Lace Tessellations:
A mathematical model for bobbin lace
and an exhaustive combinatorial search for patterns

by

Veronika Irvine
B.Sc., University of Western Ontario, 1990
M.Math, University of Waterloo, 1992

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

DOCTOR OF PHILOSOPHY

in the Department of Computer Science

University of Victoria

Lace Tessellations:
A mathematical model for bobbin lace
and an exhaustive combinatorial search for patterns

by

Veronika Irvine
B.Sc., University of Western Ontario, 1990
M.Math, University of Waterloo, 1992

Supervisory Committee

Dr. F. Ruskey, Supervisor
(Department of Computer Science)

Dr. W. Myrvold, Departmental Member
(Department of Computer Science)

Dr. G. MacGillivray, Outside Member
(Department of Mathematics and Statistics)

Supervisory Committee

Dr. F. Ruskey, Supervisor
(Department of Computer Science)

Dr. W. Myrvold, Departmental Member
(Department of Computer Science)

Dr. G. MacGillivray, Outside Member
(Department of Mathematics and Statistics)

ABSTRACT

Bobbin lace is a 500-year-old art form in which threads are braided together in an alternating manner to produce a lace fabric. A key component in its construction is a small pattern, called a bobbin lace ground, that can be repeated periodically to fill a region of any size. In this thesis we present a mathematical model for bobbin lace grounds representing the structure as the pair $(\Delta_1(G), \zeta(v))$ where $\Delta_1(G)$ is a topological embedding of a 2-regular digraph, G, on a torus and $\zeta(v)$ is a mapping from the vertices of G to a set of braid words. We explore in depth the properties that $\Delta_1(G)$ must possess in order to produce workable lace patterns. Having developed a solid, logical foundation for bobbin lace grounds, we enumerate and exhaustively generate patterns that conform to that model. We start by specifying an equivalence relation and define what makes a pattern prime so that we can identify unique representatives. We then prove that there are an infinite number of prime workable patterns. One of the key properties identified in the model is that it must be possible to partition $\Delta_1(G)$ into a set of osculating circuits such that each circuit has a wrapping index of $(1, 0)$; that is, the circuit wraps once around the meridian of the torus and does not wrap around the longitude. We use this property to exhaustively generate workable patterns for increasing numbers of vertices in G by gluing together lattice paths in an osculating manner. Using a backtracking algorithm to

process the lattice paths, we identify over 5 million distinct prime patterns. This is well in excess of the roughly 1,000 found in lace ground catalogues. The lattice paths used in our approach are members of a family of partially directed lattice paths that have not been previously reported. We explore these paths in detail, develop a recurrence relation and generating function for their enumeration and present a bijection between these paths and a subset of Motzkin paths. Finally, to draw out of the extremely large number of patterns some of the more aesthetically interesting cases for lacemakers to work on, we look for examples that have a high degree of symmetry. We demonstrate, by computational generation, that there are lace ground representatives from each of the 17 planar periodic symmetry groups.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

There are many people who have supported and encouraged me along this journey and I would like to extend my heart felt thanks to all of them.

My supervisor Frank Ruskey has patiently helped me express my ideas in a more formal and mathematically rigorous way. I thank him for taking interest in bobbin lace to the point of becoming a lacemaker. I would also like to thank Sue Whitesides; it was in her course on Computational Geometry that the seed for this thesis was planted. For the extensive feedback received from Wendy Myrvold and Gary MacGillivray, I am greatly appreciative. It was an honour to have Erik Demaine as my external examiner; his contributions to computational origami inspired me to apply computer theory to bobbin lace.

For future work suggestions, I would like to thank Jason Cantarella for the idea of using `ridgerunner` to model lace threads under tension and Robert Lang for the idea of using Ammann bars to explore aperiodic lace patterns.

I owe a great deal to the lacemakers who have enthusiastically supported me. Thank you to Lenka Suchanek for turning my work into art, to Jo Pol for asking just the right questions and to the members of the Victoria Lace Guild, Arachne and IOLI for their feedback and encouragement. I greatly appreciate all the lacemakers who have tried a TesseLace pattern.

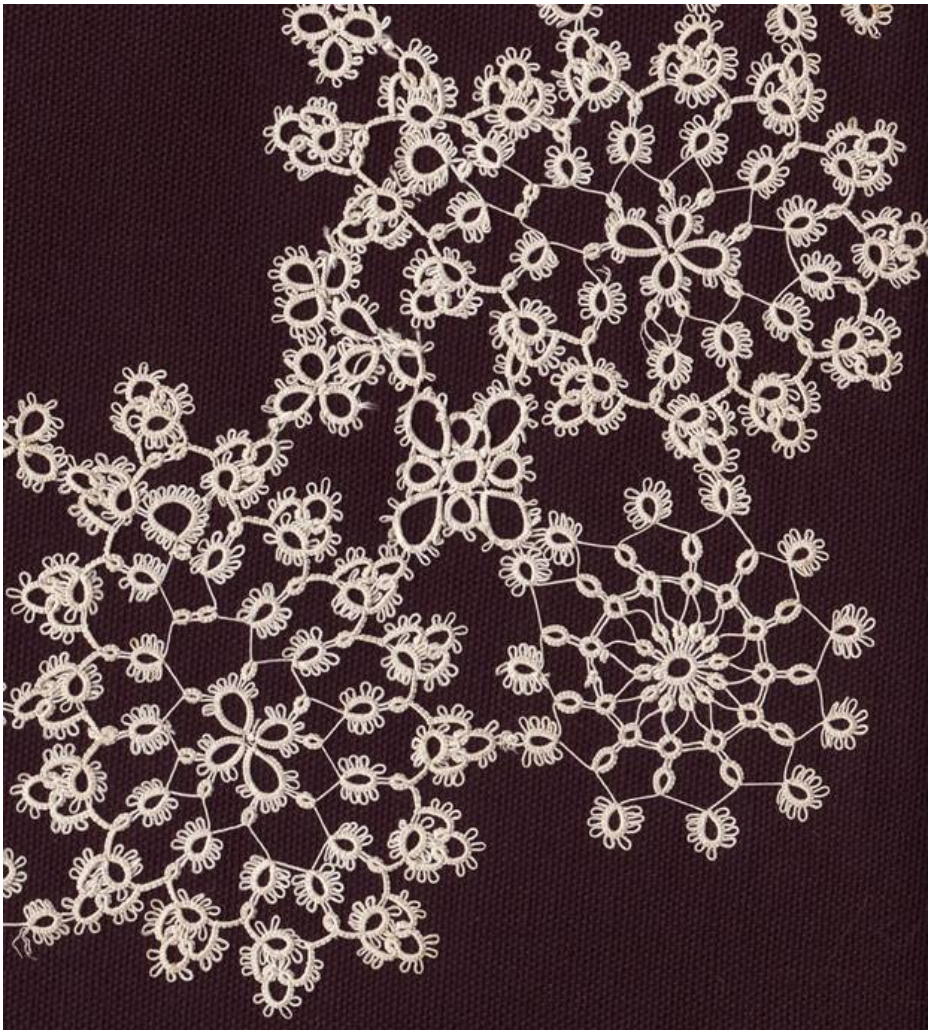
I could not have made it this far without family and friends. In particular, I wish to thank Andrew for always having confidence in me, my mom for supporting my artistic endeavours and proof reading much of my writing, and my dad for introducing me at a very young age to the idea that math, computers and art can be combined.

Finally, I would like to thank the National Science and Engineering Research Council for helping fund my research with an Alexander Graham Bell scholarship. I am also thankful for support from the University of Victoria Graduate scholarship, the E.F. and A.J. Wood scholarship and the Google Anita Borg Canada scholarship.

“ Mathematics is the art of giving the same name to different things.”
Henri Poincaré

DEDICATION

To Pauline Leijen-Mohr whose lace launched me on this journey.



Tatted lace by Pauline Leijen-Mohr (1906-1943)

Chapter 1

Introduction

The lacemakers behind the beautiful art works of centuries past would have spent 10 to 12 hours per day plying their craft. Starting from the age of 5, they worked for very little pay, hunched over pillows, often in damp cabins with poor lighting, the dampness being necessary to prevent the fine linen threads from breaking. True masters of the art form, their well developed sense of intuition would have guided them to discover new designs but such explorations probably also included many frustrating hours of trial and error, 'unproductive' hours they could ill afford.

We wish to continue their exploration, to see if there are more patterns to discover, either for beauty or for practical application, but under better conditions! Fortunately, we have the benefit of computers which are extremely good at performing repetitive tasks with tireless precision. They just need to be given explicit instructions.

In this thesis we will demonstrate that bobbin lace tessellations can be represented by a mathematical model largely based on graph theory. From this model we will prove that there is an infinite number of patterns. The model shall also act as the logical foundation for a computer algorithm to exhaustively enumerate and generate patterns of workable bobbin lace, resulting in the discovery of millions of new patterns.

1.1 Motivation

The digital age has brought with it an explosion in mechanized capabilities. Drones and 3D printing are recent additions to the growing arsenal of technologies replacing handcrafted objects. Digital technology provides wonderful new capabilities but we should not lose sight of the centuries of development invested in traditional crafts. The skills of handcrafting, now practiced by a small and aging group of artisans, are at serious risk of extinction. Like endangered plants and animals, not only is the loss of a handcraft a reduction in the beauty and diversity of our world, but it is also the loss of solutions to medical, social and environmental problems. We cite a recent example [35] in which the thousand year old tradition of fine weaving practiced by the women of Aymara was used to create a transcatheter closer (heart plug) for patients suffering from patent ductus arteriosus, an infant onset heart defect that reduces the effectiveness of blood circulation. Within a few generations, the weaving skill of the Aymaran women may quite conceivably be lost and along with it similar innovations.

Among the handcrafts teetering on the edge of extinction is the 500-year-old tradition of making bobbin lace. Bobbin lace is formed by braiding together anywhere from a dozen to several hundreds of threads to form intricate patterns. For examples, we refer the reader to detailed photographs which can be found online at several lace archives [49, 83]. In the course of its history, lace has played a major role in the fashions and economies of Europe [52]. At times it was valued more than gold and employed hundreds of thousands of workers, predominantly women and children. The economic impact was so great that trade embargoes were put in place and laws drafted to prevent the wearing of lace from foreign countries. As with any opportunity for great profit, there was extensive technological investment in manufacturing. In the 19th century, stiff competition from machine-made copies drove down the price and led to a simplification of designs. After World War I, fashions and the expectations of women in the work force changed. Handmade bobbin lace production as an industry ceased completely and was relegated to the status of hobby craft while machine-made lace was primarily used for curtains. Interest in reviving the craft started in the 1970's and initially focused on the simple designs produced in the 19th century [34, 51]. Over the past 20 years, interest has started to turn toward the more advanced techniques employed in early laces as well as the invention of new techniques. Active discussion and development continues at lace guilds [38, 68, 82] and online discussion groups [37, 88].

Today, knitting and weaving are the focus of industrial textiles; despite its past prominence, lace is a largely untapped source for advanced textiles. We feel this is an oversight resulting from a restricted vision of the potential of lace. Instead of thinking of lace as merely an ornament used in days gone by, let us consider lace in terms of its attributes. The defining feature of bobbin lace is its holes of various sizes held securely in place by an alternating braid structure. This open composition could be valuable in biomedical applications such as manifolds for tissue growth or as tendon or muscle replacements in prosthetic devices. Its airiness could be applied to create lightweight building materials or structures that allow the flow of water or air. A feature that distinguishes lace from woven cloth or knitted textiles is that lace threads follow complex paths with two threads taking significantly different paths in the same fabric. Such paths could be used for conducting messages or forces in unique ways giving us access to fabrics with unusual properties. Of special consideration is the potential for auxetic capabilities (fabrics that, when stretched, also thicken in a perpendicular direction).

1.2 Agenda

Lacemaking has a long history of innovation and adaptation to changing demands. In this thesis we will carry on that tradition by applying modern areas of mathematics and computer science to understand its construction.

For readers not familiar with bobbin lace, in Chapter 2 we give an introduction to the techniques used in its production. We also present previous work on the application of mathematics to textile design. In Chapter 3 we set forth a mathematical model that captures the fundamental principles of bobbin lace design by leveraging concepts from graph theory. To the best of our knowledge, this is the first attempt to formally describe bobbin lace in mathematical terms. In preparation for exhaustively enumerating and generating patterns based on our model, in Chapter 4 we will establish criteria for determining when two patterns are the same. Using the definition of equality from the previous chapter, in Chapter 5 we shall describe a combinatorial search algorithm that looks for bobbin lace patterns in the space of graph embeddings produced by joining lattice paths together in an osculating manner. In addition to its role in bobbin lace, the family of lattice paths introduced in Chapter 5 has intrinsic interest and is explored in more detail in Chapter 6. The combinatorial search presented in Chapter 5 yields a large number of solutions, far

greater in number than could be worked be an army of lacemakers. In the interest of identifying a subset of patterns with the greatest aesthetic appeal, in Chapter 7 we refine our search by considering symmetry in the lace designs. Finally, in Chapter 8 we compare our algorithmically produced results to catalogues of traditional patterns and suggest areas for further exploration.

We are at a fortunate crossroads in history. There still exist artisans actively working on their craft at a time when advancements in mathematics and algorithms make it possible to capture fundamental aspects of their work. Having been granted this period of overlap, it seems prudent to take advantage of it. Our hope is that the ideas presented in this thesis can serve as a launch pad for future work.

Chapter 2

Background

In this chapter we will give a brief introduction to the techniques used in the craft of bobbin lace. We will look at how other researchers have applied mathematics to textiles in general and discuss the current state of exploration for bobbin lace including some of the major aspects that still need to be addressed.

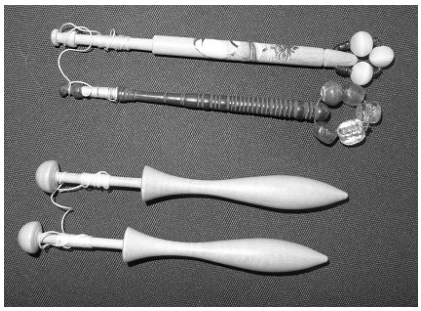
2.1 Bobbin Lace Technique

Perhaps the easiest way to understand bobbin lace is to look at the six step process by which it is made.

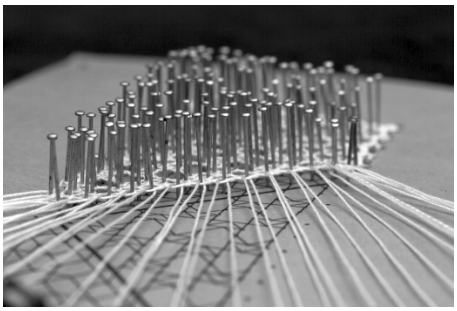
Step 1) Prepare the threads. To manage many long threads without creating a tangled mess, each end of a thread is wound evenly onto one of a pair of bobbins (see Figure 2.1(a)). A bobbin, commonly made from wood, is about 10cm in length. One end is flanged to hold a length of thread and the other end, usually thicker and sometimes weighted with beads, is the handle.

Step 2) Prepare the pattern. The lace is worked on top of a firm pillow stuffed with material such as straw, sawdust, wool, or, in some modern pillows, ethafoam or polystyrene. The pillow can be disk shaped (cookie pillow, see Figure 2.1(c)) or sausage shaped (bolster pillow, see Figure 2.1(d)). As threads are braided together, they are held in place by pins pushed into the pillow (see Figure 2.1(b)). To start a piece of lace, a pattern, such as the

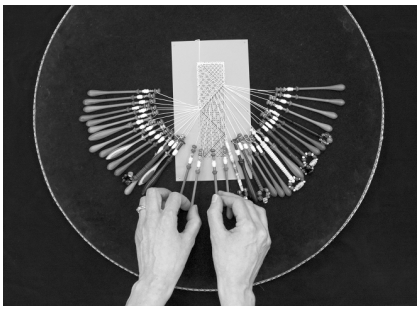
one shown in Figure 2.4(a), is copied onto stiff material - originally this may have been vellum or parchment but modern lace makers use coloured card stock or printing paper overlaid with blue contact paper. The black dots in the pattern represent the position of pins. Before starting to make the lace, all of these dots are pricked through to make small holes. The pattern is then pinned to the pillow.



