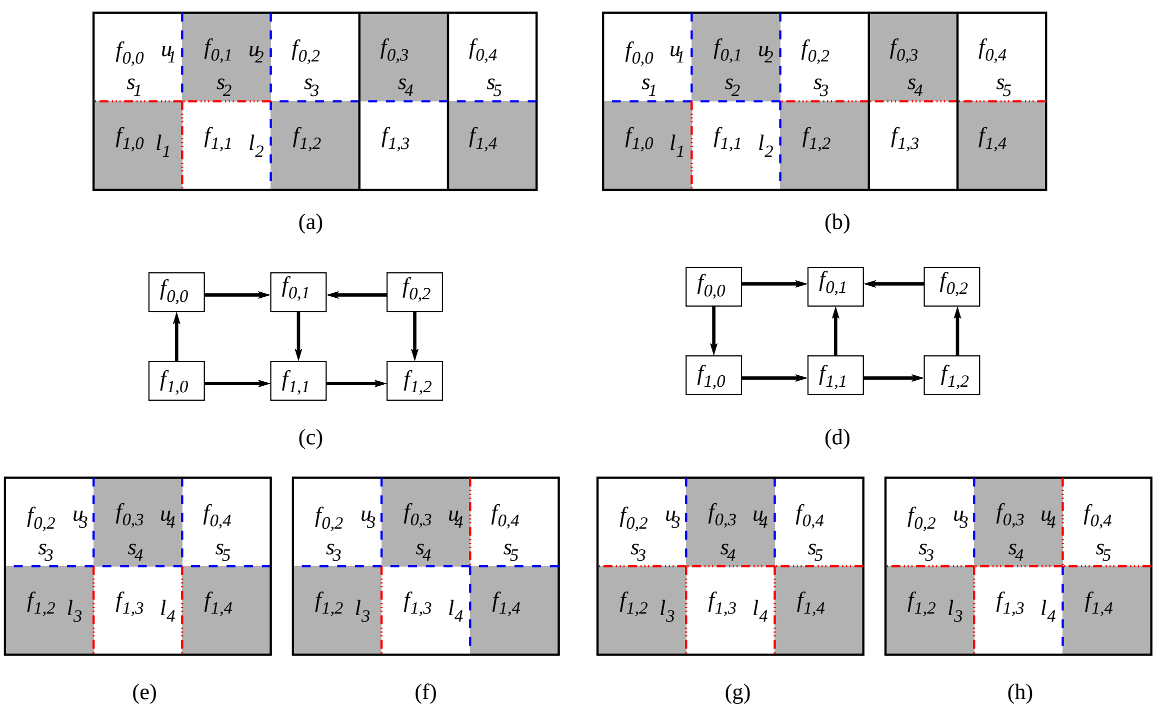


Figure 4.6: (a) A partially labeled 2×5 mountain-valley pattern $\mathbb{C}_1$, the black lines represent creases without labels, (b) another partially labeled 2×5 mountain-valley pattern $\mathbb{C}_2$, (c) directed network of $\mathbb{C}_1$, (d) directed network of $\mathbb{C}_2$.

We first assume that $\mathbb{C}$ is obtained from $\mathbb{C}_1$ (Figure 4.6(a)) by assigning labels to the unlabeled ribs. We cut the paper along the pre-spine fold $\{u_2, l_2\}$. Let the 2×2

crease pattern with the faces $f_{0,0}, f_{0,1}, f_{1,0}, f_{1,1}$ of $\mathbb{C}$ be $\mathbb{C}'$, and let the 2×3 crease pattern with the faces $f_{0,2}, \ldots, f_{0,4}, f_{1,2}, \ldots, f_{1,4}$ of $\mathbb{C}$ be $\mathbb{C}''$. From Figure 4.6(c), the only possible valid linear ordering of $\mathbb{C}'$ is $L' = f_{1,0} \prec f_{0,0} \prec f_{0,1} \prec f_{1,1}$. Observe that in L' the bottommost two faces are $f_{1,1}$ and $f_{0,1}$. Now, $\mathbb{C}''$ can be either one of the patterns shown in Figures 4.6(e) and (f) or the patterns obtained by applying MirrorX and/or MirrorY on those patterns. In any of those cases, we first apply a one layer simple fold along the spine of $\mathbb{C}''$. After applying the fold, the ribs match up and we fold the reduced 1×3 pattern by applying all-layers simple folds along u_4, l_4 (note that u_4 and l_4 lie on one another) and then along u_3, l_3. In this way we obtain a valid linear ordering L'' of $\mathbb{C}''$, where $f_{0,2}$ and $f_{1,2}$ are the topmost (or bottommost) two faces and $f_{0,2} \prec f_{1,2}$. Since the only possible linear ordering of the four faces $f_{0,1}, f_{0,2}, f_{1,1}, f_{1,2}$ is $f_{0,2} \prec f_{0,1} \prec f_{1,1} \prec f_{1,2}$ (recall from Section 3.3.1 that the directed network of a 2×2 mountain valley pattern gives a unique linear ordering of the faces), we put the stack of the faces of $\mathbb{C}'$ in the order given by L' between the two adjacent faces $f_{0,2}$ and $f_{1,2}$ in L''. We then glue the faces $f_{0,2}, f_{0,1}$ along u_2 and $f_{1,2}, f_{1,1}$ along l_2 and obtain a linear ordering L of $\mathbb{C}$ from L' and L''. For example, if $\mathbb{C}''$ is the mountain-valley pattern in Figure 4.6(e), then $L'' = f_{0,2} \prec f_{1,2} \prec f_{0,4} \prec f_{1,4} \prec f_{1,3} \prec f_{0,3}$. When we put the faces in $\mathbb{C}'$ between the faces $f_{0,2}$ and $f_{1,2}$, we get the linear ordering $L = f_{0,2} \prec f_{1,0} \prec f_{0,0} \prec f_{0,1} \prec f_{1,1} \prec f_{1,2} \prec f_{0,4} \prec f_{1,4} \prec f_{1,3} \prec f_{0,3}$.

The faces $f_{0,2}, f_{0,1}$ and $f_{1,2}, f_{1,1}$ are the wings of two west butterflies. From the linear ordering $f_{0,2} \prec f_{0,1} \prec f_{1,1} \prec f_{1,2}$ in L, we see that the pair of west butterflies l_2 and u_2 do not intersect. Since there are only two other west butterflies, u_4 and l_4, and they are stacked below (above, when $f_{0,2}, f_{1,2}$ are the bottommost two faces) the faces $f_{0,2}, f_{1,2}$ in L'', they are also stacked below $f_{0,2}, f_{1,2}$ in L. Therefore, there is no intersection of west butterflies in L. Since there is no intersection between any pair of east butterflies or south butterflies in L' or L'', there is no intersection between any pair of east or south butterflies in L. Therefore, L is a valid linear ordering of $\mathbb{C}$. We now have a final flat folded state of $\mathbb{C}$ which proves that $\mathbb{C}$ is flat foldable. Since there is no intersection between any pair of twin butterflies in L, L is a valid linear ordering of $\mathbb{C}$. Therefore, $\mathbb{C}$ is flat foldable.

We now assume that $\mathbb{C}$ is obtained from $\mathbb{C}_2$ (Figure 4.6(b)) by assigning labels to the unlabeled ribs. We cut the paper along the pre-spine fold $\{u_2, l_2\}$. Let the 2×2 crease pattern with the faces $f_{0,0}, f_{0,1}, f_{1,0}, f_{1,1}$ of $\mathbb{C}$ be $\mathbb{C}'$, and let the 2×3 crease pattern with the faces $f_{0,2}, \ldots, f_{0,4}, f_{1,2}, \ldots, f_{1,4}$ of $\mathbb{C}$ be $\mathbb{C}''$. From Figure 4.6(d), the only possible valid linear ordering of $\mathbb{C}'$ is $L' = f_{0,0} \prec f_{1,0} \prec f_{1,1} \prec f_{0,1}$. Observe that in L' the bottommost two faces are $f_{1,1}$ and $f_{0,1}$. Now, $\mathbb{C}''$ can be either one of the patterns shown in Figures 4.6(g) and (h) or the patterns obtained by applying MirrorX and/or MirrorY on those patterns. In any of those cases, we first apply a one layer simple fold along the spine of $\mathbb{C}''$. After applying the fold, the ribs match up and we fold the reduced 1×3 pattern by applying all-layers simple folds along u_4, l_4 (note that u_4 and l_4 lie on one another) and then along u_3, l_3. In this way we obtain a valid linear ordering L'' of $\mathbb{C}''$, where $f_{0,2}$ and $f_{1,2}$ are the topmost (or bottommost) two faces and $f_{1,2} \prec f_{0,2}$. Since the only possible linear ordering of the four faces $f_{0,1}, f_{0,2}, f_{1,1}, f_{1,2}$ is $f_{1,1} \prec f_{1,2} \prec f_{0,2} \prec f_{0,1}$, we put the stack of the faces of $\mathbb{C}''$ in the order given by L'' between the two adjacent faces $f_{1,1}$ and $f_{0,1}$ in L'. We then glue the faces $f_{0,2}, f_{0,1}$ along u_2 and $f_{1,2}, f_{1,1}$ along l_2 and obtain a linear ordering L of $\mathbb{C}$ from L' and L''. For example, if $\mathbb{C}''$ is the mountain-valley pattern in Figure 4.6(g), then $L'' = f_{1,2} \prec f_{0,2} \prec f_{1,4} \prec f_{0,4} \prec f_{0,3} \prec f_{1,3}$. When we put the faces in $\mathbb{C}'$ between the faces $f_{1,2}$ and $f_{0,2}$, we get the linear ordering $L = f_{1,2} \prec f_{0,0} \prec f_{1,0} \prec f_{1,1} \prec f_{0,1} f_{0,2} \prec f_{1,4} \prec f_{0,4} \prec f_{0,3} \prec f_{1,3}$.

We can prove in a similar way as in the previous case that there is no intersection between any pair of twin butterflies in L,. Therefore, L is a valid linear ordering of $\mathbb{C}$. Therefore, $\mathbb{C}$ is flat foldable. □

Let $\mathbb{C}$ be a 2×5 mountain-valley pattern. We observed in Lemmas 4.6 and 4.7 that if the number of pre-spine folds is not equal to two, then $\mathbb{C}$ is flat foldable. If the number of pre-spine folds is two and either $\{u_1, l_1\}$ or $\{u_4, l_4\}$ is one of those pre-spine folds, then $\mathbb{C}$ is flat foldable by Lemma 4.5. We now examine the case when the pre-spine folds are $\{u_2, l_2\}$ and $\{u_3, l_3\}$.

Lemma 4.8. *Let $\mathbb{C}$ be a 2×5 mountain-valley pattern, where $\{u_2, l_2\}$ and $\{u_3, l_3\}$ are the only pre-spine folds. If s_3 has the same label as $\{u_2, l_2\}$ or $\{u_3, l_3\}$, then $\mathbb{C}$ is flat foldable.*

Proof. We first assume that the label of $\{u_2, l_2\}$ is opposite to the label of $\{u_3, l_3\}$ and s_3 has the same label as $\{u_2, l_2\}$ (an example is shown in Figure 4.7(a)). Let $\mathbb{C}'$ be the 1D crease pattern obtained from $\mathbb{C}$ by removing all the creases except u_2, u_3, l_2, l_3 and then merging u_2 with l_2 and u_3 with l_3 as shown in Figure 4.7(b).

Let c_0 and c_3 denote the left and the right end of $\mathbb{C}'$. Since $|c_0 - u_2| = |u_3 - c_3| = 2$ and $|u_2 - u_3| = 1$, the pair of creases u_2, u_3 is crimpable by the definition of a crimpable pair (see Section 2.3.2). After applying the crimp operation on $\mathbb{C}'$, the spinal creases s_2, s_3 and s_3, s_4 in the original 2×5 mountain valley pattern $\mathbb{C}$ match up, since the

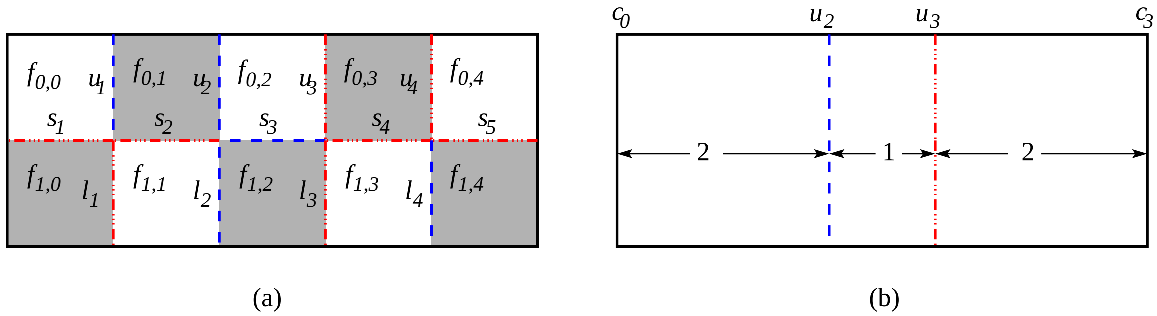


Figure 4.7: (a) A 2×5 mountain-valley pattern, where the pre-spine folds $\{u_2, l_2\}$ and $\{u_3, l_3\}$ have different labels, (b) the corresponding 1D pattern with only the creases u_2 and u_3.

labels of s_2 and s_4 are opposite to the label of s_3. $\mathbb{C}$ reduces to a 2×3 mountain-valley pattern that is simply flat foldable by Lemma 4.2. Therefore, $\mathbb{C}$ is flat foldable. In the case when the label of $\{u_2, l_2\}$ is opposite to the label of $\{u_3, l_3\}$ and s_3 has the same label as $\{u_3, l_3\}$, $\mathbb{C}$ can be proved to be flat foldable in a similar way.

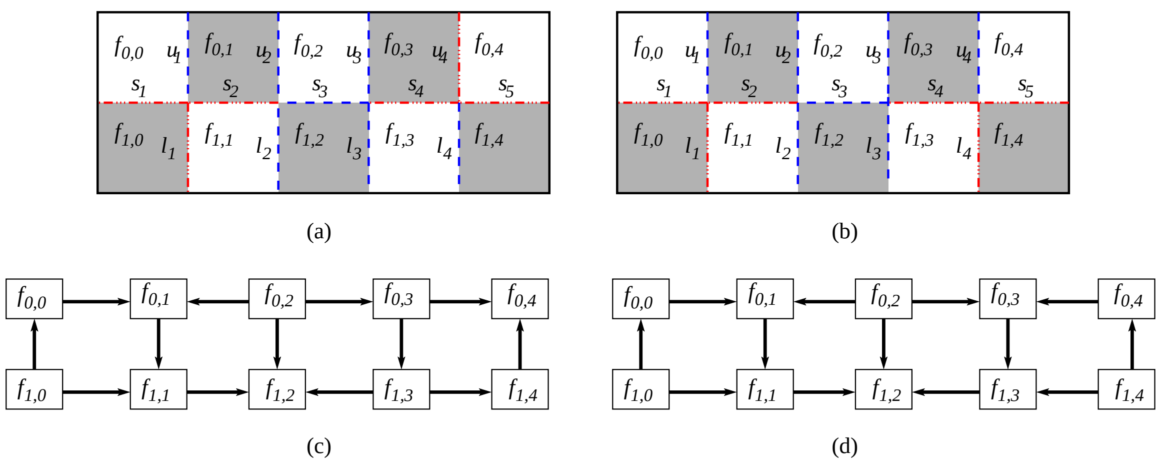


Figure 4.8: (a) A 2×5 mountain-valley pattern $\mathbb{C}_3$, (b) another 2×5 mountain-valley pattern $\mathbb{C}_4$, (c) directed network of $\mathbb{C}_3$, (d) directed network of $\mathbb{C}_4$.

We now assume that $\{u_2, l_2\}$ has the same label as $\{u_3, l_3\}$ and s_3. Then the creases s_1, s_2, s_4, s_5 have the opposite label to s_3 to preserve local flat foldability. The creases l_1 and l_4 must have labels opposite to the labels of u_1 and u_4, respectively. In this scenario, u_1 can have the same label as u_4 or the opposite label. Therefore, we consider two mountain-valley patterns $\mathbb{C}_3$ and $\mathbb{C}_4$ (See Figures 4.8(a)–(b)) such that u_1, u_4 have the same labels in $\mathbb{C}_4$ and opposite labels in $\mathbb{C}_3$. Observe that $\mathbb{C}$ belongs to the symmetric closure of either $\mathbb{C}_3$ or $\mathbb{C}_4$. If $\mathbb{C}_3$ and $\mathbb{C}_4$ are flat foldable, then by

Observation 4.1, $\mathbb{C}$ is flat foldable.

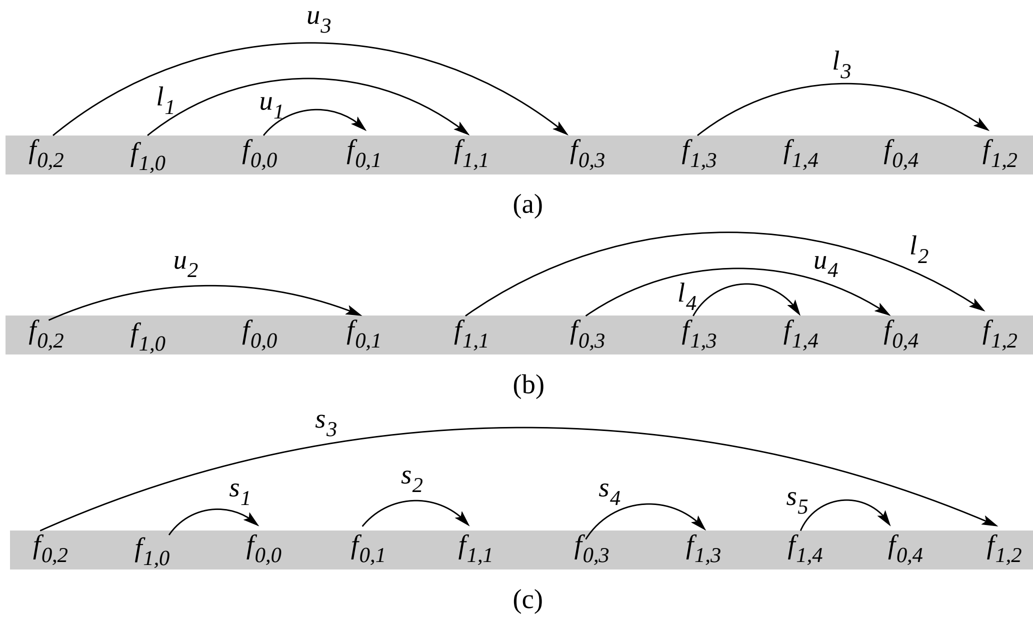


Figure 4.9: (a) The west butterflies, where the labeled arcs represent hinges, (b) the east butterflies, and (c) the south butterflies of the 2×5 mountain valley pattern in Figure 4.8(a).