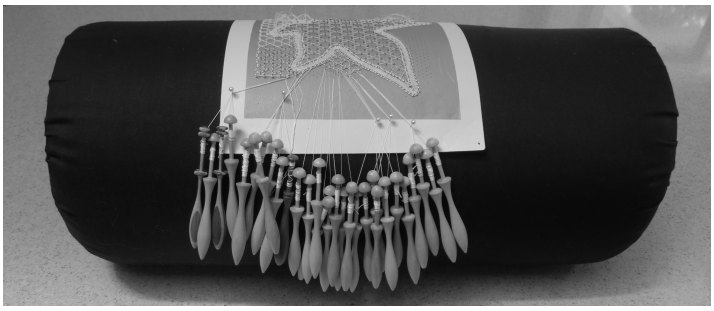
(a) Two different bobbin styles



(b) Closeup of pins



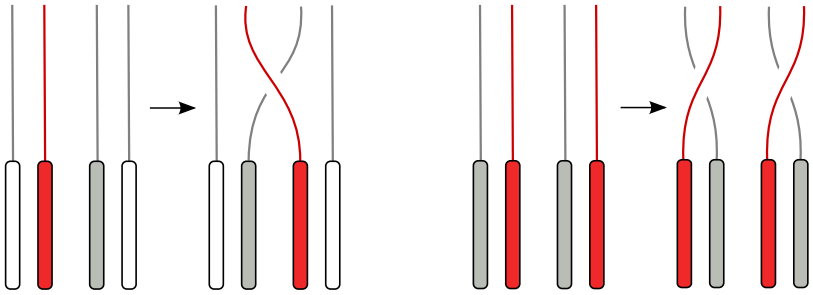
(c) Cookie pillow



(d) Bolster pillow

Figure 2.1: Materials used in the creation of bobbin lace

Step 3) Hang the bobbins. The middle of each thread is draped around an anchoring pin at the top of the pattern with the pair of bobbins hanging down on either side. Often the first row of the pattern is a simple weave to anchor the threads.



(a) Cross *(b) Twist*

Figure 2.2: Cross and twist: the two base actions used in bobbin lace

Step 4) Braid and pin. The lacemaker braids the threads by working with four consecutive threads at a time. It is important to note that the four threads are treated as two pairs of threads: a left and a right pair. Braids are made using two very simple actions. The first action, known as a 'cross' (which we will represent as C where brevity is required), is performed by taking the rightmost thread from the left pair and crossing it over the leftmost thread of the right pair (see Figure 2.2). The second action, known as a 'twist' (denoted T), is performed by crossing the rightmost thread of the left pair over the leftmost thread of the left pair and similarly crossing the rightmost thread of the right pair over the leftmost thread of the right pair. Occasionally, a variation of the twist is used: a 'left-twist' (in which only the left pair is twisted) or a 'right-twist' (in which only the right pair is twisted). During a sequence of braiding actions, the lacemaker may insert a pin to hold the braid in place (see Figure 2.1(b)). The pin provides resistance so that the lacemaker can apply tension to an individual thread without distorting its neighbours. The pinning action (denoted p) is performed by placing a pin between threads into one of the prepared holes. Pinning may take place either in the middle of a braid sequence (after which the pin is 'closed' because it is enclosed by threads) or after the braid (an 'open' pin). A lace braid can be made from combinations of these actions. For example, CT (half-stitch) repeated produces a four stranded plait similar to the three stranded plait used to braid hair. Other commonly used braids are $CTpC$ (cloth-stitch) and $CTpCT$ (whole-stitch). The exact sequence of actions used by the lacemaker depends on the pattern.

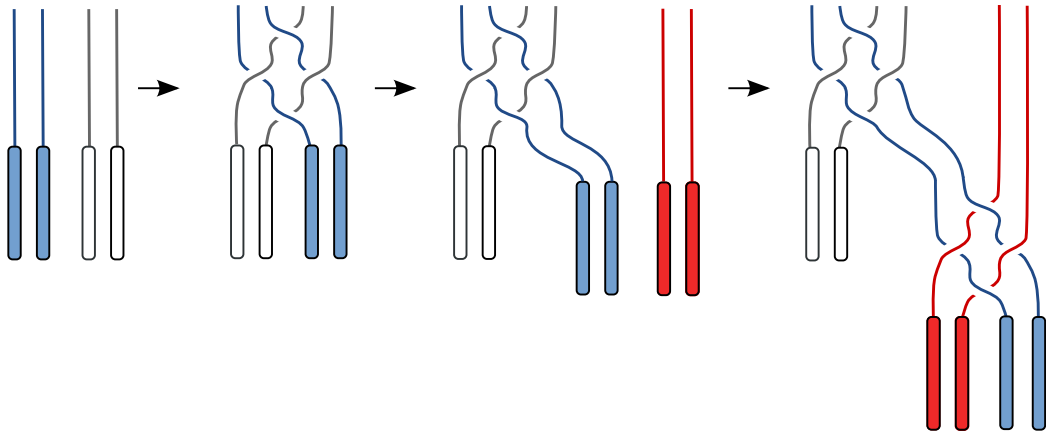


Figure 2.3: Progression of pairs of threads from one set of actions to another

Step 5) Advance to next set. Once four threads have been braided and pinned, the lace maker moves on to braid another set of four consecutive threads. This new set of threads may include two threads from the previous set, as shown in Figure 2.3, but it may also be formed from four completely new threads. Which four threads comprise the next set depends on the pattern.

Steps 4 and 5 are repeated until the lace pattern is completed.

Step 6) Finish. The threads are secured (sometimes with a knot, sometimes by weaving them back into the lace) and trimmed off. The pins are removed and the lace may be lifted off the pillow. Once the pins are removed, the lace is held together by the over and under crossings of the threads. Like knitted or woven material, if an individual thread is snagged and pulled away from the rest of the piece, the lace will distort. However, as long as force is not too great and is applied over a number of threads, well made lace will maintain its shape and arrangement.

Many aspects of the process depend on the 'pattern', so we shall take a closer look at an example and discuss how it is interpreted by the lacemaker. The pattern (e.g., see Figure 2.4 (a) is always in the form of a diagram but the information contained in the diagram can vary quite a bit. The position of a pin is indicated by a dot. Sometimes a decorative thread of a contrasting colour or thickness, known as a *gimp*, is used to outline a region. The path of the gimp is marked with a bold line. In some patterns,

lines are drawn in areas where the thread paths are complicated or ambiguous. In these cases, a line segment represents two threads. Most patterns are accompanied by additional instructions in the form of text or a working diagram. The text often refers to a known *ground* pattern.

Definition (*Lace Ground*). A *lace ground* is a small pattern that can be repeated by periodic tiling to fill a closed region.

The reader may have observed that most lace is made from a single colour of thread. In contrast to fixed-frame loom weaving where threads follow a straight either vertical or horizontal path, the path taken by an individual thread in bobbin lace can be quite complex. As a result, for all but a few simple geometric patterns, it is difficult to assign different colours to threads without creating a seemingly haphazard mix of colours. In bobbin lace designs, texture is used as a replacement for colour providing contrast, shading and interest. In this respect, lace grounds can be viewed as the palette of textures available to the artist.

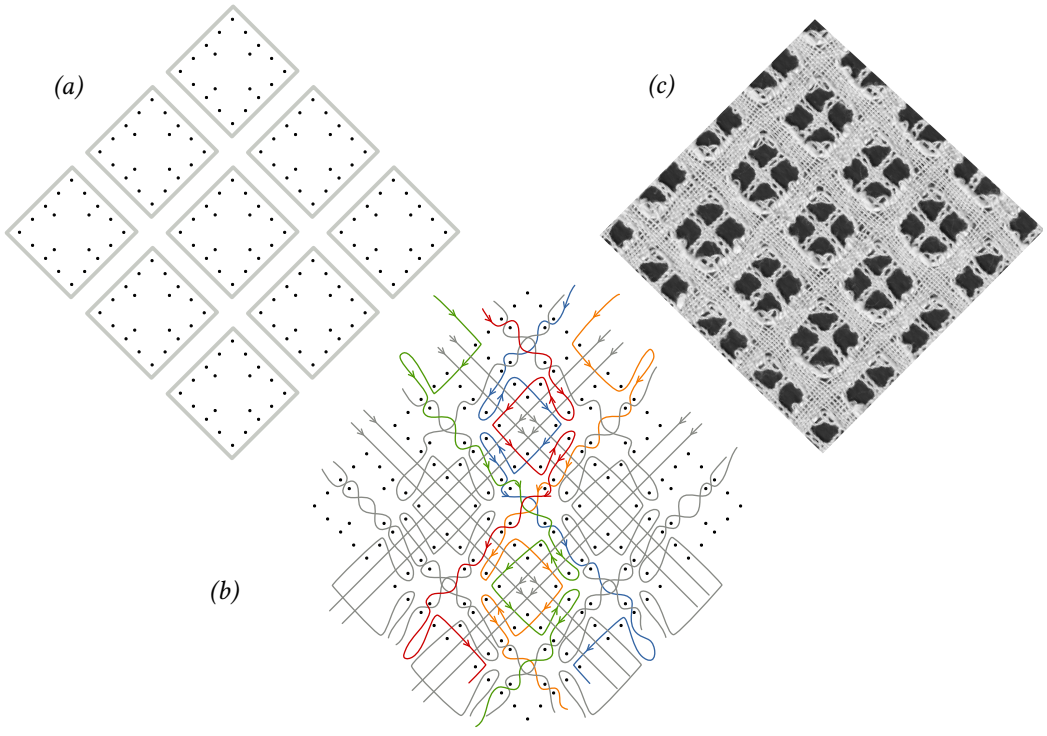


Figure 2.4: A traditional piece of Mechlin bobbin lace ground worked from a pattern in [53, p. A20]. (a) Pricking (b) Working diagram showing paths for pairs of threads (c) Finished lace in linen thread made by author

Lace from a specific geographical area is often characterized by the use of a particular set of grounds well known to the lacemakers of that region. A ground has specific instructions for the actions performed at each pair crossing, the placement of pins and the order in which pairs are combined. Many grounds have been catalogued [17, 53, 85] and commonly used grounds are described in most lace reference books.

A *working diagram* is a line drawing that illustrates how a sub-section of the lace is worked and may depict individual threads or pairs of threads. In working diagrams, cross and twist action combinations are often indicated by colouring the lines (using the International Colour Coding System [87]) or decorating the lines with hatch marks.

This has been a brief overview. For more detailed instructions on equipment and technique, the reader is referred to websites such as [25, 34] or books such as [67, 23].

2.2 Mathematical Modeling of Textiles

The application of mathematical modeling to fibre arts is a fairly new area of research, with most of the focus on weaving and knitting. In the introduction to *Making Mathematics with Needlework* [9], Belcastro and Yackel give a comprehensive overview of its history. Grünbaum and Shepherd [33] have written a seminal paper on geometry in woven fabrics in which they present a formal mathematical model for two particular types of weaves (satins and twills), classify the possible symmetries and use combinatorial methods to discover new patterns.

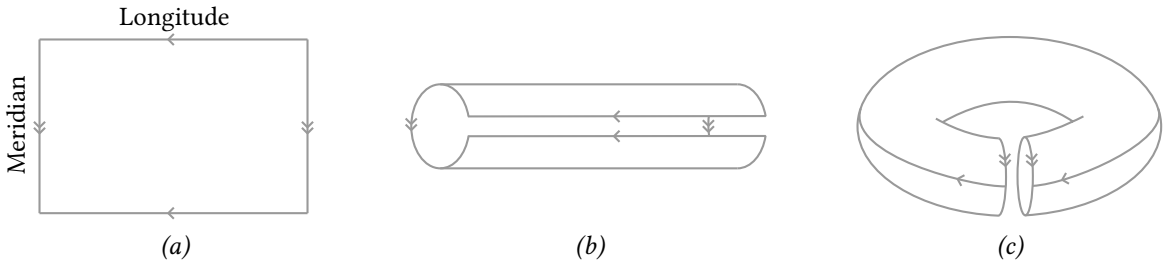
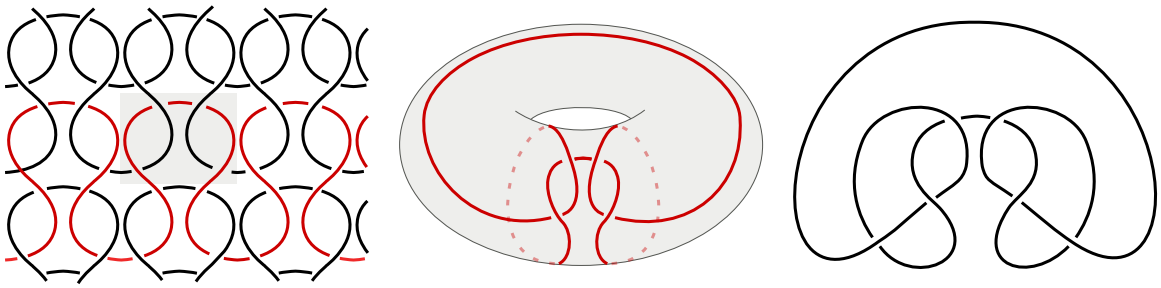


Figure 2.5: (a) Parallelogram with edge markings (b) and (c): Bending of parallelogram to form a torus



(a) A period parallelogram *(b) Wrapped around a torus* *(c) The associated knot*

Figure 2.6: 'Knit' action based on drawings by Grishanov et al. [31]

Grishanov, Meshkov and Omelchenko [31] have examined the structure of machine-made textiles and classified these textiles using the ambient isotopy invariant of knots. Textiles are typically made by repeating an arrangement of fibers in a periodic manner to cover an indefinitely large area. The arrangement of fibers can be represented as a period parallelogram which is translated in two non-parallel directions to create an edge-to-edge tiling of the plane [32]. Periodic repetition in textiles is a stronger property than just simple translation of a wallpaper decoration: fibers that terminate at the edges of the parallelogram must connect with fibers of adjacent copies. This property can be visualized by joining opposite edges of the period parallelogram to form a torus (see Figure 2.5). When wrapped around a torus, the fibers connect, forming a knot or a link as shown in Figure 2.6. The toroidal representation also reduces a pattern description from infinite to finite size without loss of information, a key idea which we will revisit when describing our own model in Section 3.2.

Both Grünbaum et al. [33] and Grishanov et al. [31] make reference to the complexity of hand made lace and exclude it from the scope of their research: "We shall only discuss those fabrics in which the strands are straight and lie in one of two directions, usually at right-angles to each other. *Without these restrictions there are many other possibilities about which extremely little seems to be known* [emphasis added]." [33, p. 139]

In 1994 I had the good fortune to take bobbin lace lessons from the Ottawa Guild of Lacemakers; a hobby and a passion that I have pursued ever since. It is a fascinating art form requiring thought, planning and patience in its execution but is otherwise quite logical and systematic. I hope through this thesis to make inroads into our understanding of this art form and dispel some of the mystery behind it.

2.3 Braid Theory

Artin's theory of braids [5], while only remotely inspired by textiles and lace, gives a precise way of describing an alternating braid which is key to the structure of bobbin lace. A lacemaker, Neff, put forward the idea on an internet discussion forum [65] that bobbin lace patterns can be represented using braid theory, however, no follow up research has been posted as far as we are aware.

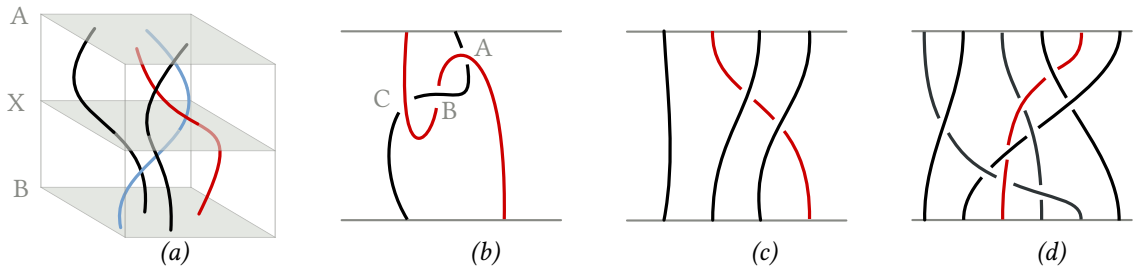


Figure 2.7: a) 3D braid between planes A and B. Plane X intersects each strand exactly once. 2D projections: b) Not a braid because the first strand can not be made monotonic without breaking the monotonicity of the second strand, c) A non-alternating braid, d) An alternating braid.