We first prove that $\mathbb{C}_3$ is flat foldable. By searching exhaustively, we find a linear ordering $L = f_{0,2} \prec f_{1,0} \prec f_{0,0} \prec f_{0,1} \prec f_{1,1} \prec f_{0,3} \prec f_{1,3} \prec f_{1,4} \prec f_{0,4} \prec f_{1,2}$ that satisfies the partial ordering in the directed network of $\mathbb{C}_3$ in Figure 4.8(c). The east butterflies are u_1, u_3, l_1, l_3 (the ribs with odd indices), the west butterflies are u_2, u_4, l_2, l_4 (the ribs with even indices) and the south butterflies are $s_1, s_2, , s_3, s_4, s_5$ (the spinal creases) as shown in Figures 4.9(a), (b) and (c), respectively. Figure 4.9 shows that there are no intersection between any pair of twin butterflies in L, and hence, L is a valid linear ordering. Therefore there exists a final flat folded state of $\mathbb{C}_3$ and hence, $\mathbb{C}_3$ is flat foldable.

Similarly, for $\mathbb{C}_4$, we find the linear ordering $L = f_{0,2} \prec f_{1,0} \prec f_{0,0} \prec f_{0,1} \prec f_{1,1} \prec f_{1,4} \prec f_{0,4} \prec f_{0,3} \prec f_{1,3} \prec f_{1,2}$ by searching exhaustively. By checking all pairs of twin butterflies as we did for $\mathbb{C}_3$, we see that L is a valid linear ordering of $\mathbb{C}_4$. Therefore, $\mathbb{C}_4$ is also flat foldable. □

The only remaining case for 2×5 locally flat foldable mountain-valley patterns is the case when $\{u_2, l_2\}$ and $\{u_3, l_3\}$ are the only pre-spine folds, $\{u_2, l_2\}$ and $\{u_3, l_3\}$ have the same label and s_3 has the opposite label to $\{u_2, l_2\}$ and $\{u_3, l_3\}$. As will be shown in the next section, when this occurs, the 2×5 pattern is unfoldable.

4.3.2 Unfoldable 2×5 Patterns

We now characterize the 2×5 mountain-valley patterns that are unfoldable.

Theorem 4.1. *Let $\mathbb{C}$ be a 2×5 mountain-valley pattern such that $\{u_2, l_2\}$ and $\{u_3, l_3\}$ are the only pre-spine folds in $\mathbb{C}$. If $\{u_2, l_2\}$ and $\{u_3, l_3\}$ have the same label and s_3 has the opposite label to $\{u_2, l_2\}$ and $\{u_3, l_3\}$, then $\mathbb{C}$ is unfoldable.*

Proof. Since $\{u_2, l_2\}$ and $\{u_3, l_3\}$ are the only pre-spine folds, the creases s_1, s_2, s_4, s_5 must have the opposite label to s_3 to preserve local flat foldability. The creases l_1 and l_4 must have labels opposite to the labels of u_1 and u_4, respectively. In this scenario, u_1 can have the same label as u_4 or the opposite label. Therefore, we consider two mountain-valley patterns $\mathbb{C}_1$ and $\mathbb{C}_2$ (see Figures 4.10(a)–(b)) such that u_1, u_4 have the same labels in $\mathbb{C}_1$ and opposite labels in $\mathbb{C}_2$. Observe that $\mathbb{C}$ belongs to the symmetric closure of either $\mathbb{C}_1$ or $\mathbb{C}_2$. If $\mathbb{C}_1$ and $\mathbb{C}_2$ are unfoldable, then by Observation 4.1, $\mathbb{C}$ is unfoldable.

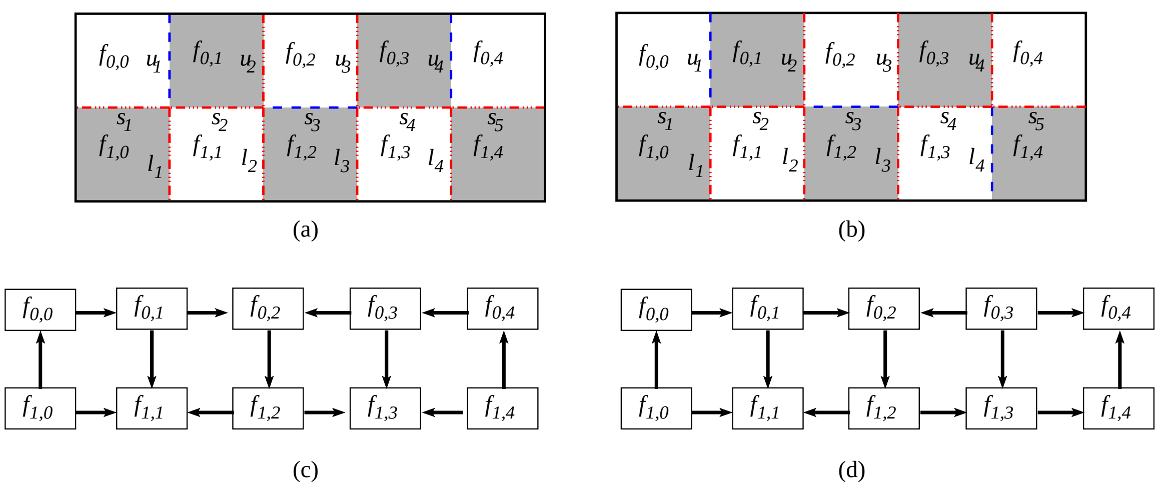


Figure 4.10: (a) An unfoldable 2×5 mountain-valley pattern $\mathbb{C}_1$, (b) another unfoldable 2×5 mountain-valley pattern $\mathbb{C}_2$, (c) directed network of $\mathbb{C}_1$, and (d) directed network of $\mathbb{C}_2$.

We first prove that $\mathbb{C}_1$ is unfoldable. Suppose for a contradiction that there is a valid linear ordering L of $\mathbb{C}_1$. From Figure 4.10(c), we observe that $f_{1,4} \prec f_{1,2} \prec f_{1,3}$ since there is a directed path from $f_{1,4}$ to $f_{1,3}$ through $f_{1,2}$. Again, $f_{1,2} \prec f_{1,3}$ and also, $f_{1,2} \prec f_{1,1}$. Suppose that in L, $f_{1,3} \prec f_{1,1}$. Then the linear ordering of the four faces $f_{1,1}, f_{1,2}, f_{1,3}, f_{1,4}$ becomes $f_{1,4} \prec f_{1,2} \prec f_{1,3} \prec f_{1,1}$. When checking the pair of west butterflies l_2 and l_4, we see that there is an intersection of the wings as shown in Figure 4.11(a). Therefore, $f_{1,1} \prec f_{1,3}$ in L. But $f_{1,0} \prec f_{1,2}$. Since $f_{1,2} \prec f_{1,1} \prec f_{1,3}$, the linear ordering of the four faces $f_{1,0}, f_{1,1}, f_{1,2}, f_{1,3}$ becomes $f_{1,0} \prec f_{1,2} \prec f_{1,1} \prec f_{1,3}$. This causes intersection of wings in the pair of east butterflies l_1 and l_3 as shown in Figure 4.11(b). Therefore, there is no linear ordering of the faces of $\mathbb{C}_1$ that satisfies the Butterfly Condition (Theorem 3.1) and $\mathbb{C}_1$ is unfoldable.

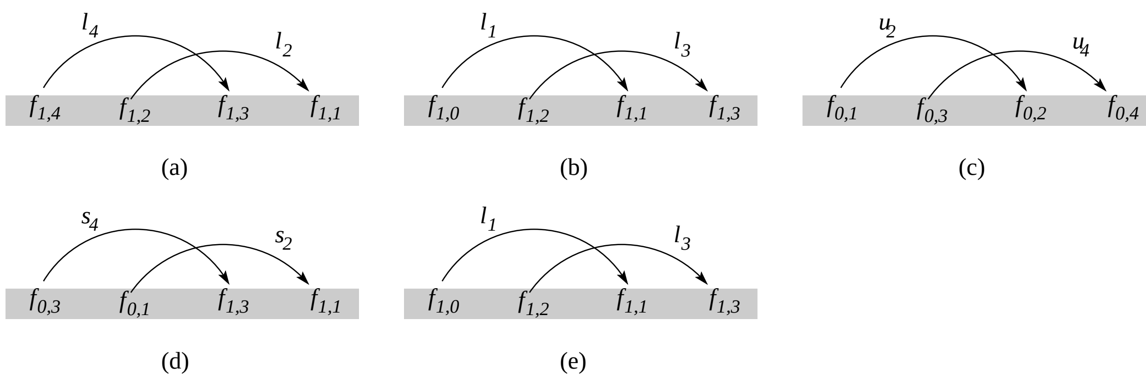


Figure 4.11: (a)–(b) Intersections in the linear ordering of $\mathbb{C}_1$, (c)–(e) intersections in the linear ordering of $\mathbb{C}_2$.

We now prove that $\mathbb{C}_2$ is unfoldable. Suppose for a contradiction that there is a valid linear ordering L' of $\mathbb{C}_2$. From Figure 4.10(d), we observe that $f_{0,3} \prec f_{0,2} \prec f_{0,4}$ and $f_{0,1} \prec f_{0,2}$. Suppose that in L', $f_{0,1} \prec f_{0,3}$. Then the linear ordering of the four faces $f_{0,1}, f_{0,2}, f_{0,3}, f_{0,4}$ becomes $f_{0,1} \prec f_{0,3} \prec f_{0,2} \prec f_{0,4}$. When checking the pair of west butterflies u_2 and u_4, we see that there is an intersection of the wings as shown in Figure 4.11(c). Therefore, $f_{0,3} \prec f_{0,1}$ in L'. From Figure 4.10(d), $f_{0,1} \prec f_{1,1}$ and $f_{0,1} \prec f_{1,3}$. If we assume that $f_{1,3} \prec f_{1,1}$, then the linear ordering of the four faces $f_{0,1}, f_{1,1}, f_{0,3}, f_{1,3}$ becomes $f_{0,3} \prec f_{0,1} \prec f_{1,3} \prec f_{1,1}$. This causes intersection of wings in the pair of south butterflies s_4 and s_2 as shown in Figure 4.11(d). Therefore, $f_{1,1} \prec f_{1,3}$ in L'. But now $f_{1,0} \prec f_{1,2} \prec f_{1,1} \prec f_{1,3}$, which causes intersection of wings in the pair of east butterflies l_1 and l_3 as shown in Figure 4.11(e). Therefore, there is no linear ordering of the faces of $\mathbb{C}_2$ that satisfies the Butterfly Condition and $\mathbb{C}_2$ is unfoldable. □

Let χ_5 be the set of all the 2×5 mountain-valley patterns that belong to the symmetric closures of $\mathbb{C}_1$ (Figure 4.10(a)) and $\mathbb{C}_2$ (Figure 4.10(b)). Note that $\mathbb{C}_1$ and

$\mathbb{C}_2$ are unfoldable by Theorem 4.1. Therefore, by Observation 4.1, all the mountain-valley patterns in χ_5 are unfoldable.

We now calculate the number of patterns in χ_5. Since $\mathbb{C}_1$ has the same label for $l_1, l_2, l_3, l_4, u_2, u_3$ and the same label for u_1, u_4, the operation MirrorX will produce the same pattern. The operation Rotation is not applicable since $m \neq n$. Therefore, we can only apply two operations MirrorY and Switch on $\mathbb{C}_1$. Therefore, there are $2^2 = 4$ unique mountain-valley patterns in the symmetric closure of $\mathbb{C}_1$ as shown in Figure 4.12.

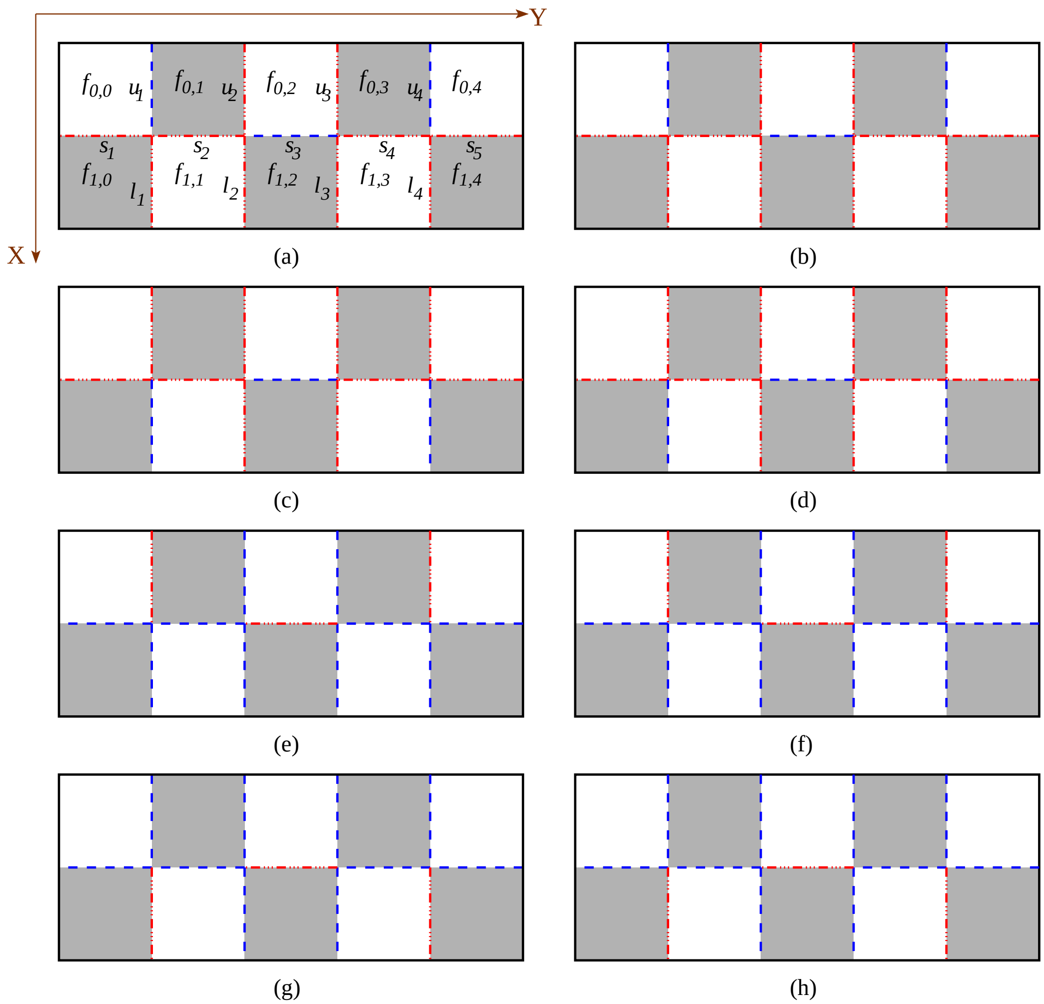


Figure 4.12: (a) $\mathbb{C}_1$, and (b) MirrorX = $\mathbb{C}_1$, (c) MirrorY, (d) (MirrorY + MirrorX) = MirrorY, (e) Switch, (f) (Switch + MirrorX) = Switch, (g) MirrorY + Switch, (h) (MirrorY + Switch + MirrorX) = (MirrorY + Switch) applied on $\mathbb{C}_1$.

Similarly, there are $2^2 = 4$ unique mountain-valley patterns in the symmetric closure of $\mathbb{C}_2$ as shown in Figure 4.13. The symmetric closures of $\mathbb{C}_1$ and $\mathbb{C}_2$ are disjoint sets. Therefore, $|\chi_5| = 4 + 4 = 8$.

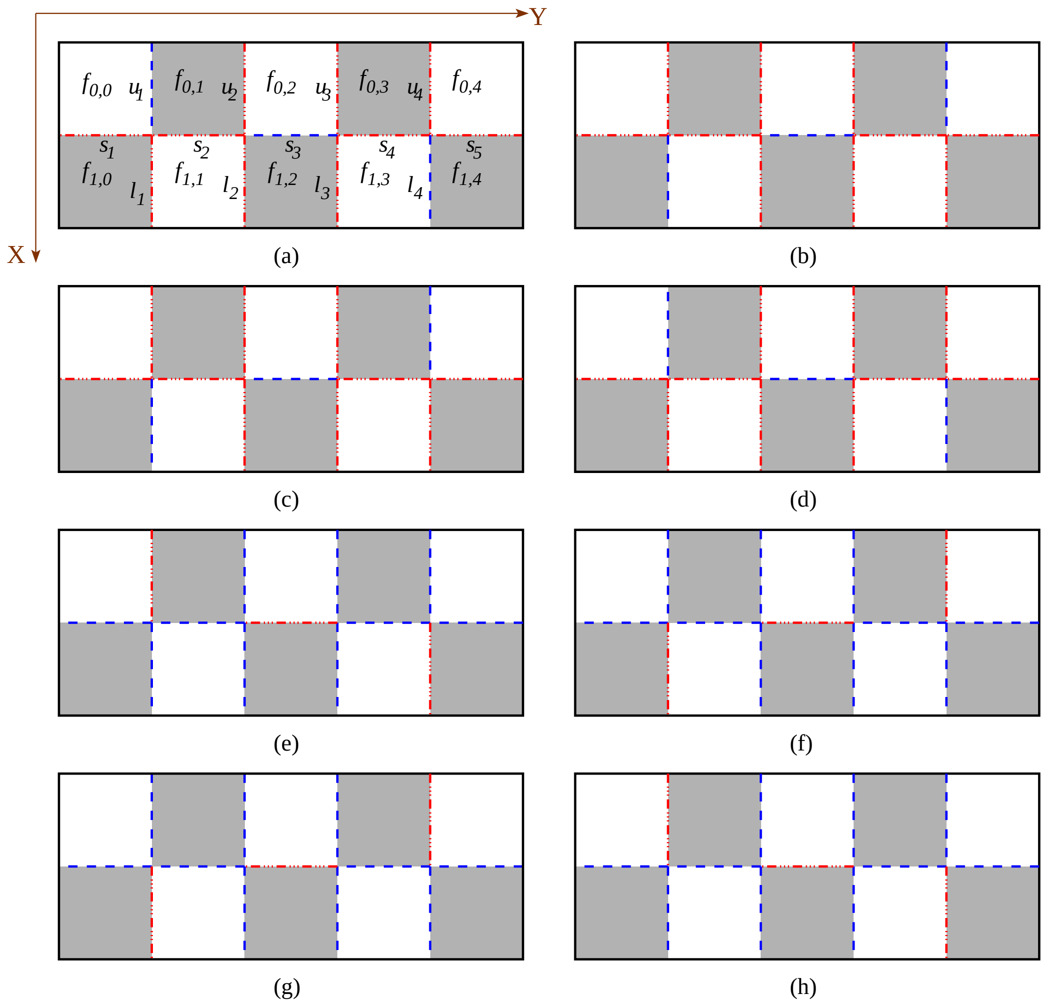


Figure 4.13: (a) $\mathbb{C}_2$, and (b) MirrorX, (c) MirrorY, (d) MirrorY + MirrorX, (e) Switch, (f) Switch + MirrorX, (g) MirrorY + Switch, (h) MirrorY + Switch + MirrorX applied on $\mathbb{C}_2$. Note that the mountain-valley patterns in (b), (d), (f), (h) are the same as the patterns in (c), (a), (g), (e), respectively.

We now show that the patterns in χ_5 are the only 2×5 patterns that are unfoldable.