With some minor exceptions[1], bobbin lace grounds are themselves braids, and, more specifically, they are alternating braids. A *braid* is defined mathematically as a set of n 'strands', each of which is a curve in $\mathbb{R}^3$. The strands travel between two horizontal planes, A and B, such that (i) each strand originates at a unique point on plane A and terminates at a unique point on plane B, (ii) strands do not intersect one another or themselves, and (iii) each strand is monotonic in the direction of a vertical line, meaning that any horizontal plane X between A and B will intersect each strand at just one point (see Figure 2.7 (a)). Braids are often represented as a 2D projection of the 3D object such that the start and end horizontal planes appears as horizontal lines. When two strands cross, one strand is drawn as being above (solid) and the other below (broken). The 2D projection is drawn in general position so that each strand has a unique start and end point and only two strands cross at any point (see Figure 2.7 (c) and (d)).

In the 2D projection, strand positions are labelled 1 to n from left to right. Using standard braid notation [64], σ_i represents a strand in position i crossing **over** its neighbour to the right. Similarly, σ_i^{-1} represents a strand in position i crossing **under** its neighbour to the right. We will use this standard notation to represent the basic cross and twist actions of bobbin lace. As mentioned in the introduction, bobbin lace actions are performed on four threads or two pairs of threads at a time. A mathematically idealized thread with no thickness can be equated to a strand. If we label the pairs from left to right, the two adjacent pairs i and $i+1$ correspond to the four threads in positions $2i-1$, $2i$, $2i+1$ and $2i+2$ where $i \in 1, \ldots, p-1, p$ being the number of pairs of threads in the pattern. The cross action is represented by σ_{2i} and the twist action is represented by $\sigma_{2i-1}^{-1}\sigma_{2i+1}^{-1}$. From this generalized description, we see that σ_x will only occur for even values of x and σ_x^{-1} will only occur for odd values of x.

An *alternating braid* is a braid in which each strand alternates going over and under the strands that it crosses (see Figure 2.7(c) and Figure 2.7(d)). Alternating braids are characterized by the property that the σ generators for even positions have the opposite sign (superscript) from the σ generators for odd positions [64]. Given the generator representation for bobbin lace actions, we infer that any combination of cross and twist will result in an alternating braid.

2.4 Systematic Explorations by Lacemakers

Very little is known about the methods of invention used by the original bobbin lace designers. It is clear that the style of bobbin lace evolved over time as fashions demanded first bold thick designs then designs that seemed to float on air. Designs have also favoured geometric, floral and pictorial motifs at different times. We can only hypothesize that a large amount of trial and error was involved, drawing on older techniques such as passementerie and needle lace.

In modern times, lacemakers have developed new grounds by systematically exploring variations on traditional patterns. One common approach is to take the working diagram from a traditional ground and apply different combinations of cross, twist and pin. This combinatorial approach has been applied extensively for Rose ground (also known as Flanders ground [84, 53]). Pol [71] has developed a web-based tool for visualizing thousands of Rose ground variations.

Another approach is to take a traditional ground and alter the location of pins in the pattern. This change does not affect the topology of the lace but has a significant impact on the shape of the holes. The spaces between the threads contribute as much to the appearance of lace as the threads themselves. This approach is described in [8] and may have been used by Kortelahti [47].

In a more spontaneous approach, contemporary lacemakers also experiment with thread thickness and colour, grid distortions and a free form or pseudo random choice of stitch or pin placements.

2.5 Main Contributions

In this thesis, we present a mathematical model for bobbin lace grounds. The properties of a workable lace pattern shall be expressed in terms of a topological graph embedding of a 2-regular digraph G on the torus, along with its set of defining characteristics, and a mapping from the vertices of G to braid words. The model will leverage previous work in graph theory and facilitate the discovery of new theorems through an exploration of the required properties of the topological graph embedding. The model will then be used to prove that there is an infinite number of workable patterns.

Two different approaches will be used to exhaustively enumerate and generate workable patterns for increasing numbers of vertices. The first approach, which involves gluing lattice paths together in an osculating manner, allowed us to identify over 5 million distinct patterns. This is well in excess of the fewer than 1000 patterns found in lace ground

catalogues. The lattice paths used in this approach are members of a family of paths that have not been previously reported. We explore these paths in detail, develop a recurrence relation and generating function for their enumeration and present a bijection between these paths and a subset of well known paths.

In the second approach, we wish to draw out of the extremely large number of patterns some of the more aesthetically interesting cases for lacemakers to work on. A high degree of symmetry is used as the criteria for aesthetic interest. We demonstrate, by computational generation, that for each of the 17 planar periodic symmetry groups there exist lace ground patterns possessing that symmetry.

Notes

1. Techniques such as ‘sewings’ do not produce braids because the threads cross back upon themselves. Patterns involving a ‘lazy’ crossing (a crossing in which two or more consecutive threads are treated as one and cross over or under other threads as a group) result in braids that are not alternating. These techniques are outside of the scope of this thesis.

Chapter 3

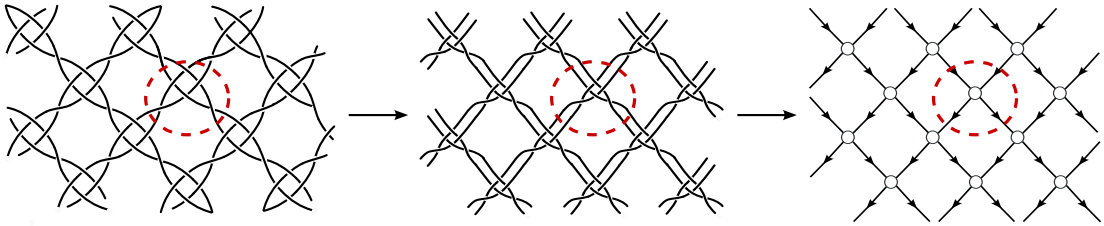
A Mathematical Model

In this chapter, the technique used to create bobbin lace is translated into a mathematical model based primarily on graph theory. It will then be possible to manipulate the model using known techniques from graph theory.

3.1 Two Components

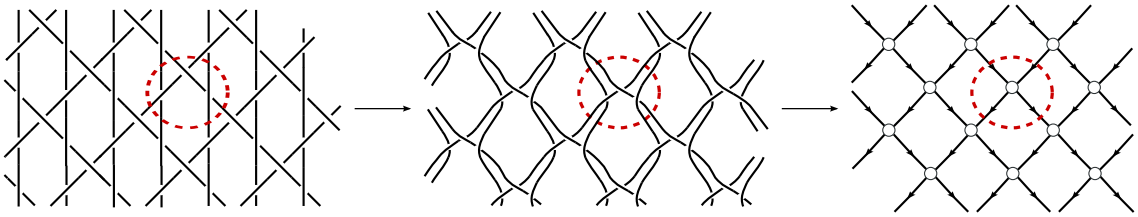
As discussed in Section 2.1, a bobbin lace tessellation is created by braiding together an even number of threads in groups of four. The braids are formed from a sequence of cross, twist and pin actions which we will denote as C, T and p.[1] As discussed in Section 2.3, the cross and twist actions can be represented in terms of braid word generators: $C = \sigma_{2i}$, $T_L = \sigma_{2i-1}^{-1}$, $T_R = \sigma_{2i+1}^{-1}$, $T = T_L T_R = T_R T_L$ where i is the left to right index of a pair of threads starting at one. To fully describe the pin action, one must specify the index j of the thread that will be to the left of the pin and the (x, y) coordinates where the pin will be inserted: $p(j, x, y)$. In our current research, details of the pin action are not a primary concern and will be represented merely as p.

Because of its construction technique, a bobbin lace tessellation can be decomposed into two key components which shall be illustrated using a well known pattern called *Torchon ground* shown in Figure 3.1(a). To formalize this idea, the term interaction is defined. Consider one instance of four consecutive threads and the braid that is formed from them.



$\zeta(v) = CTpCT \,\forall\, v \in V$

(a) Torchon ground



$\zeta(v) = CTp \,\forall\, v \in V$

(b) Half-stitch ground

Figure 3.1: Two example lace tessellations
left: Diagram of individual threads. **middle**: Exaggerated distance between interactions. **right**: Planar embedding of digraph representing pair movement.

Definition (*Interaction*). An *interaction* is a sequence of actions on four consecutive threads labelled a, b, c, d which begins when thread b first crosses over thread c and ends when any of the four threads a, b, c, d crosses over or under a thread x where $x \notin \{a, b, c, d\}$.[2]

The sequence of actions in an interaction is a non-empty combination specified by the regular expression $C\{C, T, p\}^*$ and including at least one T. In Figure 3.1, a red oval is used to highlight an instance of an interaction.

To illustrate how the Torchon ground is composed of multiple interactions, the interactions in Figure 3.1(a) have been decreased in size in the middle figure. In this exaggerated diagram, we can see more clearly that there are two parallel threads entering the interaction from top-left and two from top-right. Similarly, at the bottom, two parallel threads leave on the left side and two on the right side. The entire lace tessellation can be represented as a set of interactions with pairs of threads travelling between interactions.

We take this idea one step further on the right side of Figure 3.1(a) and shrink each interaction to a dot, abstracting away the actual thread crossings that occur between the four threads. Because threads travel between interactions as a pair without any additional crossings, we can simplify the diagram by using a single line to represent two threads. An interaction has a distinct start and end, one interaction must be finished before threads can move on to the next interaction. To represent this property, the lines between interactions are assigned a direction, indicated by an arrow, allowing us to specify the orderly flow of threads between interactions.

We shall introduce terminology that will help describe the pattern abstraction more precisely. The main reference we will use for graph theory is *Graphs on Surfaces* by Mohar and Thomassen [62]. For terms in topology we will refer to *General Topology* by Willard [89]. For ease of reference, some of the key definitions will be repeated here.

Definition (*Graph*). A graph G is a pair of sets $V(G)$ and $E(G)$, where $V(G)$ is nonempty and $E(G)$ is a set of 2-element subsets of $V(G)$. The elements of the set $V(G)$ are called *vertices* of the graph G, the elements of $E(G)$ are the *edges* of G. For an edge $e = (u, v) \in E(G)$, the vertices u and v are called *endvertices* of e. [62, p. 3]

The graphs considered in this thesis are *multigraphs* meaning that distinct edges are allowed to have the same pair of endvertices. If each edge is an **unordered** pair of endvertices, we will refer to the graph as an *undirected graph* and represent the edge as $e = (u, v)$. If each edge is an **ordered** pair of endvertices, we will refer to the graph as a *directed graph* and represent the edge as $e = \{u, v\}$ where the direction of the edge is from u to v. The term directed graph will often be contracted to *digraph* for brevity.

Definition (*Simple arc*). Let X be a topological space. An arc in X is the image of a continuous function $f : [0, 1] \to X$. The arc is simple if f is $1-1$. [62, p. 18]

Definition (*Topological graph embedding*). A graph G is embedded in a topological space X if the vertices of G are distinct elements of X and every edge of G is a simple arc connecting in X the two vertices which it joins in G, such that its interior is disjoint from other edges and vertices. [62, p. 19]

Definition (*Surface*). A surface is a connected compact Hausdorff topological space S which is locally homeomorphic to an open disk in the plane. All surfaces discussed in this thesis are closed and oriented. [62, p. 78]

Definition (*Genus of a surface*). The genus g of an orientable surface S is the number of handles that must be added to a sphere in order to obtain the surface S. [62, p. 81]

The notation $\Delta_g(G)$ will be used to represent a topological embedding of a digraph G on a surface of genus g. We shall represent the decomposition of a bobbin lace tessellation into two components by introducing the term *tesselace pattern*.

Definition (*Tesselace Pattern*). A *tesselace pattern* is a pair $(\Delta_0(G^\infty), \zeta(v))$. The first element, $\Delta_0(G^\infty)$, is a topological embedding of an infinite, directed graph, $G^\infty = (V(G^\infty), E(G^\infty))$ in the plane. Each vertex in $V(G^\infty)$ corresponds to an interaction and a directed edge $\{u, v\} \in E(G^\infty)$ corresponds to a pair of thread segments travelling from interaction u to interaction v. The second element, $\zeta(v)$, is a mapping from vertices to action sequences i.e., $\zeta(v) : V \to C\{C, T, p\}^*$.

For brevity, we have coined the word 'tesselace' as a contraction of 'tessellation' and 'lace'. The name was inspired by the Czech translation of tessellation which is 'teselace'. [3]

For Torchon ground, a representative subset of $\Delta_0(G^\infty)$ is shown on the far right of Figure 3.1(a). It bears some resemblance to the pair working diagram shown in Figure 2.4. The action sequence for Torchon ground is $\zeta(v) = CTpCT$ for all $v \in V$. Figure 3.1(b) shows a second example of a tesselace pattern called *Half-stitch ground*. It has the same topological embedding $\Delta_0(G^\infty)$ as Torchon but a different sequence of actions: $\zeta(v) = CTp$ for all $v \in V$. Notice that a significantly different appearance results from the two $\zeta(v)$ mappings.

As discussed in Section 2.4, lacemakers have given considerable attention to the systematic analysis of different $\zeta(v)$ mappings. For example, thirty-six variations of the tesselace pattern known as Rose ground (also known as fond de mariage, fond à la vierge, cinq trous or Flanders ground) can be found in [85, pp. 130-143]. These variations apply different cross, twist combinations to the same topological embedding.

Our contribution will focus on the motion of pairs of threads between interactions represented by $\Delta_0(G^\infty)$.

3.2 Properties of a Tesselace Embedding

Our goal is to identify the subset of infinite digraphs embedded in the plane that can produce workable lace patterns. By *workable*, we mean that the pattern can be used to make a physical piece of lace from thread following traditional construction methods. The criteria for workable bobbin lace will explored in this section. We will use the term *tesselace embedding* to refer to a topological embedding of a digraph G that possesses all of the necessary properties required to create workable lace when used in a tesselace pattern. The formal definition of a tesselace embedding will appear in Section 3.3 after we have laid down some ground work.

A tesselace embedding possesses the five fundamental properties outlined below.

1) Bobbin lace is constructed by braiding four threads at a time; two pairs of threads enter an interaction and two pairs leave.

Property 3.2.1 (2-Regular). *Every vertex in the directed graph G^∞ of a tesselace embedding has two incoming edges and two outgoing edges; such graphs are known as 2-regular digraphs.*

2) Bobbin lace grounds have a doubly periodic structure, meaning that the pattern can be translated in two non-parallel directions and appear unchanged. We refer the reader to *Tilings and Patterns* [32] by Grünbaum and Shepherd for a detailed analysis of periodic structures and their properties. The periodic structure of bobbin lace allows us to represent the infinite digraph of the tesselace embedding in a much more compact form — namely, as a finite graph embedded on the torus.

Definition (*Covering*). We say that a graph G covers a graph H if there exists a map $f : V(G) \to V(H)$ such that f is onto and for any vertex v of G, the restriction of f to the neighbourhood $N(v)$ is a bijection to $N(f(v))$. [62, p. 201]

Property 3.2.2 (Periodic). *Let M be a maximal set of vertices in $\Delta_0(G^\infty)$ such that no two vertices are equivalent under periodic translation. Let H be the sub-digraph induced in G^∞ by M. The infinite topological embedding $\Delta_0(G^\infty)$ of a tesselace embedding can be represented by a finite topological embedding $\Delta_1(G^\tau)$ where:*

1. $V(G^\tau) = V(H)$
2. $E(G^\tau) = E(H) \cup \{\{u, C(w)\} : u, C(w) \in V(H), w \in V(G^\infty - H)$ *such that G^∞ has an edge $\{u, w\}$ and $C(w)$ is equivalent to w under periodic translation*$\}$.
3. *The digraph G^∞ covers G^τ.*
4. *G^τ is topologically embedded on the torus.*

We shall refer to $\Delta_1(G^\tau)$ as the *fundamental embedding* of the tesselace embedding which is analogous to the fundamental domain used to describe a periodic tiling.