Theorem 4.2. *Any* 2×5 *locally flat foldable mountain-valley pattern* $\mathbb{C} \notin \chi_5$ *is flat foldable.*

Proof. We count the total number of locally flat foldable 2×5 mountain-valley patterns. There are 4 upper ribs $u_1, \dots, u_4$ and 5 spinal creases $s_1, \dots, s_5$. We can assign either mountain or valley to each of those 9 creases. For each l_i, $1 \le i \le 4$, if s_i, s_{i+1}, u_i, have the same label, then to ensure local flat foldability, we must assign the opposite label to l_i. Otherwise, we must assign the label to l_i that is assigned to two of s_i, s_{i+1}, u_i. Therefore, we have two choices for 9 creases and for the rest of the four creases we have only one choice restricted by local flat foldability rules. Hence, there are $2^9 = 512$ locally flat foldable 2×5 mountain-valley patterns. Since the number of unfoldable patterns is at least 8, there are at most $512 - 8 = 504$ flat foldable patterns.

In the rest of the proof, we count the number of 2×5 mountain-valley patterns that are flat foldable according to Lemmas 4.6–4.8. As we will prove, there are in fact 504 such patterns. This will account for all flat foldable patterns and will thus complete the proof.

We now count the number of flat foldable 2×5 mountain-valley patterns. We consider the following cases based on the number of pre-spine folds. In each of these cases, we have two choices (mountain or valley) for the labels of the 4 upper ribs and any one spinal crease unless otherwise stated. The labels of the rest of the spinal creases and the lower ribs are determined by local flat foldability.

We first calculate the number of 2×5 patterns that have at least three pre-spine folds. These patterns are foldable by Lemma 4.6.

(a) **Three pre-spine folds:** We can choose 3 pre-spine folds from the four upper ribs in $\binom{4}{3} = 4$ ways. Therefore, the number of patterns in this case is $4(2^5) = 128$.

(b) **Four pre-spine folds:** We can choose 4 pre-spine folds from the four upper ribs in $\binom{4}{4} = 1$ way. Therefore, the number of patterns in this case is $2^5 = 32$.

We next calculate the number of 2×5 patterns that have at most one pre-spine fold. These patterns are foldable by Lemma 4.7.

(a) **One pre-spine fold:** We can choose 1 pre-spine fold from the four upper ribs in $\binom{4}{1} = 4$ ways. Therefore, the number of patterns in this case is $4(2^5) = 128$.

(b) **No pre-spine fold:** The number of patterns in this case is $2^5 = 32$.

We now calculate the number of 2×5 patterns that have exactly two pre-spine folds.

(a) **One of the pre-spine folds is $\{u_1, l_1\}$:** We can choose the other pre-spine fold from u_2, u_3, u_4 in $\binom{3}{1} = 3$ ways. Therefore, the number of patterns in this case is $3(2^5) = 96$. These patterns are foldable by Lemma 4.5.

(b) **One of the pre-spine folds is $\{u_4, l_4\}$ and $\{u_1, l_1\}$ is not a pre-spine fold:** We can choose the other pre-spine fold from u_2, u_3 in $\binom{2}{1} = 2$ ways. Therefore, the number of patterns in this case is $2(2^5) = 64$. These patterns are foldable by Lemma 4.5.

(c) $\{u_2, l_2\}$ **and** $\{u_3, l_3\}$ **are the only pre-spine folds:** We consider three cases.

   i. $\{u_2, l_2\}$ *and* $\{u_3, l_3\}$ *receive different labels:* Then the assignment of the label of u_3 is dependent on the label of u_2. Therefore, only three upper ribs and any one spinal crease have two options for labels. The number of patterns in this case is $2^4 = 16$. These patterns are foldable by Lemma 4.8.

   ii. u_2, l_2, u_3, l_3 *and* s_3 *receive the same label:* Then the assignment of the labels of u_3 and s_3 is dependent on the label of u_2. Therefore, only three upper ribs have two options for labels. The number of patterns in this case is $2^3 = 8$. These patterns are foldable by Lemma 4.8.

   iii. u_2, l_2, u_3, l_3 *receive the same label and* s_3 *receives the opposite label:* Only three upper ribs have two options for labels. The number of patterns in this case is $2^3 = 8$. These are the patterns in χ_5, which are unfoldable by Theorem 4.1.

Summing up all the patterns, we have exactly 504 flat foldable patterns. Together with the 8 unfoldable patterns, this accounts for all 512 locally flat foldable 2×5 mountain-valley patterns. This completes the proof. □

4.4 Generalized Unfoldable Patterns

In this section, we define a class χ_n of unfoldable $2 \times n$ mountain-valley patterns, $n \geq 5$. We first show a subclass S_n of unfoldable $2 \times n$ mountain-valley patterns, for every $n > 5$, that do not contain any pattern in χ_5 as a fragment. We then show that any map with an unfoldable pattern (i.e., a pattern in S_n) as a fragment is unfoldable. Using this result, we define the class χ_n, which includes S_n as a subclass.

We now define the subclass S_n of unfoldable $2 \times n$, $n > 5$, mountain-valley patterns. Let $\mathbb{C}$ be a pattern for S_n that satisfies the following (a)–(d).

(a) $\mathbb{C}$ is locally flat foldable.

(b) There are exactly two pre-spine folds $\{u_2, l_2\}$ and $\{u_{n-2}, l_{n-2}\}$.

(c) All the upper ribs receive the same label.

(d) s_3 receives the opposite label of the upper ribs.

Condition (b) ensures that $\mathbb{C}$ does not contain any pattern in χ_5 as a fragment.

The upper ribs of $\mathbb{C}$ can be labeled either mountain or valley. Without loss of generality we assume that the upper ribs of $\mathbb{C}$ receive the label mountain as shown in Figure 4.14(a). Consequently, all the lower ribs of $\mathbb{C}$ except l_2 and l_{n-2} must be labeled valley. Since u_2 and u_{n-2} are the pre-spine folds, l_2 and l_{n-2} receive the same label mountain as u_2 and u_{n-2}. The spinal creases $s_4, \ldots s_{n-2}$ must all be labeled the same as s_3, which according to requirement (s) must be labeled valley. To preserve local flat foldability, the other spinal creases s_1, s_2, s_{n-1}, s_n must be labeled mountain (opposite to the label of s_3).

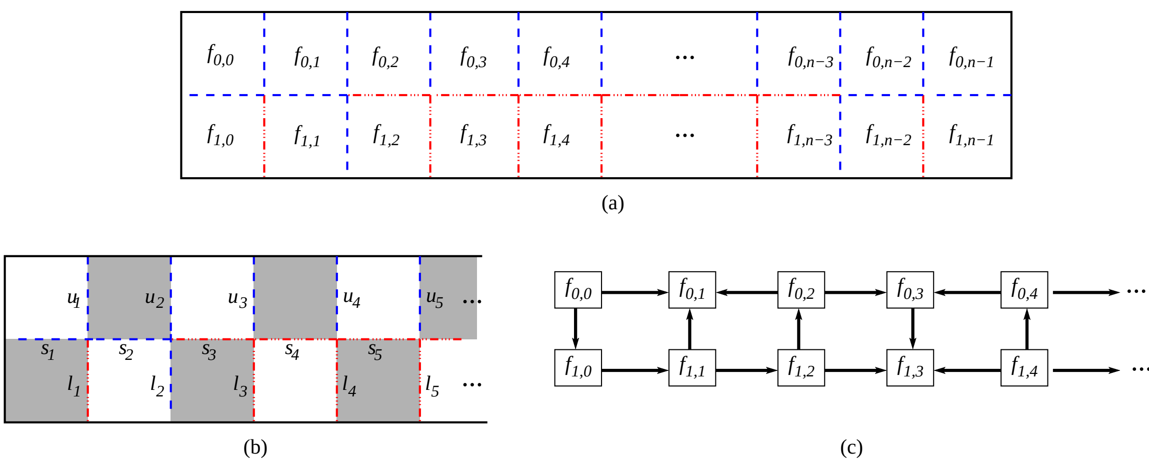


Figure 4.14: (a) An unfoldable $2 \times n$ mountain-valley pattern $\mathbb{C}$. (b) Checkerboard pattern of the first five columns $c_0, \ldots, c_4$, and (c) the corresponding directed network.

Let L_i be a candidate for a valid linear ordering of the faces in the columns $c_0, \ldots, c_i$ of $\mathbb{C}$, where $0 \leq i \leq n-1$. From Figure 4.14(c), $L_2 = f_{0,0} \prec f_{1,0} \prec f_{1,1} \prec f_{1,2} \prec f_{0,2} \prec f_{0,1}$ is the unique candidate for a valid linear ordering of the faces in $c_0, \ldots, c_2$. Since $f_{0,0} \prec f_{0,2} \prec f_{0,1}$ and $f_{0,2} \prec f_{0,3}$, we can consider whether $f_{0,1} \prec f_{0,3}$ or $f_{0,3} \prec f_{0,1}$. If $f_{0,1} \prec f_{0,3}$, then the linear ordering $f_{0,0} \prec f_{0,2} \prec f_{0,1} \prec f_{0,3}$ causes intersection between the pair of east butterflies u_1 and u_3. Therefore, $f_{0,3} \prec f_{0,1}$, and hence $f_{0,3}$ must be placed between $f_{0,2}$ and $f_{0,1}$. Similarly, $f_{1,3}$ must be placed between $f_{0,2}$ and $f_{0,1}$ to avoid intersection between the pair of east butterflies u_1 and l_3. Therefore, the linear ordering $L_3 = f_{0,0} \prec f_{1,0} \prec f_{1,1} \prec f_{1,2} \prec f_{0,2} \prec f_{0,3} \prec f_{1,3} \prec f_{0,1}$

is the unique candidate for a valid linear ordering for the case $i = 3$. We now show that for each $4 \leq i \leq n-3$, there is a unique candidate for a valid linear ordering.