The relationship between $\Delta_0(G^\infty)$ and $\Delta_1(G^\tau)$ can be visualized by choosing a period parallelogram (a smallest parallelogram such that no two vertices within the bounds of the parallelogram are equivalent under periodic translation) in the infinite embedding of the tesselace embedding in the plane and identifying opposite sides to form a torus. Edges crossing the boundary of the parallelogram wrap around and connect to vertices within the bounds of the parallelogram that are images of the original endvertices in G^∞ under translation (see Figure 3.2).

3) If you lift a piece of lace off the table, none of the threads are left behind. The requirement is that the entire lace piece must hang together. In order for this to be true in a general way, it must also apply to any closed region filled by a tesselace pattern. Again we shall introduce some terminology in order to give a more precise description of this property.

Definition (*Path*). A path P_n on n vertices is the graph with vertices $\{v_1, v_2, \ldots, v_n\}$ and $n - 1$ edges (v_i, v_{i+1}), $1 \leq i \leq n$. Vertices in the path are distinct as are edges. [62, p. 4]

Definition (*Cycle*). A *cycle* is a path P_n with the addition of the edge (v_1, v_n). [62, p. 4]

A directed path or directed cycle is one that respects edge orientations in the digraph.

Definition (*Connected*). An undirected graph G is connected if all pairs of vertices are connected by a path in G. A directed graph G is connected if its underlying undirected graph is connected. [62, p. 4]

Definition (*Homotopy*). Let f and g be continuous functions from topological space X to topological space Y. We say that f is homotopic to g if and only if there is a function $H : X \times [0, 1] \to Y$ such that $H(x, 0) = f(x)$ and $H(x, 1) = g(x)$ for all $x \in X$. The map H is a homotopy between f and g. [89, p. 223]

Definition (*Contractible Cycle*). A cycle embedded on surface S is *contractible* if it is homotopic to a point.

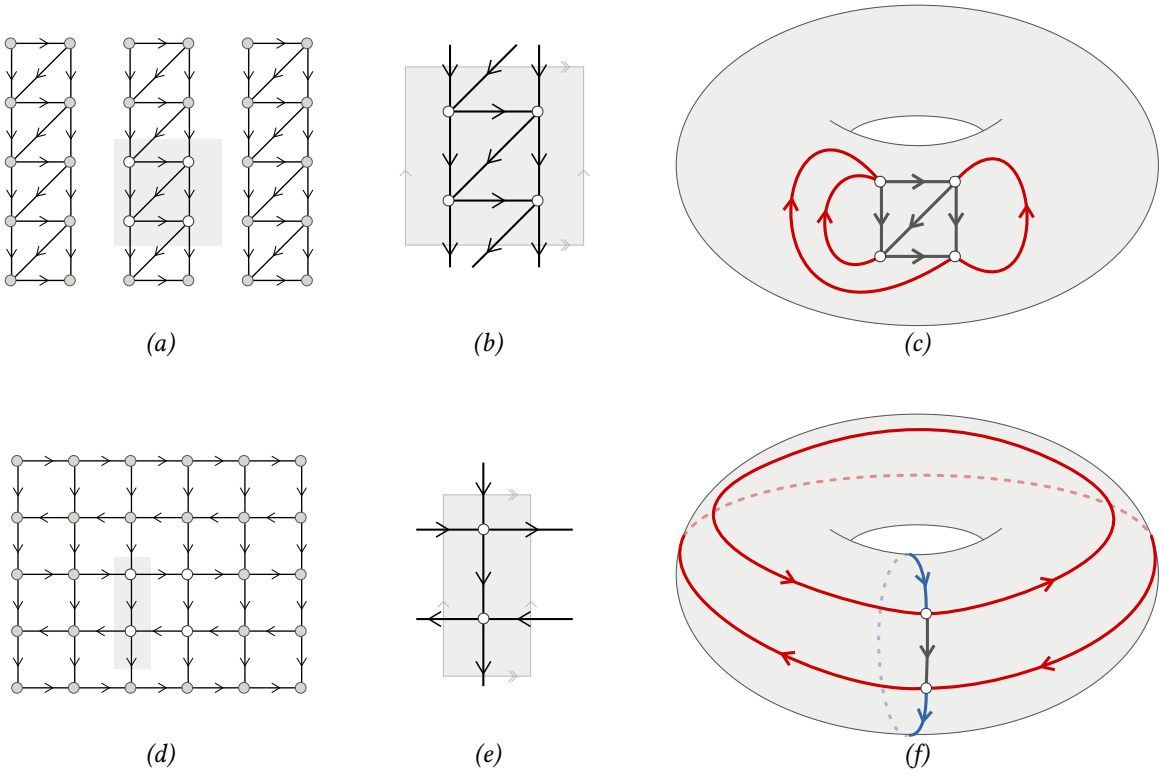


Figure 3.2: **a**,**b**,**c**: The digraph G^τ is connected but the corresponding digraph G^∞ has multiple components. **d**,**e**,**f**: The digraph G^τ is connected and the fundamental embedding includes edges across all four boundaries of the period parallelogram. The corresponding digraph G^∞ is connected. **a**,**d**: A subregion of an infinite graph embedded in the plane, **b**,**e**: a period parallelogram for the embedding, and **c**,**f**: the period parallelogram wrapped around a torus.

It is fairly self-evident that for a given tesselace embedding the digraph of G^∞ must be connected in order for the associated pattern to hang together. For convenience, we wish to express this same property in terms of the fundamental embedding. Independent of which period parallelogram is selected in $\Delta_0(G^\infty)$, the subgraph $H \subseteq G^\infty$ induced by vertices within the bounds of the parallelogram is connected. Without loss of generality, we can assume that all the edges of H can be drawn within the bounds of the parallelogram. If H' is an adjacent translated copy of H, then H and H' must be connected by at least one edge across the border of the parallelogram. When the sides of the parallelogram are identified to form a torus, we wish to show that an edge connecting H to H' becomes an edge in a non-contractible cycle. Since H is connected, there exists a path between any pair of vertices u and $v \in V(H)$. Let v' be a vertex in H' that is an image of v in G^∞ under periodic translation. If $\{u, v'\}$ is an edge connecting H and H' across a border of the period parallelogram, then, when the border wraps around to meet its opposite side to form a torus, the edge $\{u, v'\}$ maps to the edge $\{u, v\}$ thus closing the path from u to v and forming a cycle. The border of the parallelogram corresponds to either a meridian circle or a longitudinal circle in the torus (see Figure 2.5). The cycle intersects the border once and is therefore non-contractible.

Definition (*Rotation system*). Assume that G is embedded in a surface S. Let $\pi = \{\pi_v \mid v \in V(G)\}$ where π_v is the cyclic permutation of the edges incident with the vertex v such that $e_{i+1} = \pi_v(e_i)$ is the successor of e_i in the clockwise ordering around v. The cyclic permutation of π_v is called the local rotation of v, and the set π is the *rotation system* of the given embedding of G in S. [62, p. 90]

Definition (*Combinatorial embedding*). A combinatorial embedding of a graph G is a pair $\Pi = (\pi, \lambda)$ where $\pi = \{\pi_v \mid v \in V(G)\}$ is a rotation system and λ is a signature mapping which assigns to each edge $e \in E(G)$ a sign $\lambda(e) \in \{-1, 1\}$. [62, p. 99]

Because the fundamental embedding has non-contractible cycles in both the meridian and longitude directions of the torus on which it is embedded, we can conclude that the combinatorial embedding associated with the fundamental embedding has a genus of one [30]. In polynomial time one can count the faces of a combinatorial embedding via a facial walk and, applying Euler's formula, determine its genus.

We can now state the connected property of a tesselace embedding in terms of its fundamental embedding:

Property 3.2.3 (Connected). *For a tesselace embedding, the digraph G^τ must be connected and the combinatorial embedding of $\Delta_1(G^\tau)$ must have a genus of one.*

4) Bobbin lace is an alternating braid. The mathematical definition of a braid specifies that all strands are simultaneously monotonic with respect to a vertical axis. By continuous deformation, the threads of a bobbin lace pattern can be made monotonic while preserving the crossings of its 2D projection. However, in practice, while there is a preferred direction for threads, which we will specify as downward or meridional for the purpose of this thesis, they may travel horizontally and even upward for short distances as shown in the traditional ground pattern in Figure 2.4.

Another way to describe the monotonic property of a braid is to say that sequential crossings always occur in a forward direction. Starting at the top end of a strand and following it to its bottom end, we can construct a chain by labelling the crossings in the order in which they are encountered. For a braid, the union of all chains must form a partially ordered set. For example, in Figure 2.7(b), tracing the black strand gives the chain $A \prec B \prec C$ while tracing the red strand yields the chain $C \prec B \prec A$. The union of these two chains does not form a partial order, therefore, we can conclude it is not a braid.

A directed cycle in a digraph embedded in the plane represents a circular dependency which is not a valid order. On the torus, this corresponds to a directed contractible cycle.

Property 3.2.4 (Partially Ordered). *For a tesselace embedding, the combinatorial embedding $\Pi_0(G^\infty)$ does not have any directed cycles and the associated combinatorial embedding $\Pi_1(G^\tau)$ is free from contractible directed cycles.*

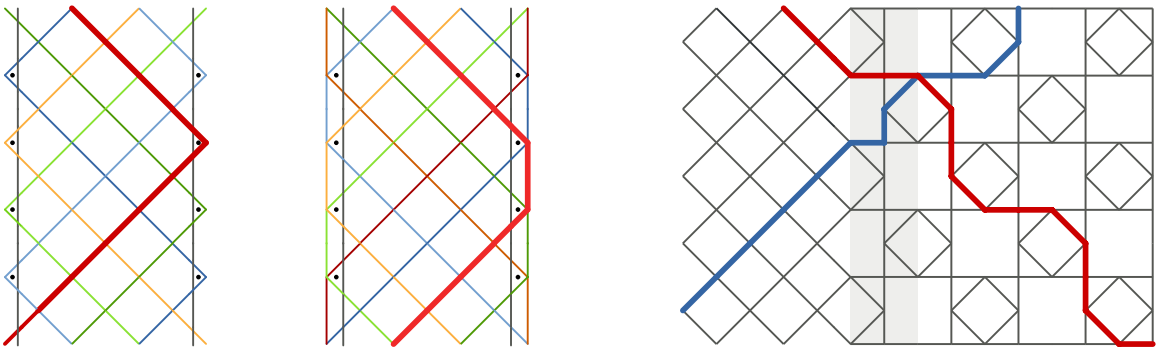
5) Loose ends, caused by cutting threads or adding new ones, are undesirable because they inhibit the speed of working, can fray or stick out in an unsightly manner and, perhaps most importantly, degrade the strength of the fabric.[4] In this thesis, loose ends are disallowed by insisting on a property we will call conservation.

Property 3.2.5 (Thread conserving). *A tesselace pattern is thread conserving. That is, translated copies of the periodic pattern will fill a rectangle of fixed width and unbounded length using a single, finite set of threads.*

To visualize the key point in Property 3.2.5 — using a single, finite set of threads — one can think of creating a rectangular piece of lace that consists of just one ground. To start, the lacemaker hangs a fixed number of pairs of threads on pins at the top of the pattern. At any row after the start, regardless of how long the rectangle may be, additional pairs are not required. If one period of the pattern requires n pairs of bobbins to complete and k repeats of the period are required to span the width of the rectangle, then $k \times n$ pairs of threads must be hung on at the start of the piece. Until the end of the rectangle is reached, no pairs are cut off and no pairs are added.

In general, in this thesis we think of the fabric created from a pattern in an idealized way without concern for what happens at the sides. However, because it has some bearing on understanding the conservation of threads, we shall give a brief overview. As we have seen in Property 3.2.3, a period parallelogram has pairs of threads entering from the top, exiting from the bottom and a mix of both entering and exiting pairs on the left and right sides. What happens when there is no adjacent copy of the period parallelogram for the threads to enter? For the property of thread conservation, we are particularly interested in what happens when there is no adjacent copy to the left or right. There are two cases to consider: (1) The side of the rectangle is an edge or 'selvage' of the fabric. (2) The side of the rectangle R abuts another region S that is filled with a different ground. In case (1), the pairs of thread reflect back into the rectangle. A 'footside' is a type of small bobbin lace pattern specifically designed to reflect pairs back into the pattern. There are several commonly used footside patterns, two of which are shown in Figures 3.3(a) and (b). Conservation of threads in a ground requires that every outgoing edge on the side can be matched with an incoming edge on the same side to complete the reflection. In case (2), pairs of threads will exit R and enter S along their adjoining edge and vice versa. Every pair exiting R is replaced by a pair coming into R. Exactly how this is done depends on the two grounds filling R and S but in general it requires that every outgoing edge on a side can be matched with an incoming edge on the same side. A simple example is shown in Figure 3.3 (c).

For an example of a digraph embedding in which outgoing edges along the side of the period parallelogram do not match up with incoming edges on the same side, see Figure 3.8(b).



(a) *(b)* *(c)*

Figure 3.3: Behaviour at edge of shape. Line segments of the same colour show the path taken by one pair of threads when $CTCT$ is used at every interaction. (a) & (b) Examples of traditional footsides. (c) Example of two grounds that abut.

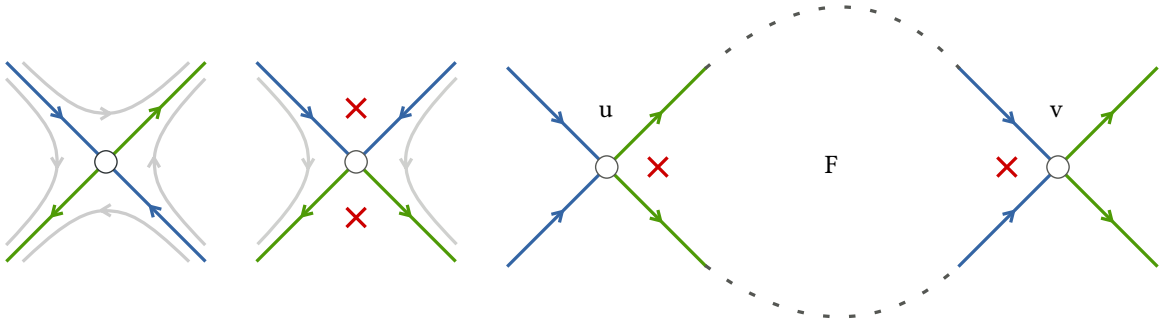
We wish to emphasize that thread conservation is specific to filling a right angled parallelogram, i.e., a rectangle. If a parallelogram of some other angle is the basis for a ground's thread conservation, threads of that ground will be biased to lean left or lean right. If one ground in a larger lace piece leans at a particular angle then all grounds in that piece must lean at the same angle to allow shapes to connect together in a continuous fashion.[5] Most likely influenced by the tradition of woven cloth, the common angle chosen by bobbin lacemakers was the right angle. As an aside, it would be interesting to see what designs could result from choosing a different angle. Lace could be worked on a cylinder, for example, with set of grounds that all lean $45°$.

From now on we will consider the period parallelogram of a tesselace embedding to be rectangular. This does not prevent a tesselace pattern from possessing other symmetries as will be discussed in Chapter 7, however, at the most basic level, all tesselace embeddings are periodic in two **perpendicular** directions.

In the following section we will determine the necessary and sufficient conditions for thread conservation.

3.3 Conservation of Threads

In order to prove a theorem that characterizes thread conservation, we shall first prove two auxiliary lemmas.



(a) Alternating *(b) Consecutive* *(c) Incoming and outgoing blocking vertices on a face boundary*

Figure 3.4: Edge arrangements around a vertex

Directed edges in an embedding of a 2-regular digraph can be arranged in one of two possible ways around a vertex: either *rotationally alternating* in which edges alternate between incoming and outgoing directions or *rotationally consecutive* with edges in the order incoming, incoming, outgoing, outgoing (see Figure 3.4(a) & (b)). We shall refer to a vertex, with reference to its edge arrangement, as a *rotationally alternating vertex* or a *rotationally consecutive vertex.*

Lemma 3.3.1. *Let $\Pi_g(G)$ be a 2-regular digraph with n vertices and a combinatorially embedding of genus g. If $\Pi_g(G)$ has fewer than $2-2g+n$ rotationally consecutive vertices it will contain a contractible directed cycle.*

Proof. From the Euler characteristic, we can calculate that $\Pi_g(G)$ in Lemma 3.3.1 has $2-2g+n$ faces. We shall prove that if the number of rotationally consecutive vertices is less than the number of faces in $\Pi_g(G)$ then a contractible directed cycle will exist.

Consider a face F of $\Pi_g(G)$ with a vertex a on its boundary. For every vertex a with a rotationally alternating edge configuration, the edges incident to F at a form a directed path. If every vertex in the boundary of F has a rotationally alternating edge configuration, then the face boundary is a contractible directed cycle. Assume that F has at least

one vertex c with a rotationally consecutive edge configuration and that the edges incident on F at c are either both incoming or both outgoing. Since an outgoing edge at c is an incoming edge at the next vertex in the boundary of F, it follows that two outgoing edges at c must be balanced by two incoming edges incident to F at another vertex on the boundary of F (see Figure 3.4(c)). Therefore, a walk around the face must encounter at least two rotationally consecutive vertices on its boundary.

The vertex c prevents a directed cycle for two (of the at most four) incident faces, one blocked by its incoming edges and one by its outgoing edges. To block a directed cycle around a face, each face must have at least two rotationally consecutive vertices. Therefore there must be at least as many rotationally consecutive vertices as faces. □

By applying Lemma 3.3.1 to embeddings in the plane ($g = 0$) and in the torus ($g = 1$), we derive the following corollaries:

Corollary 3.3.2. *A finite 2-regular digraph embedded on the plane will always have at least one face bounded by a contractible directed cycle.*

Corollary 3.3.3. *If the arrangement of edges around any vertex of a 2-regular digraph embedded on the torus is rotationally alternating, the embedding will have a contractible cycle.*

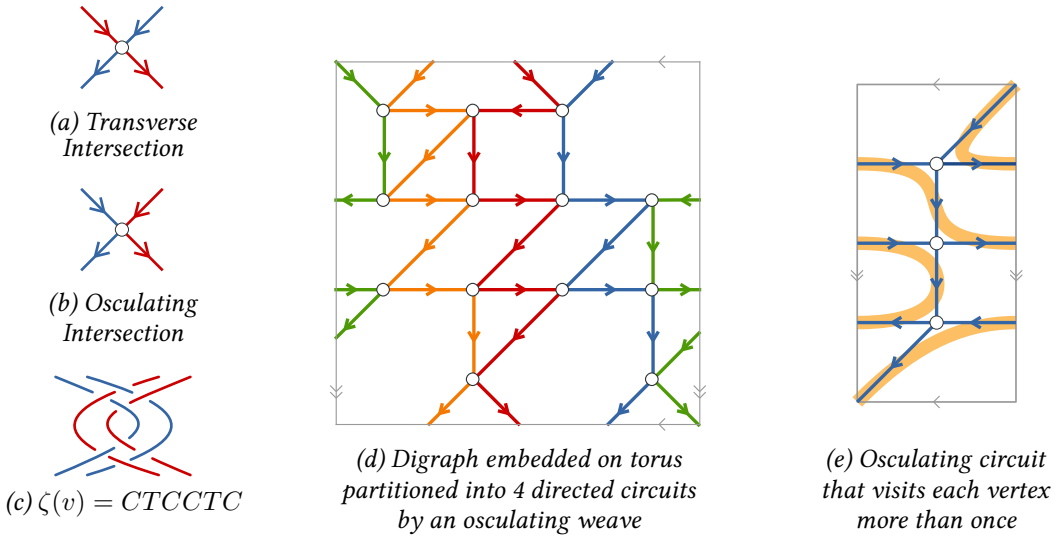


Figure 3.5: (a) & (b) Two crossing types on a 2-regular digraph. (c) A braid word in which pairs of threads exit on the same side as they enter an interaction. (d) & (e) Examples of an osculating partition of a tesselace embedding.

Definition (*Walk*). A sequence $W = v_0 e_1 v_1 \dots e_k v_k$ $(k \geq 0)$ of vertices and edges of G such that e_i is an edge joining the vertices v_{i-1} and v_i $(1 \leq i \leq k)$ is said to be a walk in G. If there is no repeated edges in a walk it is called a *trail*. The trail is closed, i.e., $v_k = v_0$, it is called a circuit. [62, p. 4]

A directed walk or directed circuit is one that respects edge orientations in the digraph.

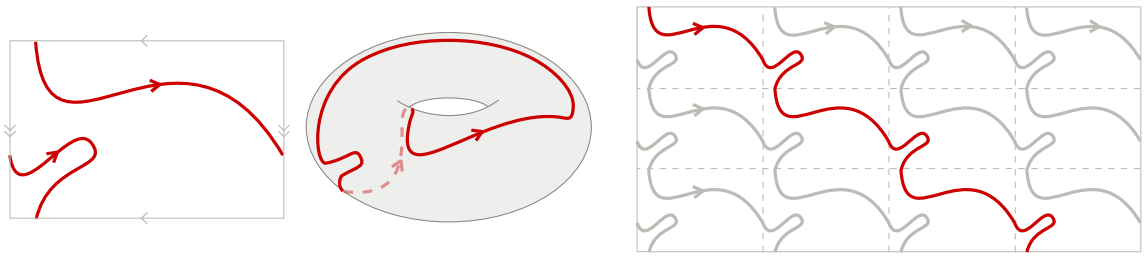
A *transverse intersection* of walks occurs in an embedding when one walk crosses another walk as illustrated in Figure 3.5(a). In contrast, an *osculating intersection* of walks occurs when two walks meet but continue without crossing (see Figure 3.5(b)). Osculating intersections are also referred to as *kissing intersections*.

Lemma 3.3.4. *Let $\Pi_1(G)$ be a combinatorial embedding of a 2-regular digraph $G(V, E)$ in the torus. If $\Pi_1(G)$ does not contain any contractible directed cycles then the edges of G can be partitioned into a set of directed circuits such that at each vertex in the circuit, the intersection of edges is osculating.*

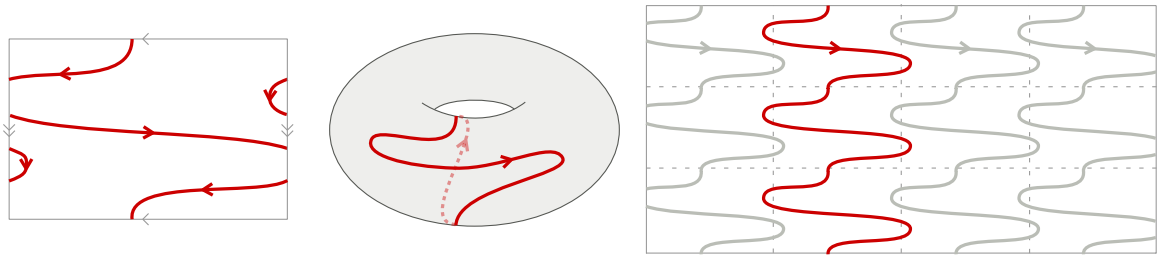
Proof. Choose a directed circuit K by selecting any directed edge $(u, v) \in E$. Then add the unique, rotationally consecutive outgoing edge $(v, w) \in E$ to K. The existence and uniqueness of (v, w) is guaranteed by Corollary 3.3.3. The process is repeated using vertex w and so on until K returns to its initial vertex u via edge $(z, u) \in E$ where (z, u) is rotationally consecutive to (u, v) at vertex u. Note: If K returns to its initial vertex u via the edge (z', u) which is not rotationally consecutive to (u, v), continue tracing the circuit until it returns to u a second time.

The circuit K is guaranteed to complete because G is finite and 2-regular. Remove all edges traversed by K from $\Pi_1(G)$. The resulting embedding still has an equal number of incoming and outgoing edges at each vertex and the relative rotational order of the remaining edges is unchanged so the process can be repeated until zero edges remain. The resulting set of circuits will not intersect each other transversely because at each stage, the selected edges were consecutive. □

Figure 3.5(d) shows an example of the osculating partition of a graph embedding.



(a) Circuit wraps once around torus longitudinally and once meridionally



(b) Circuit wraps once around the torus meridionally, no longitudinal wrapping

Figure 3.6: Circuit on a torus

Consider the torus represented as a rectangle with opposite borders identified and the meridional direction is vertical. From topology we can state that an oriented closed curve which wraps longitudinally around the torus L times will cross a meridian circle a net of L times, where a left to right crossing is positive and a crossing in the opposite direction is negative. An analogous relation exists for a closed curve that wraps meridionally around the torus M times. It will cross a longitudinal circle M times. We can therefore describe the way a curve wraps around a torus using the pair (M, L) which we will refer to as the *wrapping index*.

Theorem 3.3.5. *Let $\Delta_1(G^\tau)$ be a topological embedding of a 2-regular digraph on the torus that has no contractible directed cycles. The embedding is thread conserving if and only if $\Delta_1(G^\tau)$ can be partitioned into a set of non-contractible osculating circuits each of which is homotopic to an oriented closed arc with wrapping index $(1, 0)$.*

Proof. As a consequence of Lemma 3.3.4, the edges of $\Delta_1(G^\tau)$ can be partitioned into a set C of osculating directed circuits, each of which is non-contractible. By using $\zeta(v) = CTCpCTC$ for all vertices in $\Delta_1(G^\tau)$ (see Figure 3.5(c)), a pair of threads will follow the arcs of a circuit in this partition set.

A directed circuit in the fundamental embedding $\Delta_1(G^\tau)$ that wraps longitudinally around the torus L times corresponds to a walk in the associated unbounded periodic graph $\Delta_0(G^\infty)$ with a horizontal distance between start and finish vertices of L times the width of the fundamental rectangle. For a constant number of threads to cover a rectangle of fixed width and indeterminate length, there must not be ant horizontal displacement of the walk between one repeat and the next, implying that $L = 0$. All circuits in C must therefore have a wrapping index of $(M, 0)$ where $M \geq 0$.

A directed circuit in C with a wrapping index of $(M, 0)$ can only connect back to its starting vertex if it wraps zero or one times in the meridional direction. All other walks back to the beginning of the circuit involve a transverse crossing. A wrapping index of $(0, 0)$ corresponds to a contractible cycle which is not allowed. Therefore the wrapping index of the circuit is $(1, 0)$. Since the circuits in C do not intersect each other transversely, they all have the same wrapping index. Therefore, all circuits are homotopic to an oriented closed arc with wrapping index $(1, 0)$.

This result can be generalised for any $\zeta(v)$ function by noting that an interaction is a mathematical braid which, by definition, conserves the number of strands. Therefore, replacing $\zeta(v) = CTCpCTC$ with any valid $\zeta(v)$ will not alter the number of threads required. □

We conclude this section with a final observation about the topology of the tesselace embedding as required by thread conservation.

Definition (*Dehn twist*). A Dehn twist on a surface S is achieved by cutting S along a closed curve, rotating one of the resulting boundaries by $360°$ and gluing along the boundaries of the cut to form the surface S'.

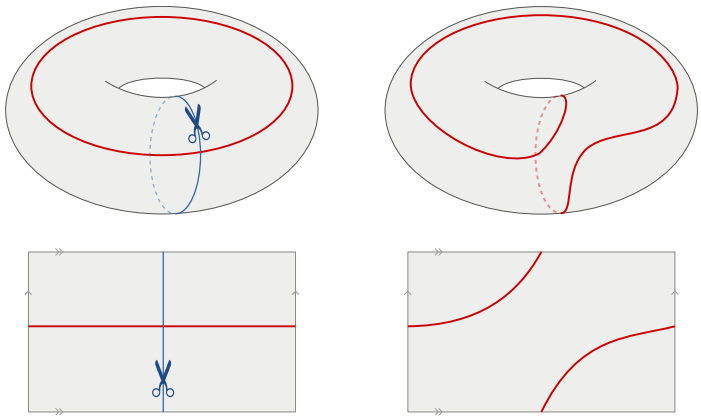


Figure 3.7: A closed path on a torus before and after a Dehn twist. The torus is cut along the blue curve and ends are rotated $360°$ around the meridian.

A Dehn twist preserves all topological properties of the surface, namely a neighbourhood on the original surface is still a neighbourhood on the surface after the mapping, a contractible closed curve is still a contractible closed curve and a non-contractible closed curve is still non-contractible. The Dehn twist operation, proposed by Dehn in 1938 [21], affects the wrapping number of any curve a that crosses b.

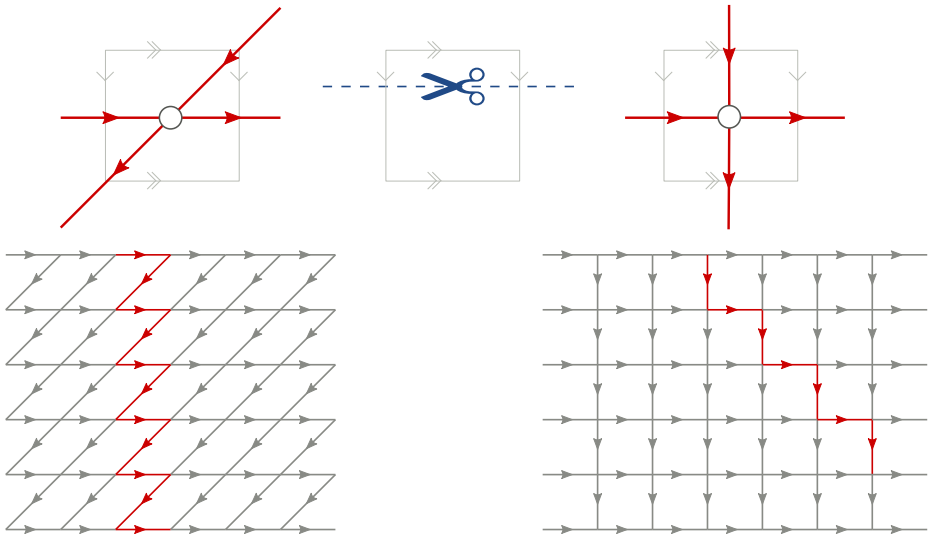


Figure 3.8: Effect of Dehn twist on a tesselace embedding: (a) A workable tesselace embedding. (b) The same graph embedding after a Dehn twist (indicated by the dashed blue line) making it unworkable.

As shown in Figure 3.8, the effect of a Dehn twist along a longitudinal curve has a direct impact on the wrapping number of an osculating path and therefore on whether a graph embedding is thread conserving

It is important to note that when a Dehn twist is applied to a graph embedding, it changes the wrapping index of the osculating circuits but has no impact on the rotational order of the edges around a vertex. For a given combinatorial embedding that meets all the other requirements of a tesselace embedding, there may not exist a sequence of Dehn operations that produces a set of osculating circuits with wrapping index $(1, 0)$. If one does exist, it may not be unique as shown Figure 3.9.

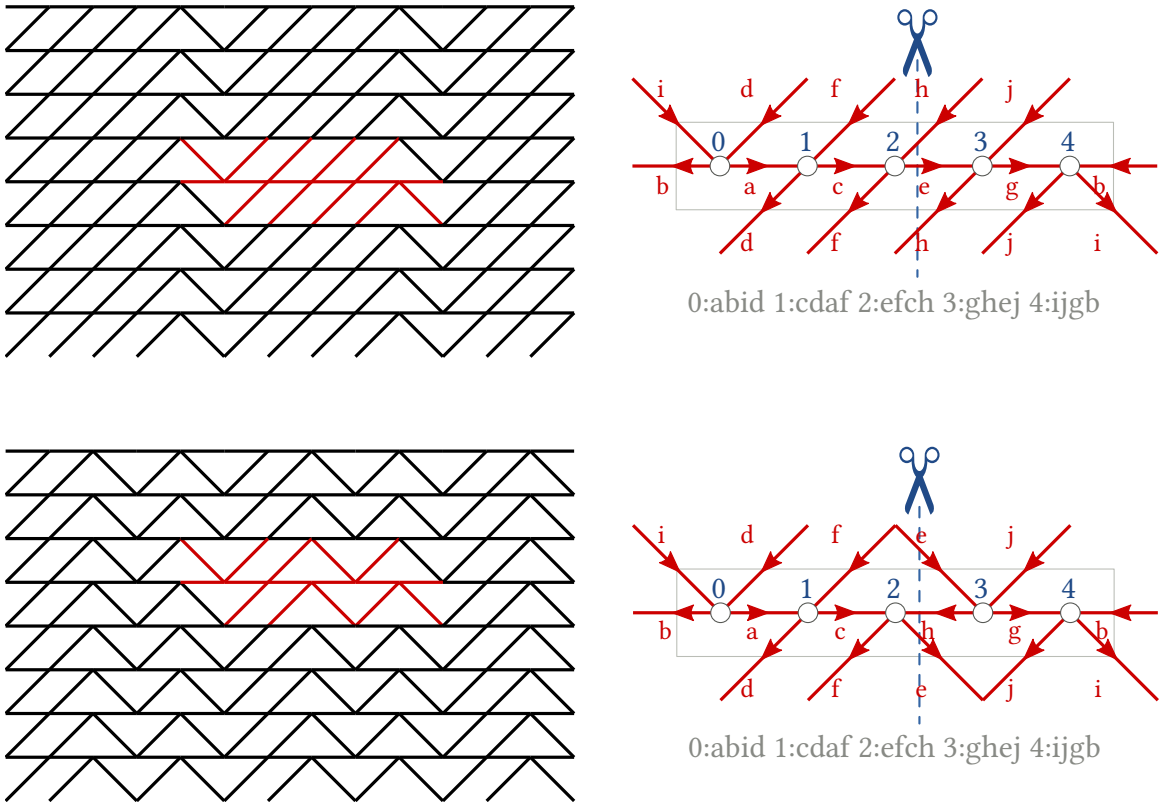


Figure 3.9: Two workable tesselace embeddings that differ by a a Dehn twist along the meridian as indicated by the dashed blue line. Both have the same combinatorial embedding

As discussed in the next chapter, our enumeration and generation algorithm will need to operate on topological embeddings.