Lemma 4.9. *Let L_i be a candidate for a valid linear ordering of the faces in columns $c_0, \ldots, c_i$ of $\mathbb{C} \in S$, where $4 \leq i \leq n-3$. Then the following conditions hold.*

(a) L_i is the only candidate,

(b) the order of the faces $f_{0,i}, f_{1,i}, f_{0,i-2}$ and $f_{0,i-1}$ in L_i is $f_{0,i-2} \prec f_{1,i} \prec f_{0,i} \prec f_{0,i-1}$, when i is even, and $f_{0,i-1} \prec f_{0,i} \prec f_{1,i} \prec f_{0,i-2}$, when i is odd, and

(c) the faces $f_{0,i}, f_{1,i}, f_{0,i-2}$ and $f_{0,i-1}$ are consecutive as a set (i.e., they appear together as a set, with no other faces lying between the extremal faces in this set).

Proof. We prove the claim by induction on i. The base case is $i = 4$. We remember that $L_3 = f_{0,0} \prec f_{1,0} \prec f_{1,1} \prec f_{1,2} \prec f_{0,2} \prec f_{0,3} \prec f_{1,3} \prec f_{0,1}$ was the unique candidate for a valid linear ordering of the first four columns of $\mathbb{C}$. Since $f_{1,4} \prec f_{0,4} \prec f_{0,3}$, we can decide either $f_{1,4} \prec f_{0,2}$ or $f_{0,2} \prec f_{1,4}$. If $f_{1,4} \prec f_{0,2}$, then the linear ordering $f_{1,4} \prec f_{0,2} \prec f_{1,3} \prec f_{0,1}$ causes intersection between the pair of west butterflies u_2 and l_4 (the indices of the ribs are even for the west butterflies). Since L_3 is unique and the faces $f_{0,4}$ and $f_{1,4}$ have unique valid positions in L_3, $L_4 = f_{0,0} \prec f_{1,0} \prec f_{1,1} \prec f_{1,2} \prec f_{0,2} \prec f_{1,4} \prec f_{0,4} \prec f_{0,3} \prec f_{1,3} \prec f_{0,1}$ is the unique candidate for a valid linear ordering. This satisfies Condition (a). The order of the faces $f_{0,4}, f_{1,4}, f_{0,2}$ and $f_{0,3}$ is $f_{0,2} \prec f_{1,4} \prec f_{0,4} \prec f_{0,3}$ in L_4 and they are consecutive as a set, which satisfies Conditions (b) and (c), respectively.

We now assume that Conditions (a)–(c) hold for each $4 \leq k < i$, where $4 \leq i \leq n-3$. Therefore,

(a) L_k is the only candidate for a valid linear ordering of the first $k+1$ columns of $\mathbb{C}$,

(b) the order of the faces $f_{0,k}, f_{1,k}, f_{0,k-2}$ and $f_{0,k-1}$ in L_k is $f_{0,k-2} \prec f_{1,k} \prec f_{0,k} \prec f_{0,k-1}$, when k is even, and $f_{0,k-1} \prec f_{0,k} \prec f_{1,k} \prec f_{0,k-2}$, when k is odd, and

(c) the faces $f_{0,k}, f_{1,k}, f_{0,k-2}$ and $f_{0,k-1}$ are consecutive as a set.

We now show that the conditions hold for $k+1$. We have to consider two cases.

Case 1: k is odd. In this case, $f_{0,k-1} \prec f_{0,k} \prec f_{1,k} \prec f_{0,k-2}$. From Figure 4.15(b), we see that since $k+1$ is even, we have $f_{1,k+1} \prec f_{0,k+1} \prec f_{0,k}$. We can decide

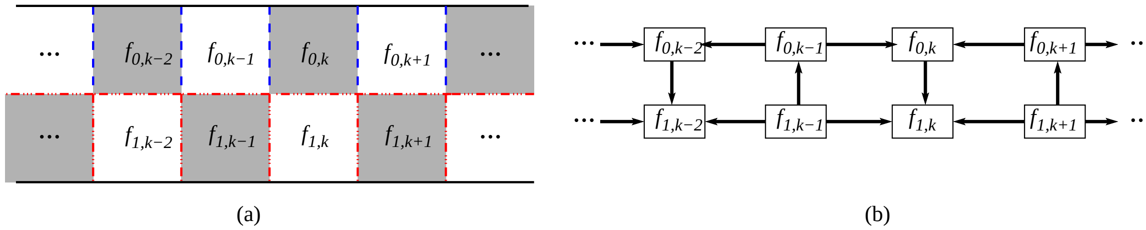


Figure 4.15: (a) Checkerboard pattern of the columns $c_{k-2}, \ldots, c_{k+1}$, when k is odd (so $f_{0,k}$ is dark). (b) the corresponding directed network.

whether $f_{1,k+1} \prec f_{0,k-1}$ or $f_{0,k-1} \prec f_{1,k+1}$. If $f_{1,k+1} \prec f_{0,k-1}$, then the linear ordering $f_{1,k+1} \prec f_{0,k-1} \prec f_{1,k} \prec f_{0,k-2}$ causes intersection between two west butterflies u_{k-1} and l_{k+1}. Therefore, the linear ordering of the four faces $f_{0,k+1}, f_{1,k+1}, f_{0,k-1}$ and $f_{0,k}$ is $f_{0,k-1} \prec f_{1,k+1} \prec f_{0,k+1} \prec f_{0,k}$ (since k is odd, $k+1$ is even), which satisfies Condition (b). Since L_k is unique and the faces $f_{0,k+1}, f_{1,k+1}$ must be placed between two consecutive faces $f_{0,k-1}$ and $f_{0,k}$ in L_k, L_{k+1} is also unique, satisfying Condition (a), and the faces $f_{0,k+1}, f_{1,k+1}, f_{0,k-1}$ and $f_{0,k}$ are consecutive as a set in L_{k+1}, satisfying Condition (c).

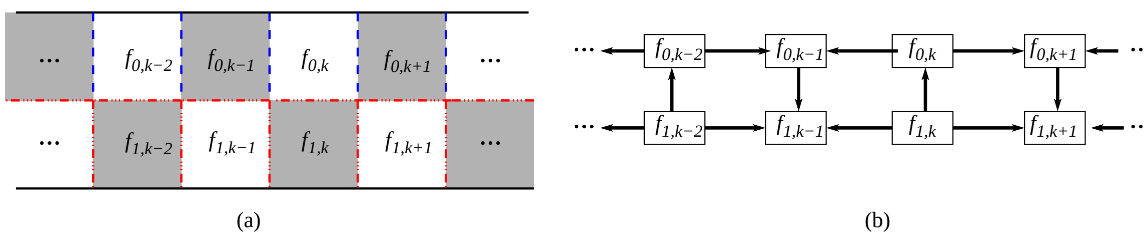


Figure 4.16: (a) Checkerboard pattern of the columns $c_{k-2}, \ldots, c_{k+1}$, when k is even. (b) the corresponding directed network.

Case 2: k is even. In this case, $f_{0,k-2} \prec f_{1,k} \prec f_{0,k} \prec f_{0,k-1}$. From Figure 4.16(b), we see that since $k+1$ is odd, we have $f_{0,k} \prec f_{0,k+1} \prec f_{1,k+1}$. We can decide whether $f_{0,k-1} \prec f_{1,k+1}$ or $f_{1,k+1} \prec f_{0,k-1}$. If $f_{0,k-1} \prec f_{1,k+1}$, then the linear ordering $f_{0,k-2} \prec f_{1,k} \prec f_{0,k-1} \prec f_{1,k+1}$ causes intersection between two east butterflies u_{k-1} and l_{k+1}. Therefore, the linear ordering of the four faces $f_{0,k+1}, f_{1,k+1}, f_{0,k-1}$ and $f_{0,k}$ is $f_{0,k} \prec f_{0,k+1} \prec f_{1,k+1} \prec f_{0,k-1}$ (since k is even, $k+1$ is odd), which satisfies Condition (b). Since L_k is unique and the faces $f_{0,k+1}, f_{1,k+1}$ must be placed between two consecutive faces $f_{0,k}$ and $f_{0,k-1}$ in L_k, L_{k+1} is also unique, satisfying Condition (a), and the faces $f_{0,k+1}, f_{1,k+1}, f_{0,k-1}$ and $f_{0,k}$ are consecutive as a set in L_{k+1}, satisfying Condition (c).

Therefore, Conditions (a)–(c) hold for $k+1$ in both cases which concludes the proof. □

We now show that $\mathbb{C}$ is unfoldable.

Lemma 4.10. *Let $\mathbb{C}$ be a $2 \times n$ mountain-valley pattern in S. Then $\mathbb{C}$ is unfoldable.*

Proof. We first construct the unique candidate L_{n-3} for a valid linear ordering as described in the proof of Lemma 4.9. We now construct L_{n-2} from L_{n-3}. We have to consider two cases.