Having fully defined the five properties of a tesselace embedding, we can now give a formal definition.

Definition (*Tesselace embedding*). A *tesselace embedding* is a connected, 2-regular digraph embedded on the torus. The underlying combinatorial embedding of the tesselace embedding has genus one and is free from contractible directed cycles. Its topological embedding can be partitioned into a set of osculating circuits, each homotopic to a closed curve with wrapping index $(1, 0)$.

3.4 Related Work

We conclude this chapter with a discussion of previous work in the areas of graph theory identified in the definition of a tesselace embedding.

2-Regular Digraphs In 1987 Ramanath and Walsh [72] presented a backtracking approach for exhaustively generating 2-regular digraphs and published results for graphs with up to 20 vertices. In 1998, Meringer [60] announced the tool `genreg` which uses an orderly approach to generate undirected regular graphs and published the results for 4-regular graphs with up to 18 vertices. More recently, in 2013, Brinkmann [13] proposed an algorithm for generating simple regular digraphs that starts with bipartite undirected graphs and orients the edges. Brinkmann reported the results for 2-regular digraphs with up to 13 vertices.

Doubly periodic digraphs Doubly periodic structures have been investigated extensively in the field of crystallography [81]. In 1864, Brown was the first to use graph theory to represent a chemical compound. Pasteur launched the field of stereochemistry by using graph embeddings to describe the arrangements of atoms in space. In 1977, Wells described the infinite periodic graph of a crystal and its representation by a fundamental finite graph.

In the area of textile research, Grishanov et al. [31] identified the doubly periodic structure as a property of all textiles in their 2011 study of the topology of textiles and discussed the representation of textiles as knots on the torus.

Doubly periodic graphs, also known as dynamic graphs, have been studied since the 1980's where they became of interest in relation to the design of VLSI (Very Large Scale Integration) circuits [42].

Conservation in graphs The concept of a conserved quantity represented in a periodic digraph is fairly novel. There is no analogous concept in chemistry where the crystals are made of discrete atoms, not continuous objects. In most cases of textile design, the concept of conservation is trivial. In techniques such as knitting and crochet, the fabric is made from only one or two strands. In weaving, the strands are partitioned into a small number of sets of parallel strands, usually just 2 sets consisting of the perpendicular warp and weft threads but sometimes 3 sets as in triaxial weave or 3D weave. In graph theory, the concept of conservation is somewhat similar to the idea of conservation of flow in a network.

Graph theory has long been a valuable tool in knot theory and braid theory. In 1877, Tait represented knots as planar graphs and studied knot isomorphism through medial graphs. In braid theory, conservation is a fundamental property, however, periodicity is explored in one dimension only; the cyclic permutation of the order of strands in braid crossings. One dimensional periodic braids are used in the representation of flow dynamics [12] as well as in the representation of juggling patterns [55].

Notes

1. Lace grounds which use techniques such as sewings, sideways twists (e.g. Lille worked sideways [17, p. 21] or QuerFlechter [8, pp. 183-184]), lazy crossings or layering are not covered by this model.

2. An interaction always begins with a cross action. The definition of an interaction should not be confused with the 'closed' versus 'open' methods used by lacemakers. In the 'closed' method, typically used on flat, cookie pillows, a lacemaker picks up four bobbins and always starts to braid with a cross (e.g., a "half-stitch" is CT). In the 'open' method, used primarily on bolster style pillows, a lacemaker usually starts to braid with a twist (e.g., a "half-stitch" is TC) but not always (e.g., "cloth stitch" in the open method is CTC). Both methods result in the same appearance of the final lace; the twist action can often be performed either at the end of one group of actions or the beginning of another without changing the end result. Our definition of an interaction aligns more closely with the 'closed' method because it has a consistent and easily defined boundary.

3. The term tesselace was proposed by lacemaker Lenka Suchanek in private discussion with the author.

4. One exception to the conservation of threads is the gimp thread which is a decoration and does not typically contribute to the structure of the lace. Even so, adding and removing gimp threads at each repeat is quite tedious and to be avoided.

5. Not all bobbin lace styles are made as a continuous piece. Some lace styles, known as part lace or sectional lace, are made by joining smaller motifs together either with a net background or plaits. However, grounds used in part laces also have a perpendicular orientation.

Chapter 4

Enumeration Criteria

Before launching into counting and generating patterns, we must first be specific about what it is that we wish to count and how we will determine whether two patterns are the same. In this chapter, we will define the criteria used for making this determination. We have chosen to cast our net quite wide, identifying factors that will result in a noticeable difference in the finished lace.

Because a tesselace embedding represents a tiling of the infinite plane, it is important to identify the smallest unit that can represent the pattern and avoid counting multiples of this unit. To this end, we will introduce the idea of a prime tesselace embedding and an algorithm for determining if a pattern is prime.

Finally, we will look at the size of the solution space for prime tesselace embeddings and demonstrate that it is infinite.

4.1 Equivalence

When counting unique tesselace embeddings we want to be able to spot duplicates and therefore need to be able to compare two patterns and determine whether they are equal in our context. There are, however, different degrees to which two patterns can be considered the same as illustrated in Figure 4.1. Definitions for the terms used in Figure 4.1 will follow.

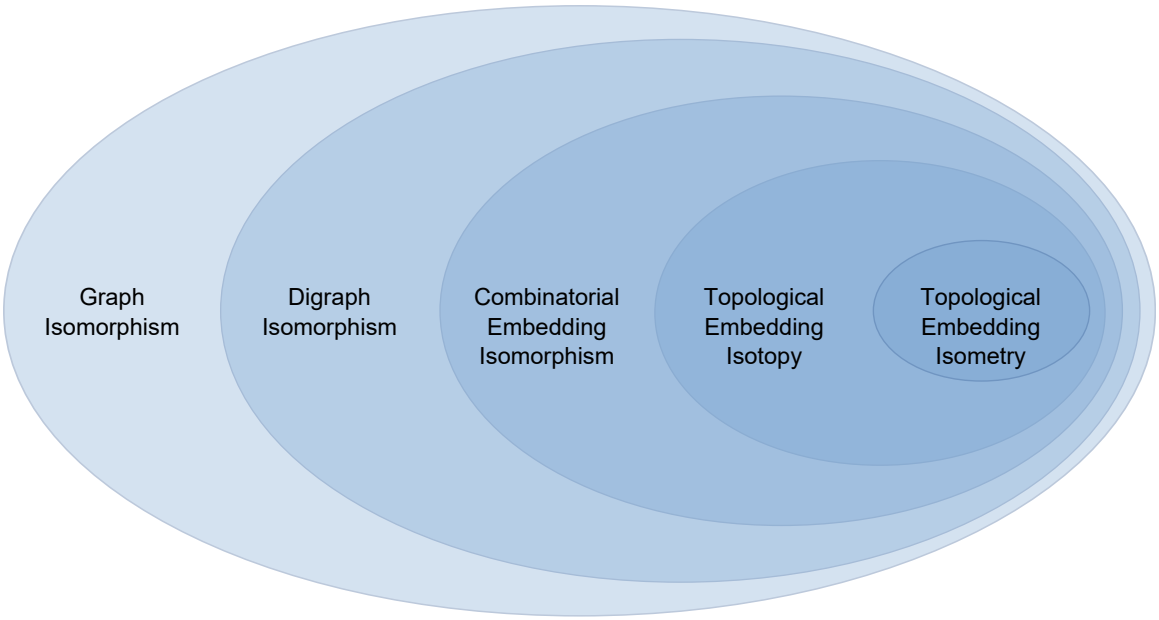


Figure 4.1: Different degrees of similarity for tesselace embeddings

In this section we define an equivalence relation for tesselace embeddings that will be applied to the enumerations reported in Chapters 5 and 7.

Definition (*Equivalence relation*). A binary relation R is an equivalence relation if R is reflexive, transitive and symmetric. R partitions a set into classes of equivalent objects.

At the broadest level, we can think of two patterns as being similar if they have the same underlying graph using the equivalence relation of graph isomorphism. Here we can choose to compare patterns at the level of the digraph of a tesselace embedding or its associated undirected graph.

Definition (*Graph isomorphism*). An *isomorphism between two graphs* G and H is a bijective mapping ϕ from the vertices of G to the vertices of H which preserves both edges and non-edges. That is, there is an edge $\{u, v\}$ in G if and only if there is an edge $\{\phi(u), \phi(v)\}$ in H where $u, v \in V(G)$ and $\phi(u), \phi(v) \in V(H)$. [15]

Isomorphism between two digraphs requires the additional condition that the order of vertices for a directed edge (u, v) in G must correspond to the order in which the mapped vertices appear in the corresponding edge $(\phi(u), \phi(v))$ in H.

We can strengthen our equivalence relation by comparing the embedding of two graphs at the level of incident vertices, edges and faces. The faces of a graph embedding are uniquely described by a rotation system of the graph [62]. We can therefore define an equivalence relation by comparing the rotation systems of two combinatorial embeddings.

Definition (*Combinatorial embedding isomorphism*). Given two graphs G and H that are isomorphic under the bijective mapping ϕ, an *isomorphism between two combinatorial embeddings* $\Pi(G) = (V(G), E(G), \pi(G))$ and $\Pi(H) = (V(H), E(H), \pi(H))$ exists if and only if $e_{i+1} = \pi_v(e_i)$ implies $\phi(e_{i+1}) = \pi_{\phi(v)}(\phi(e_i))$ where $v \in G(V)$ and $e_i, e_{i+1} \in E(G)$. In other words, the mapping ϕ preserves the cyclic order of edges around corresponding vertices. When considering the isomorphism of two combinatorially embedded digraphs, the mapping ϕ must also preserve the direction of the edges. [43]

A key property of a tesselace embedding is that it is thread conserving, a property determined from the wrapping index of an osculating circuit. Recall from Section 3.3 that a combinatorial embedding is unaffected by Dehn twists and does not provide enough information to determine the wrapping index of a circuit. A topological embedding, however, does provide sufficient information. Our equivalence relation should therefore also look for isotopy between two topological embeddings.

Definition (*Topological embedding isotopy*). Two topological embeddings G_1 and G_2 of the same abstract graph G on a surface S are isotopic if there exists a continuous family of embeddings of G (possibly moving the vertices of G) connecting G_1 with G_2. [19]

Isotopy of the topological embeddings encompasses all of properties of a tesselace embedding identified in Section 3.2. However, we will take the definition of equivalence one step further and look at distance and angles, leading to a definition of equivalence based on the isometry of graph drawings.

Definition (*Isometry*). An isometry or congruence transformation is any mapping of the Euclidean plane $\mathbb{R}^2$ onto itself which preserves all distances. Here the *distance* between two points (x_1, y_1) and (x_2, y_2) is the Euclidean distance $d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. [32, p. 26]

The geometry of a graph drawing has significant impact on the appearance of the resulting lace. The length of the threads and the angles at which they cross and bend are determined by the placement of pins in the pattern, a placement which is determined by the position of vertices in the drawing. In [8], Wanzenried discusses the creation of new lace grounds by modifying the position of pins in traditional patterns. For example, consider Figure 4.2 which shows two drawings of the same graph embedding. Notice how changing vertex position affects the shape of the holes (holes being the primary feature that distinguishes lace from cloth) and whether a local section of a path continues in a straight line or bends.

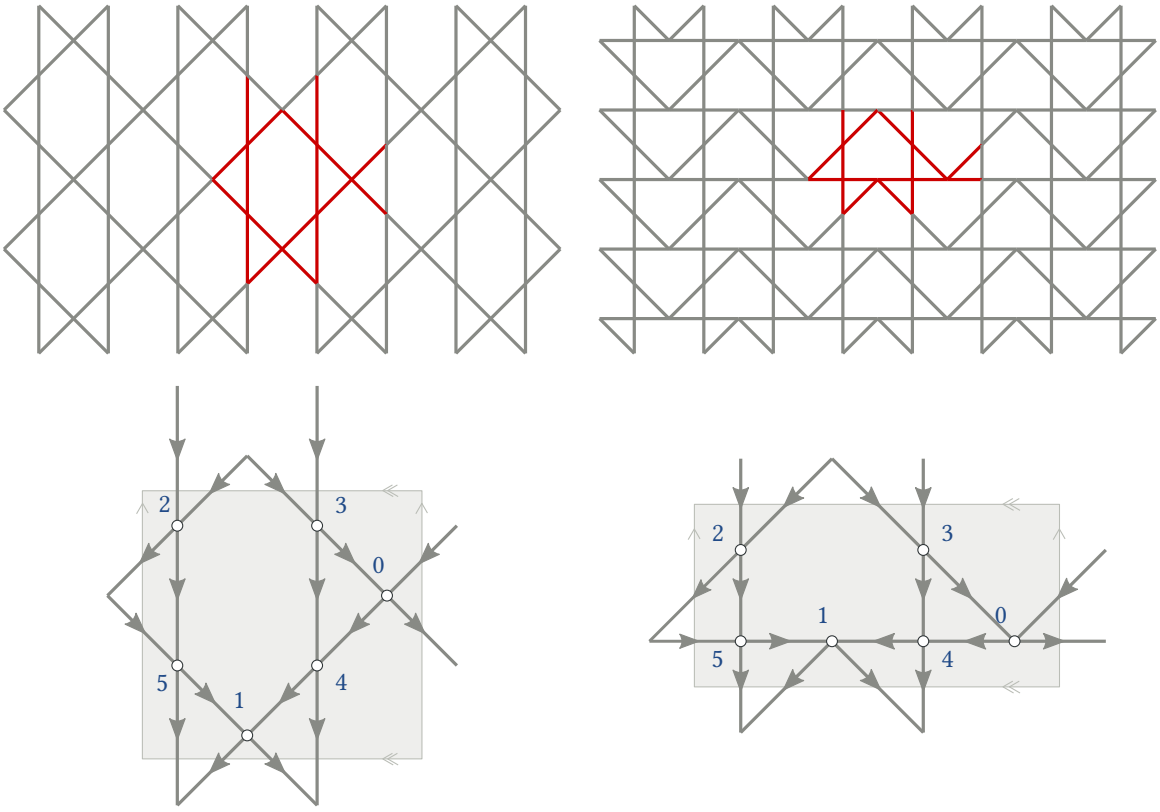


Figure 4.2: Two lace tessellation patterns that are isotopic but not isometric

Isometry of two tesselace embeddings implies isotopy of their topological graph embeddings, isomorphism of their combinatorial embeddings and isomorphism of the associated digraph and undirected graphs.

In addition to the position of edges and their endvertices, the directed edges of a tesselace embedding have an orientation which must be considered. We shall define an *orientation preserving* symmetry as an isometry that maps an edge onto an edge of the same orientation. This is a fairly restrictive type of equality so we introduce a transformation called orientation flipping:

Definition (*Orientation flip*). An *orientation flip* changes the direction of all edges in a digraph. The resulting digraph is called the converse digraph.

Orientation flipping is analogous to the permutation of colours discussed by Grünbaum and Shepherd for coloured patterns and tiles [32, Ch. 8] and is isomorphic with the cyclic group C_2. Like C_2, orientation flipping does not alter the isotopy of two topological embeddings or affect any of the inherited equivalence relations implied by their isotopy. Following the model of Grünbaum and Shepherd, we can therefore define an orientation isometry as follows:

Definition (*Orientation isometry*). An *orientation isometry* is a mapping between two tesselace embeddings, A and B, described by the pair (s, θ) where s is an isometry that maps the unoriented edges of A onto B and θ is an element from the orientation flip group that aligns the direction of the edges in A with the direction of the edges in B.

We note here that there exist pairs of patterns in which the undirected edges of one can be mapped to the other by an isometric transformation and for which each pattern in the pair possess a set of edge orientations that will produce a workable pattern. However, mapping from the edge orientation in one pattern in the pair to the other can not be achieved by an orientation flip. That is, only a subset of edges change orientation. An example is shown Figure 4.3. It is not trivial to determine which edges must change direction. Further, the non-unilateral change in orientation affects the isotopy of topological embeddings. As a result, we will consider such embeddings as different tesselace embeddings.

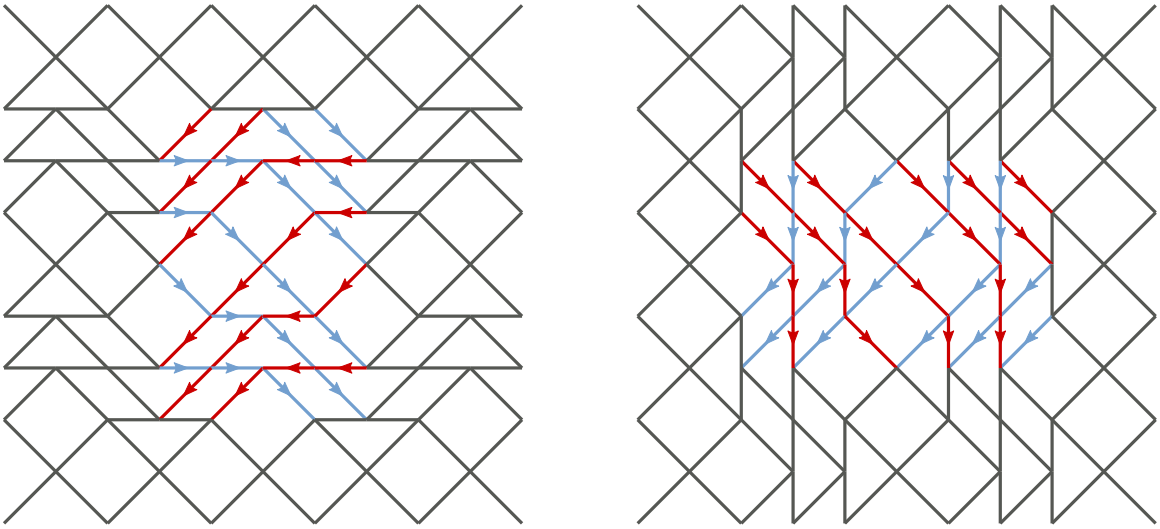


Figure 4.3: Directed graphs with significantly different edge directions but the same geometry and underlying undirected graph under a $90°$ rotation. Edges that have the same direction in both embeddings are blue, edges that are different are red. Note: The symmetry of undirected tesselace embedding does not include a $90°$ rotation

We can now give a full description of the equivalence relation used in the following chapters.

Definition (*Tesselace embedding isomorphism*). Two tesselace embeddings A and B are isomorphic if there exists an orientation isometry that maps A onto B.

4.2 Canonical Label

A canonical label is unique for each isomorphism class of a type of object based on the equivalence relation defined for that object type.

Definition (*Canonical label*). Let S be a set and $\sim$ an equivalence relation on S. A canonical labelling of S is a function $f : S \to L$, where L is a set of labels such that $f(S_1) = f(S_2) \Leftrightarrow S_1 \sim S_2$.

A common practice for determining if two objects are isomorphic is to construct a canonical label for each object [59]. If both objects have the same canonical label, then they are isomorphic. The canonical label for a tesselace embedding must encode the position of its vertices as well as the set of edges and their orientation.

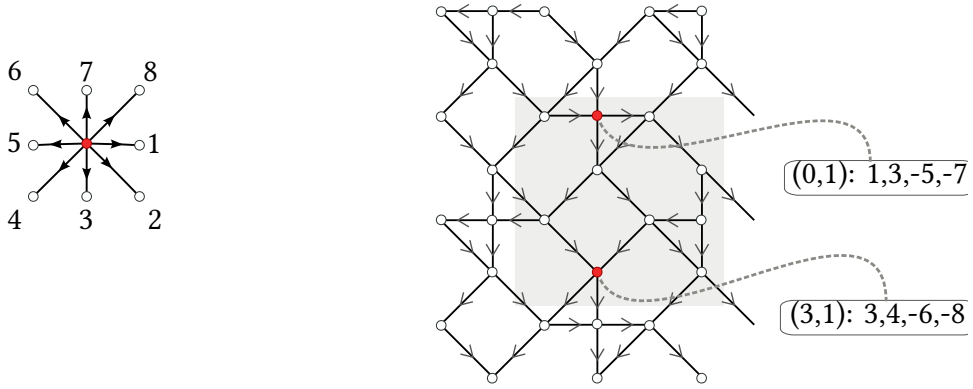
In a completely general approach, the number of possible positions for a graph vertex is infinite even if the graph drawing is confined to a finite region. Clearly enumerating graphs under these conditions is meaningless. In our exploration we make use of the fact that bobbin lace grounds are traditionally designed on a lattice of points.

Definition (*Lattice*). A lattice is the set of images of a point O formed by repeated translation through a fixed distance in two non-parallel directions. The translations can be represented by two vectors a and b. The lattice therefore consists of all points formed by the translations $na + mb$ where $n, m \in \mathbb{Z}$. [32, p. 29]

In bobbin lace the angle between vectors a and b can range from $24°$ to $78°$ [53]. In our enumerations, we shall restrict vertex positions to points on a lattice and examine lattices of a fixed number of rows and columns. This is a more restrictive condition than actually exists in bobbin lace patterns but it is a good place to start and captures the majority of traditional patterns.

For simple graphs, graphs without self-loops and multiedges, an adjacency list in which edges are labelled by their **endvertices** is sufficient to specify the rotation order of edges at a vertex. The graphs associated with tesselace embeddings are multigraphs and therefore the rotation system we use must list the incident **edges** around a vertex. We should note here that a tesselace embedding does not permit contractible cycles, therefore, any loops must be non-contractible. Similarly, any cycle formed from two edges of a multiedge must be non-contractible.

The label for a vertex will consist of a lattice row and column position followed by an ordered list of edges. Edges will be labelled by a vector which encodes the following data: the angle an edge makes with a horizontal line originating at the vertex; the length of the edge; and the orientation of the edge (incoming or outgoing) with respect to the current vertex. Only a finite set $\mathfrak{A}$ of step vectors will be considered. The set $\mathfrak{A}$ will be ordered according to clockwise rotation starting with a horizontal east facing vector. Therefore, instead of specifying the actual magnitude and direction of the step vector, the label will refer to the one based index of the step vector in the ordered set $\mathfrak{A}$. The direction of the edge will be indicated by using a positive (outgoing) or a negative (incoming) value for the set index. Edges in the label will be ordered according to the absolute value of their step vector index. An example is shown in Figure 4.4.



(a) Allowed step vectors *(b) Vertex labels*

Figure 4.4: Left: Labelling scheme for arcs around a vertex. Right: Example solution for 4×4 grid with two vertices labelled. Shaded region indicates the period rectangle of the tesselace embedding.

A *pattern label* is constructed by concatenating vertex labels in row-by-row order. The canonical form of this identifer is the lexicographically least label under isometric transformations of the pattern.

The label for a pattern can constructed time $\mathcal{O}(|V(G)|)$; each edge is reported twice and, because the tesselace embedding is a 2-regular digraph, the number of edges is two times the number vertices. To determine whether a label is in its lexicographically least form, it is compared to all isometric transformations and orientation flips. For a given transformation, this comparison can be made in time $\mathcal{O}(|V(G)|)$, by creating the transformed label and performing a pairwise comparison of sub-elements during construction. Under translation, any of the $|V(G)|$ vertices can appear as the first vertex in the label. The complete set of transformations consists of horizontal and vertical translation composed with the identity, horizontal reflection, vertical reflection (composed with an orientation flip) and $180°$ rotation (composed with an orientation flip) resulting in an overall time complexity of $\mathcal{O}(|V(G)|^2)$. For the example shown in Figure 4.4, there are 10 vertices, therefore, the number of vertex label comparisons required is $4\times10\times10$.

In Chapter 7 we will show that the size of the transformation set we need to consider can be reduced by leveraging the symmetry of the pattern. In some cases, when a fixed point exists, the label can be generated in such a way that no transformations need to be examined.

In concluding this section, we note that there exist several ways for determining a canonical label for the tesselace embedding. The subtask that consumes the most time is determining the order of the vertices so we could look at alternate ways to do this. We could, for example, use a breadth first search (BFS) to find a lexicographically least order. In a BFS, each vertex in the graph embedding is a potential root for the label. Because outgoing arcs are always consecutive in rotational order, the first child of a vertex can be uniquely defined as the far endvertex of the first outgoing edge in clockwise rotational order. The transformations of horizontal reflection, vertical reflection with a flip and $180°$ rotation with a flip also need to be considered. The complexity of a BFS based canonical labelling algorithm is therefore $\mathcal{O}(|V(G)|^2)$ — the same as the array based approach we chose to use.

4.3 Prime Tesselace Embedding

As mentioned in Property 3.2.2, the fundamental embedding is determined by finding a maximal set vertices in $\Delta_0(G^\infty)$ such that no two vertices are equivalent under periodic translation. In this section we will discuss a practical approach for determining whether a topological embedding of a graph in a torus with endvertices restricted to lattice points meets this requirement.

4.3.1 Prime k-ary Toroidal Array

In Section 4.2 we introduced the restriction that vertices must be located on points in a lattice. We can view integer positions of points in a lattice as rows and columns of an array. Because the lattice is on the surface of a torus, the lattice positions wrap around in both the meridional and longitudinal directions which we can equate with cyclic rotation of the rows and columns of an array. The problem of identifying whether a tesselace embedding is prime is therefore analogous to the problem of determining if a k-ary toroidal array is prime.

Definition (*Toroidal array*). A *toroidal array* is a two dimensional array in which opposite edges have been identified allowing for simultaneous rotation of rows and simultaneous rotation of columns.

Definition (*k-ary toroidal array*). A *k-ary toroidal array* is a toroidal array with entries from a set of k symbols.

For example, there are 7 binary 2×2 toroidal arrays; of these, exactly 3 are prime.

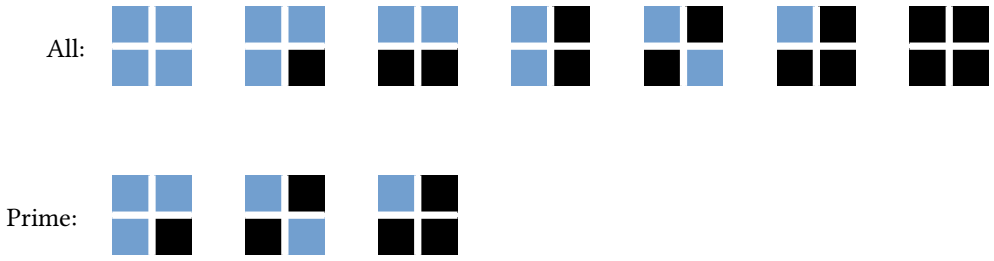


Figure 4.5: The seven binary 2×2 toroidal arrays and their prime subset

4.3.2 Prime Algorithm

In this section we will describe an algorithm to determine whether a given k-ary toroidal array is prime.

We will start be considering the one dimensional variation of this problem.

Definition (*Necklace*). A necklace is the lexicographically least representative of an equivalence class of strings under rotation. Two strings α and β are equivalent if and only if there exist non-empty strings u and v such that $\alpha = uv$ and $\beta = vu$. [73, Ch. 7]

The equivalence relation used in the definition of a necklace can be thought of as the rotation of beads around a string. Like a physical necklace of beads which you can rotate around your neck but cannot reflect without taking the necklace off, elements in string representation also cannot be reflected.

Definition (*Lyndon word*). A string α is periodic if $\alpha = \beta^q$ where β is a non-empty string and $q > 1$. It is aperiodic if $q = 1$. If β is aperiodic, then it is called the *periodic reduction* of α. A Lyndon word is an aperiodic necklace. [73, Ch. 7]

An efficient algorithm for Lyndon word factorization, which can be used for determining the periodic reduction of a string, was proposed by Duval in 1983. Ruskey [73] provides a nice analysis of the running time of this algorithm, which is $\mathcal{O}(n)$ for a string of length n. To find the prime necklace of a string α, Lyndon word factorization is performed on two concatenated copies of α. The longest Lyndon word w resulting from the factorization of $\alpha\alpha$ is the prime necklace of α. If the length of w is less than the length of α, we can conclude that α is not prime.

A toroidal array can be thought of as an extension of the one dimensional necklace in two dimensions.

Definition (*Prime k-ary toroidal array*). A k-ary toroidal array $A = \{\alpha_{i,j}, 0 \leq i \leq n, 0 \leq j \leq m\}$ is periodic if there exists a non-empty array $B = \{\beta_{i,j}, 0 \leq i \leq k, 0 \leq j \leq \ell\}$ such that

$$\alpha_{i,0}...\alpha_{i,m} = (\beta_{i \pmod k,0}...\beta_{i \pmod k,\ell})^q \text{ for } 0 \leq i \leq n \text{ and } q > 1 \textit{ or}$$
$$\alpha_{0,j}...\alpha_{n,j} = (\beta_{0,j \pmod \ell}...\beta_{k,j \pmod \ell})^r \text{ for } 0 \leq j \leq m \text{ and } r > 1.$$

If A is aperiodic, then it is a prime k-ary toroidal array.

No factorization algorithm has been reported for toroidal arrays. Amir and Benson [1] provide an algorithm in the field of string matching for finding periodicity in rectangular arrays, however, the rectangular arrays do not have cyclic symmetry and the concept of periodicity in this case refers to finding regions of partial self-overlap under translation. More recently, Marcus and Sokol [57] described an algorithm for two dimensional pattern matching with horizontal rotation of entire columns which they refer to as horizontal 2D conjugacy. Again their algorithm is focused on the problem of finding partial self-overlap and requires calculating the period of each row.

For our purposes, we propose a new algorithm based on the following theorems.

Lemma 4.3.1. *Let A be a k-ary toroidal array of size $n \times m$. Concatenate the columns of A into a string S of length nm. Let N represent the aperiodic necklace of S, that is let $S = N^a$ where $a > 0$.*

The period p of A in the horizontal direction is given by:

$$p = \frac{\text{lcm}(|N|, n)}{n}. \tag{4.1}$$

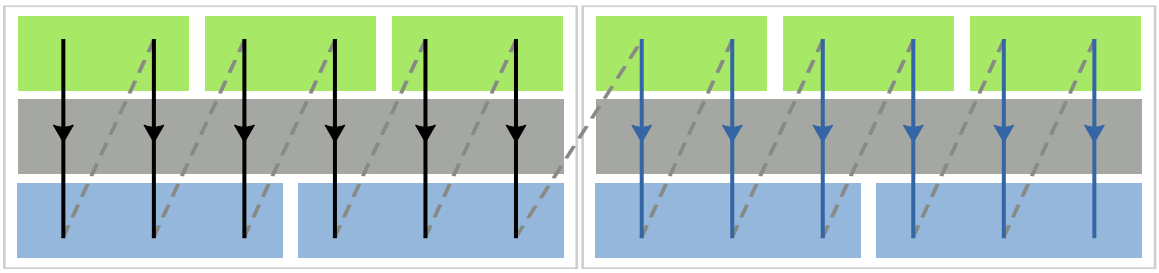


Figure 4.6: A visual representation of Theorem 4.3.1 applied to an example toroidal array. Coloured blocks correspond to the periodic reduction of rows in the array.