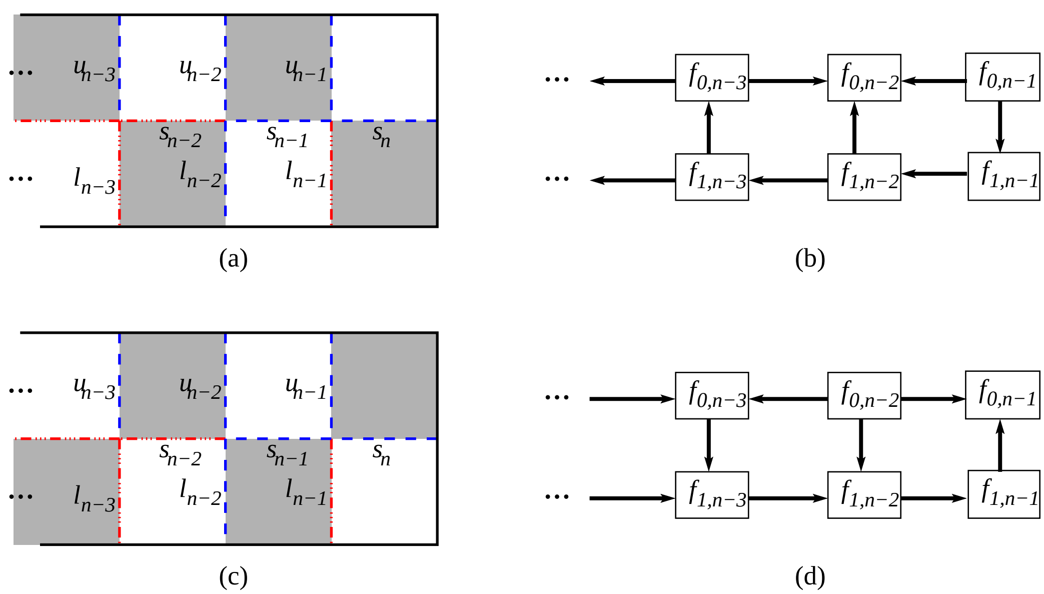


Figure 4.17: (a) The checkerboard pattern of the last three columns $c_{n-3}, \ldots, c_{n-1}$ of $\mathbb{C}$, when n is odd and (b) the directed network of $\mathbb{C}$, when n is odd. (c) The checkerboard pattern of the last three columns of $\mathbb{C}$, when n is even and (d) the directed network of $\mathbb{C}$, when n is even.

Case 1: n is odd. From Figure 4.17(b), we see that $f_{0,n-3} \prec f_{0,n-2}$. By Lemma 4.9, $f_{0,n-5} \prec f_{1,n-3} \prec f_{0,n-3} \prec f_{0,n-4}$ in L_{n-3} (since $n-3$ is even) and these four faces are consecutive. We now decide whether $f_{0,n-4} \prec f_{0,n-2}$ or $f_{0,n-2} \prec f_{0,n-4}$. If $f_{0,n-4} \prec f_{0,n-2}$, then the linear ordering $f_{0,n-5} \prec f_{0,n-3} \prec f_{0,n-4} \prec f_{0,n-2}$ causes intersection between the east butterflies u_{n-4} and u_{n-2} (since n is odd, $n-4$ and $n-2$ are odd and hence the butterflies u_{n-4}, u_{n-2} are east butterflies). Therefore, $f_{0,n-2} \prec f_{0,n-4}$ and the linear ordering of the faces $f_{0,n-3}, f_{0,n-2}, f_{0,n-4}$ in L_{n-2} is $f_{0,n-3} \prec f_{0,n-2} \prec f_{0,n-4}$. Since there is a directed path from $f_{0,n-1}$ to $f_{0,n-3}$ in the directed network, $f_{0,n-1} \prec f_{0,n-3}$. We then have to place $f_{0,n-1}$ somewhere above

$f_{0,n-3}$. But any such placement will have the linear ordering $f_{0,n-1} \prec f_{0,n-3} \prec f_{0,n-2} \prec f_{0,n-4}$, and thus cause intersection between the west butterflies u_{n-3} and u_{n-1} (the indices of the ribs are even since n is odd and hence the butterflies are west butterflies). Therefore, there is no valid linear ordering of $\mathbb{C}$, i.e., $\mathbb{C}$ is unfoldable.

Case 2: n is even. From Figure 4.17(d), we see that $f_{0,n-2} \prec f_{0,n-3}$. By Lemma 4.9, $f_{0,n-4} \prec f_{0,n-3} \prec f_{1,n-3} \prec f_{0,n-5}$ in L_{n-3} (since $n-3$ is odd) and these four faces are consecutive. We now decide whether $f_{0,n-2} \prec f_{0,n-4}$ or $f_{0,n-4} \prec f_{0,n-2}$. If $f_{0,n-2} \prec f_{0,n-4}$, then the linear ordering $f_{0,n-2} \prec f_{0,n-4} \prec f_{0,n-3} \prec f_{0,n-5}$ causes intersection between the west butterflies u_{n-4} and u_{n-2} (since n is even, $n-4$ and $n-2$ are even and hence the butterflies u_{n-4}, u_{n-2} are west butterflies). Therefore, $f_{0,n-4} \prec f_{0,n-2}$ and the linear ordering of the faces $f_{0,n-3}, f_{0,n-2}, f_{0,n-4}$ in L_{n-2} is $f_{0,n-4} \prec f_{0,n-2} \prec f_{0,n-3}$. Since there is a directed path from $f_{0,n-3}$ to $f_{0,n-1}$ in the directed network, $f_{0,n-3} \prec f_{0,n-1}$. We then have to place $f_{0,n-1}$ somewhere below $f_{0,n-3}$. But any such placement will have the linear ordering $f_{0,n-4} \prec f_{0,n-2} \prec f_{0,n-3} \prec f_{0,n-1}$, and thus cause intersection between the east butterflies u_{n-3} and u_{n-1} (the indices of the ribs are odd since n is even and hence the butterflies are east butterflies). Therefore, there is no valid linear ordering of $\mathbb{C}$, i.e., $\mathbb{C}$ is unfoldable.

In each of the two cases, $\mathbb{C}$ is unfoldable. This completes the proof. □

Since any mountain-valley pattern in S_n, $n > 5$, is in the symmetric closure of $\mathbb{C}$, by Lemma 4.10 Observation 4.1, all the patterns in S_n are unfoldable. We now show that any $m \times n$ mountain-valley pattern that has an unfoldable pattern as a fragment is also unfoldable.

Lemma 4.11. *Let $\mathbb{C}$ be an $m \times n$ mountain-valley pattern. Let $\mathbb{C}'$ be a fragment of $\mathbb{C}$. If $\mathbb{C}'$ is not flat foldable, then $\mathbb{C}$ is not flat foldable.*

Proof. We prove the claim by proving the contrapositive, i.e., if $\mathbb{C}$ is flat foldable, then $\mathbb{C}'$ is flat foldable. Let l be a valid linear ordering of the faces of $\mathbb{C}$ in a final flat folded state. We remove all the faces of $\mathbb{C}$ from l that are not in $\mathbb{C}'$. Let the resulting linear ordering be l'. Since l satisfies the Butterfly Condition, there cannot be any intersection of wings between any pair of butterflies in l'. Therefore, l' is a valid linear ordering of $\mathbb{C}'$ and $\mathbb{C}'$ is flat foldable. □

We are now ready to characterize the class χ_n of unfoldable $2 \times n$ mountain-valley patterns, where $n \geq 5$. Let $\mathbb{C}$ be a pattern in χ_n. Then by definition of χ_n, we require $\mathbb{C}$ to satisfy the following conditions (a)–(d).

(a) $\mathbb{C}$ is locally flat foldable.

(b) There are exactly two pre-spine folds u_i and u_j, where $2 \leq i < j \leq n-2$.

(c) All the upper ribs receive the same label.

(d) s_{i+1} receives the opposite label of the upper ribs.

We now show that $\mathbb{C}$ is unfoldable.

Theorem 4.3. *Let $\mathbb{C}$ be a $2 \times n$ mountain-valley pattern in χ_n, where $n \geq 5$. Then $\mathbb{C}$ is unfoldable. Furthermore, membership in χ_n can be tested in linear time.*

Proof. Let $i = 2$ and $j = n-2$. Then $\mathbb{C} \in S_n$, and hence the pattern is unfoldable by Lemma 4.10. Therefore, we assume that $\mathbb{C} \notin S_n$. Let $\mathbb{C}'$ be the fragment of $\mathbb{C}$ with the faces in the columns $c_{i-2}, \ldots, c_{j+1}$. Then $\mathbb{C}' \in S_x$, where $x = j-i+4$. Therefore, $\mathbb{C}'$ is unfoldable by Lemma 4.10. Since a fragment of $\mathbb{C}$ is unfoldable, $\mathbb{C}$ is unfoldable by Lemma 4.11.

We can check in $O(n)$ time whether all the upper ribs receive the same label (i.e., Condition (c) is satisfied) by scanning from left to right. We can check in $O(n)$ time whether $\mathbb{C}$ is locally flat foldable (i.e., Condition (a) is satisfied) by checking the creases incident to each of the $n-1$ vertices of $\mathbb{C}$. If Conditions (a) and (c) are satisfied, we check in $O(n)$ time whether there are exactly two pre-spine folds (Condition (b)) and get the index i for the leftmost pre-spine fold $\{u_i, l_i\}$. If Conditions (a)–(c) are satisfied, then it takes $O(1)$ time to check whether the label of s_{i+1} is opposite to the label of the upper ribs (Condition (d)). Therefore, it takes $O(n)+O(n)+O(n)+O(1) = O(n)$ time to check whether $\mathbb{C}$ is a member of χ_n. □

4.5 Conclusion

In this chapter, for the case $n = 5$, we determined all the unfoldable 2×5 mountain-valley patterns. We found 8 such patterns, which agrees with the results of Justin [19]. In face, we also determined all the flat foldable 2×5 patterns, of which there are 504. We gave a combinatorial proof of the unfoldability of the 2×5 patterns in this chapter using the concepts we introduced in this chapter and the previous chapter.Moreover, we proved that a set of unfoldable patterns exists for every $n \geq 5$. However, we do not have a characterization of all the unfoldable patterns for $n \geq 5$, which we leave as an open problem.

Chapter 5

Conclusions

In this thesis, we investigated the problem of folding an $m \times n$ grid paper with a given mountain-valley crease pattern on it. We introduced the concepts of *butterflies*, *checkerboard patterns*, *directed networks* and formalized the concept of *symmetric closures*. We have demonstrated how powerful these tools are by achieving the following results using those concepts.

1. In Section 3.2.3, we gave a linear time algorithm (**Algorithm 3**) to recognize whether a given linear ordering of an $m \times n$ unassigned crease pattern is a valid linear ordering. We used the concepts of butterflies and checkerboard patterns in the algorithm.

2. In Section 3.3.2, we gave a linear time algorithm (**Algorithm 4**) to recognize whether a given linear ordering of an $m \times n$ mountain-valley pattern is a valid linear ordering. We used the concepts of butterflies, checkerboard patterns and directed networks in the algorithm.

3. In Section 3.4, we showed that it is possible to enumerate all possible valid linear orderings of a given $m \times n$ pattern (assigned or unassigned) using checkerboard patterns and directed networks (only for the case of mountain-valley patterns).

4. In Section 4.2 and 4.3, we showed that all these tools can be used to enumerate flat foldable $2 \times n$ crease patterns and also, to recognize unfoldable crease patterns, for $n \geq 2$. Although we restricted our discussion to $2 \times n$ patterns, these concepts can easily be generalized for $m \times n$ crease patterns, where $m \geq 2$.

5. In Section 4.4, we defined a class χ_n of unfoldable $2\times n$ mountain-valley patterns for every $n \geq 5$.

Although we have recognized some unfoldable $2 \times n$ mountain-valley patterns in Chapter 4, we do not have a complete characterization of all the unfoldable $2 \times n$ mountain-valley patterns. For example, we have the following example of an unfoldable 2×6 pattern in Figure 5.1 which is not a member of χ_6.

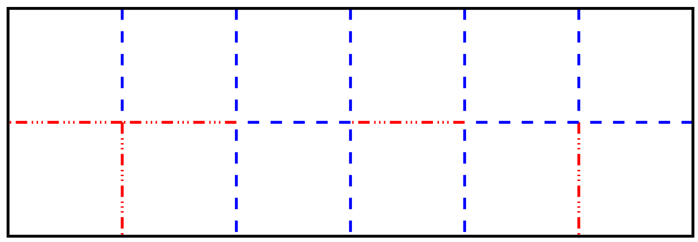


Figure 5.1: An unfoldable 2×6 mountain-valley pattern.

Based on this example and the set of unfoldable 2×5 patterns, we give the following conjecture.

Conjecture *Let $\mathbb{C}$ be a $2\times n$, $n \geq 6$, mountain-valley pattern, where $\{u_2, l_2\}, \{u_3, l_3\}, \ldots, \{u_{n-2}, l_{n-2}\}$ are the only pre-spine folds and all the upper ribs have the same label. Then $\mathbb{C}$ is unfoldable.*

In Lemma 4.7, we applied a technique of cutting the paper into disjoint fragments, getting a final flat folded state for each of the fragments and then combining them together to get a final flat folded state of the input crease pattern. Although we applied that technique only on 2×5 patterns with exactly one pre-spine fold, it is in fact applicable to any $2 \times n$ crease pattern with exactly one pre-spine fold for $n \geq 4$. We therefore, have the following open problem.

Open Problem 1: *Is there a subclass of $2 \times n$ mountain-valley patterns that are not flat foldable by simple folding but a final flat folded state of those patterns can be reached by physical operations (possibly with cutting and gluing the faces back together)?*

Of course, the famous open problem posed by Edmonds [6] (see Section 1.2) still remains open.

length of paper n	Sum of flat foldable crease patterns	1 crease	2 creases	3 creases	4 creases
2	2	2			
3	8	4	4		
4	26	6	12	8	
5	76	8	20	32	16

Table 5.1: Count of flat foldable 1D mountain-valley patterns.

Since we investigated mainly combinatorial properties of maps, we also enumerated the number of flat foldable patterns of small sizes. For example, Table 5 shows the total number of flat foldable 1D mountain-valley patterns, where columns $3, 4, 5, 6$ show how many of the patterns have 1 crease, 2 creases, 3 creases and 4 creases, respectively.

From this observation, we have the following two open problems.

Open Problem 2: *Is there a backtracking algorithm to enumerate all flat foldable mountain-valley assignments for a given orthogonal crease pattern (1D or 2D)?*

Open Problem 3: *Is there a closed form expression for the number of flat foldable 1D mountain-valley patterns on a paper of length n?*

Bibliography

[1] Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Martin L. Demaine, Joseph S. B. Mitchell, Saurabh Sethia, and Steven S. Skiena. When can you fold a map? *Computational Geometry : Theory and Applications*, 29:23–46, September 2004. ISSN 0925-7721.

[2] Marshall Bern and Barry Hayes. The complexity of flat origami. In *Proceedings of the 7th annual ACM-SIAM symposium on Discrete algorithms (SODA 1996)*, SODA 1996, pages 175–183. Society for Industrial and Applied Mathematics, 1996.

[3] C. Davis and D. E. Knuth. Number representations and dragon curves - i. *Journal of Recreational Mathematics*, 3(2):66–81, April 1970.

[4] C. Davis and D. E. Knuth. Number representations and dragon curves - ii. *Journal of Recreational Mathematics*, 3(3):133–149, July 1970.

[5] Michel Dekking. Paperfolding morphisms, planefilling curves, and fractal tiles. *Theoretical Computer Science*, 414(1):20–37, 2012.

[6] Erik D. Demaine and Joseph O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra.* Cambridge University Press, New York, NY, USA, 2007.

[7] P Di Francesco. Folding and coloring problems in mathematics and physics. pliages et coloriages en mathmatiques et en physique. *Bulletin of American Mathematical Society*, 37:251–307, 2000.

[8] Lill E. Résolution graphique des equations numériques d'un degré quelconque á une inconnue. *Nouv. Annales Math.*, 6:359–362, 1867.

[9] Vacca G. Della piegatura della carta applicata alla geometria. *Periodico diMathematiche*, 10:43–50, 1930.

[10] Martin Gardner. **Mathematical Games** column in *Scientific American*, 1967.

[11] Martin Gardner. *Wheels life and other mathematical amusements*. W. H. Freeman and Company, New York, 1983.

[12] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[13] Thomas Hull. On the mathematics of flat origamis. *Congressus Numerantium*, 100:215–224, 1994.

[14] Thomas Hull. A note on "impossible" paper folding. *American Mathematical Monthly*, 103(3):240241, March 1996.

[15] Thomas Hull. The combinatorics of flat folds: A survey. In *Origami3: 3rd International Meeting of Origami Science, Math, and Education*, page 2938, 2001.

[16] Thomas Hull. Counting mountain-valley assignments for flat folds. *Ars Combinatorica*, 67, 2003.

[17] Thomas Hull. *Project Origami: Activities for Exploring Mathematics*. A K Peters. Taylor & Francis, 2006.

[18] H. Huzita. Axiomatic development of origami geometry. In H. Huzita, editor, *In Proceedings of the 1st International Meeting of Origami Science and Technology*, page 143158, 1989.

[19] Jacques Justin. Aspects mathematiques du pliage de papier (mathematical aspects of paper fold. In H. Huzita, editor, *1st International Meeting of Origami Science and Scientific Origami*, page 263277, 1989.

[20] Jeff Kahn and Jeong Han Kim. Entropy and sorting. In *Proceedings of the 24th annual ACM symposium on Theory of computing (STOC 1992)*, pages 178–187. ACM, 1992.

[21] Kunihiko Kasahara and Toshie Takahama. *Origami for the Connoisseur*. Japan Publications Inc., 1987.

[22] Toshikazu Kawasaki. On the relation between mountain-creases and valley-creases of a flat. In H. Huzita, editor, *1st International Meeting of Origami Science and Scientific Origami*, pages 229–Icra237, 1989.

[23] F. Klein. *Famous Problems of Elementary Geometry*. Ginn, Boston, 1897.

[24] Jon M. Kleinberg and Éva Tardos. *Algorithm design*. Addison-Wesley, 2006.

[25] H Kobayashi, B Kresling, and J F V Vincent. The geometry of unfolding tree leaves. *Proceedings of the Royal Society B: Biological Sciences*, 265(1391):147–154, 1998.

[26] J.E. Koehler. Folding a strip of stamps. *Journal of Combinatorial Theory*, 5: 135–152, 1968.

[27] Robert J. Lang. A computational algorithm for origami design. In *Proceedings of the 12th annual symposium on Computational geometry*, SCG '96, pages 98–105, New York, NY, USA, 1996. ACM.

[28] Liang Lu and Srinivas Akella. Folding cartons with fixtures: A motion planning approach. In *IEEE International Conference on Robotics and Automation (ICRA 1999)*, pages 1570–1576, 1999.

[29] W. F. Lunnon. A map-folding problem. *Mathematics of Computation*, 22:193–199, 1968.

[30] W. F. Lunnon. Multi-dimensional map-folding. *The Computer Journal*, 14: 75–80, 1971.

[31] Beloch M. P. Sul metodo del ripiegamento della carta per la risoluzione dei problemi geometrici. *Periodico diMathematiche*, 16:104–108, 1936.

[32] Gara Pruesse and Frank Ruskey. Generating linear extensions fast. *SIAM Journal on Computing*, 23(2):373–386, 1994.

[33] Samuel Randlett. *The Best of Origami*. E. P. Dutton, 1963.

[34] T. Sundara Rao. *Geometric Exercises in Paper Folding*. Addison, 1893.

[35] J. Touchard. Contribution l'tude du probleme des timbres poste (contribution to the study of the problem of postage stamps). *Canadian Journal of Mathematics*, 2:385–398, 1950.

[36] Ryuhei Uehara. On stretch minimization problem on unit strip paper. In *The 22nd Canadian Conference on Computational Geometry (CCCG 2010)*, pages 223–226, 2010.

[37] Cheng-Hua Wang. *Manufacturability-Driven Decomposition of Sheet Metal Products*. PhD thesis, The Robotics Institute, Carnegie Mellon University, September, 1997.