Proof. Let p_i be the period of row i in A. The set of all row periods is $\{p_1, p_2, ..., p_m\}$. Cells in row i will repeat at column indices that are a multiple of p_i. All cells in a column will therefore repeat at a distance that is the least common multiple of all p_i for $1 \leq i \leq m$. The horizontal period p of the array is therefore the least common multiple of $\{p_1, p_2, ..., p_m\}$. Since all cells in a column repeat at a distance p, the columns concatenated into a string will also repeat at a distance np since there are n cells in each column. Therefore, there is substring S' of S of length np which divides evenly into S and which by replication can be used to create S. If S' is the shortest substring of S with this property, then, by definition, S' is the prime necklace of S and the result is proven.

It is also possible that there exists a substring $S'' \subset S$ which is shorter than S', that also divides evenly into S and which, by replication, can be used to create S. We can also state that S'' bears the same relation to S' as S' bears to S. Namely, S'' is a substring of S' that divides evenly into S' and can create S' by replication. We know that the length of S'' is not divisible by n because if it was, the columns of cells would repeat at a distance less than p which would contradict the statement that p is the horizontal period of A. Copies of the two substrings must therefore align at a distance that is a multiple of n given by $lcm(|S''|, n)$. The same argument can be made for smaller substrings until eventually we reach the smallest such substring that is the prime necklace of S. □

Theorem 4.3.2. *The array A is aperiodic in the horizontal direction if and only if the horizontal period of A given in Lemma 4.3.1 is equal to m, the number of columns in A.*

Corollary 4.3.3. *Theorem 4.3.2 also holds true for the vertical period of A.*

The horizontal and vertical cyclic symmetries of the toroidal array are independent. Changing the orientation of the array does not change the algorithm.

Lyndon factorization has time complexity $\mathcal{O}(nm)$ for a string of length $2nm$. The least common multiple can be calculated using Euclid's algorithm for greatest common denominator which has time complexity $\mathcal{O}(log_{10}(min(n, m)))$ [46]. The time complexity of the greatest common denominator algorithm is small compared to that of Lyndon factorization, therefore, applying Theorem 4.3.2 to a k-ary $n \times m$ toroidal array can be performed in $\mathcal{O}(nm)$ time.

For a tesselace embedding, each bead in the 2D necklace corresponds to a vertex in the graph embedding. The number of colours k corresponds to the number of possible edge configurations around a vertex. There are 93 possible bead colours for the tesselace embeddings discussed in Chapter 5.

4.4 Solution Set Size

We conclude this discussion on enumeration criteria with a look at the size of the problem space we wish to explore.

Theorem 4.4.1. *There is an infinite number of prime tesselace embeddings.*

We will show two different proofs for this theorem, each based on modifications to a traditional, well established tesselace embedding.

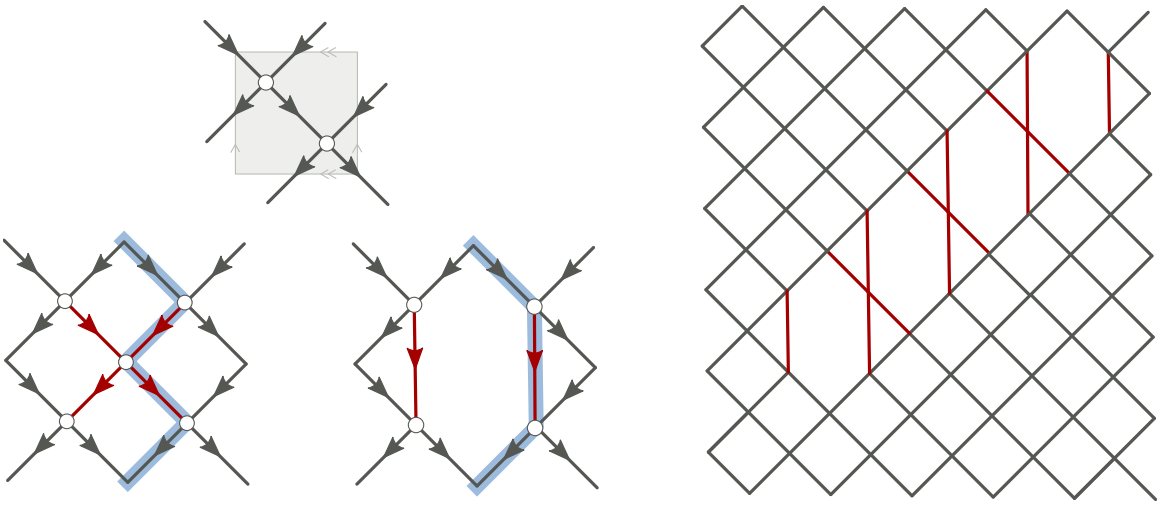


Figure 4.7: Graph modifier μ acting on a diamond lattice graph embedding. Prime tesselace embedding generated by applying μ in accordance with the Lyndon word 00001.

Proof. We will prove that there is an infinite number of prime tesselace embeddings by showing that there is an injection from Lyndon words to prime tesselace embeddings.

We start with the topological graph embedding D shown in the top left of Figure 4.7. It is very simple but is perhaps one of the oldest and most widely used pair traversal patterns appearing in *Torchon*, *Dieppe*, *Valenciennes* and *tulle* grounds to name just a few examples. That this embedding is a tesselace embedding has been verified through 500 years of use by lacemakers and can be quickly confirmed by analysis of the two vertex embedding.

The graph modification μ, illustrated on the bottom left of Figure 4.7, can be applied to this diagonal lattice. Under the application of μ, a vertex at which two osculating paths kiss is removed. The pair of edges from the left side of the kiss is replaced by a single edge such that the left path continues along its original trajectory, minus a kiss. The mirror reflection of this change is made to the right pair of edges. The modification preserves the degree of the remaining vertices, does not change the genus of the graph (a vertex, a face and two edges are removed by this operation leaving the Euler characteristic of the combinatorial embedding unchanged) and does not introduce any cycles. Further, because the paths continue along their original trajectory and continue to osculate at all

remaining vertices, the thread conservation property is preserved. When applied to non-consecutive vertices, the graph remains connected under the modification μ. The resulting graph embedding therefore satisfies all 5 properties of a tesselace embedding and is itself a tesselace embedding.

Consider a lattice grid with n rows and n columns formed from $n \times n$ repeats of the prime graph embedding D. We can apply μ to vertices v_{ii}, where i is an even number, along the diagonal of this diamond lattice. If we choose to modify vertex v_{ii} (create a hole at position (i, i)) we can represent this as 1 otherwise we can skip this vertex and represent this condition by 0. In this manner we can create a binary string to represent the pattern formed by applying μ along the diagonal. Any arbitrary binary string of length $\lfloor n/2 \rfloor$ is permissable. Let us choose a Lyndon word as our binary string. Figure 4.7 shows an example of such a pattern. We know that there exists at least one Lyndon word of length l for all $l \in \mathbb{N}$ and that it is aperiodic. The pattern created by applying μ in accordance with a Lyndon word of length l will therefore be a prime tesselace embedding. We can therefore conclude that an infinite number of prime tesselace embeddings exist. □

We can also observe that the number of Lyndon words increases exponentially as expressed by Moreau's necklace counting formula. Since any Lyndon word is a suitable generator for a prime tesselace embedding, we can therefore also expect the number of prime tesselace embeddings to increase exponentially.

Lemma 4.4.2. *[Moreau's necklace-counting function] The number $P_k(n)$ of binary Lyndon words of length n is as follows:*

$$P_k(n) = \frac{1}{n} \sum_{d \backslash n} \mu(n/d) 2^d \tag{4.2}$$

where μ is the Möbius function

For the second proof we introduce the idea of a uniform tiling.

Definition (*Uniform tiling*). A unform tiling is a vertex-transitive, planar tessellation of regular polygons. In a vertex-transitive tiling, each vertex is surrounded by the same kind and number of polygons in the same rotational order. [32]

Proof. We will prove that there is an infinite number of prime tesselace embeddings by showing that there is an injection from a family of tilings with infinite size to a set of tesselace embeddings.

The following proof is based on Grünbaum and Shepherd's proof of an infinite family of 2-uniform tilings described in [32, Sec. 2.1].

We start with the uniform tiling $3.6.3.6$ in which each vertex is surrounded, in alternating order, by two hexagons and two equilateral triangles as shown on the left hand side of Figure 4.8. This tiling can be modified by translating a horizontal strip of triangles so that the triangles in the strip change from being vertex aligned to being edge aligned with the triangles above and below the strip. In this way, a strip can be in one of two distinct positions: edge aligned or vertex aligned. It is possible to perform this modification on any number of strips in the pattern. More specifically, if the pattern consists of n strips of triangles, there are $\sum_{i=0}^{n} \binom{n}{i} = 2^n$ ways to arrange the strips (some of the configurations will be equivalent due to symmetry but their number is very small compared to the number of non-isomorphic patterns as n approaches infinity and can therefore be ignored). Since the summation gives positive answer for all $n \in \mathbb{N}$ we can conclude that the size of this tiling family is infinite.

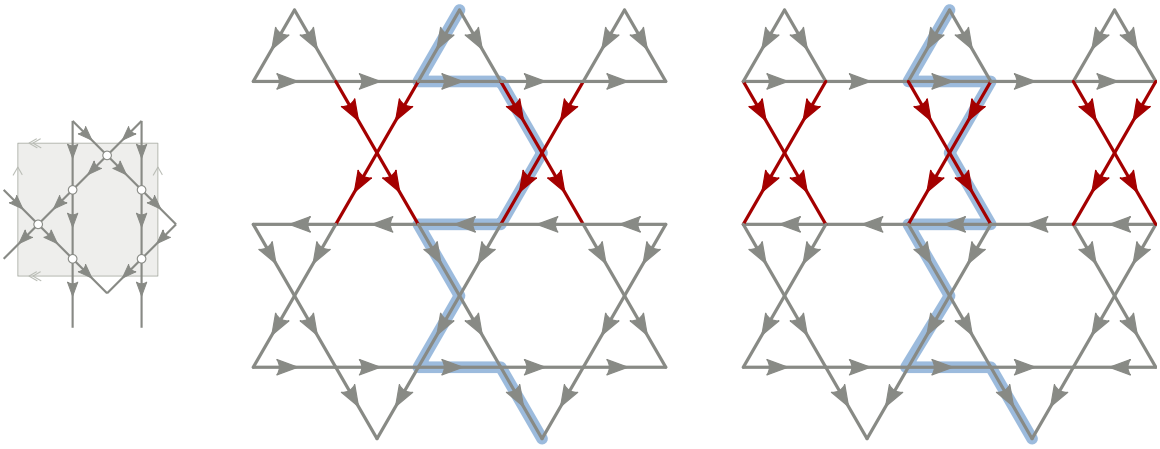


Figure 4.8: Modification of digraph by exchanging partners. Based on Figure 2.1.2 of [32]

It remains to show that all tilings in this family meet the criteria of a tesselace embedding.

The 3.6.3.6 tiling corresponds to a well know lace ground which goes by the names *Point de Paris*, *Kat stitch*, *fond chant* as well as several others. The five properties of a tesselace embedding can be tested on the graph ground embedding shown on the left side of Figure 4.8 and found to hold. Similar to the argument used in the first proof, we see that sliding a horizontal strip of triangles has the effect of separating two osculating paths at a kissing vertex. In this case, instead of the kiss disappearing, the paths turn away from their original partner and find a new partner in the opposite direction to kiss. Because of the rotational symmetry of the torus, it will always be true that both members of a kissing pair will be able to find a new partner to embrace. This little act of infidelity has not changed the genus, the connectedness or the degree of the vertices. It is also clear that leading up to and following on from the graph modification all osculating paths follow their original trajectory and the modification does not affect the wrapping number. We can therefore conclude that the resulting graph embedding is also a tesselace embedding. □

Chapter 5

Generation via Lattice Paths

From Lemma 3.3.4, we see that a tesselace embedding can be partitioned into a set of circuits that do not intersect transversely. We have also observed that lace grounds are traditionally designed on a lattice grid. It seems logical to marry these two ideas together and generate tesselace embeddings by joining lattice paths together in an osculating manner.

Definition (*Lattice path*). A *lattice path* is a sequence of connected line segments that travel between lattice points using a finite set of allowed step vectors.

Alternatively, a lattice path is a path such that vertex positions are constrained to be points on a lattice and the end vertex of an edge in the path must be an image of the start vertex under a fixed set of translations.

Motzkin paths are a well known family of lattice paths. They have the allowed steps $\nearrow$, $\rightarrow$ and $\searrow$ which can be expressed more precisely as the set of step vectors $\mathfrak{M} = \{\langle 1, 1\rangle, \langle 1, 0\rangle, \langle 1, -1\rangle\}$ in which $\langle x, y\rangle$ indicates a step of x units in the horizontal direction and y units in the vertical direction [22].

Motzkin paths are typically drawn travelling from left to right, therefore, for the next two chapters we will rotate our tesselace embeddings 90° counter-clockwise to align with this convention.

Following an analysis of traditional bobbin lace grounds, a representative set of step vectors was chosen. The set does not exhaustively include all transitions found in traditional lace grounds, however, it covers the majority of commonly used patterns and provides a good start point for analysis. A more detailed discussion of the pattern elements not covered is given in Section 8.1. The step vectors, illustrated in Figure 5.1, form the following set:

$$\mathfrak{L} = \{\langle 1,1\rangle, \langle 1,0\rangle, \langle 2,0\rangle, \langle 1,-1\rangle, \langle 0,1\rangle, \langle 0,2\rangle, \langle 0,-1\rangle, \langle 0,-2\rangle\}.$$

The paths produced from these step vectors, which we will refer to as *lace paths*, start at position (x, y), end at position $(x+n, y)$ and may travel above and/or below the horizontal line connecting these two points.

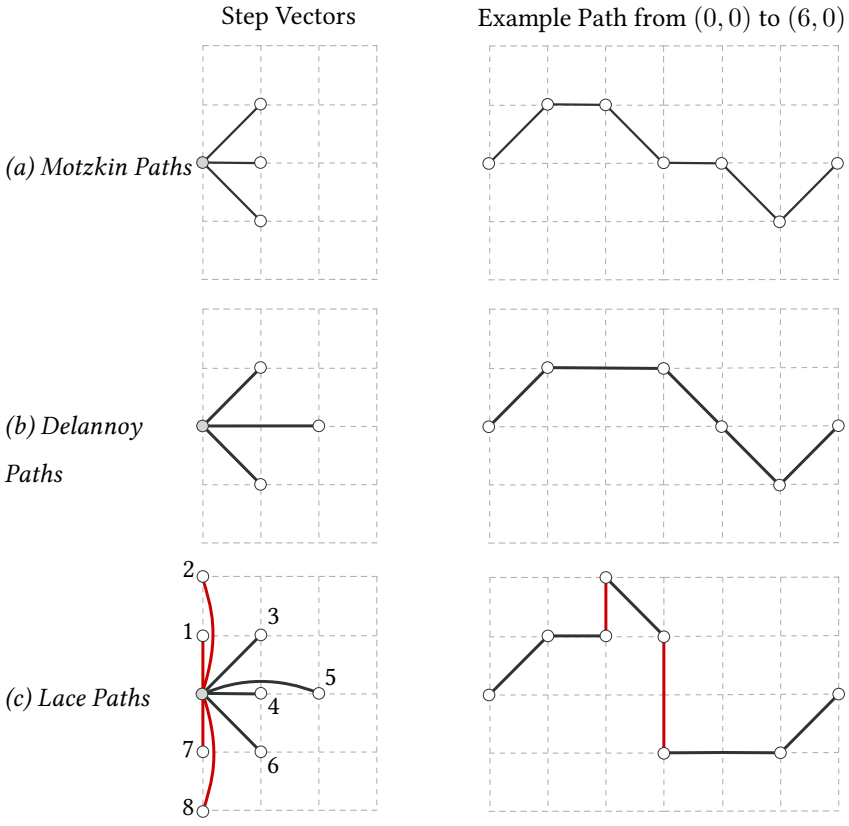


Figure 5.1: Motzkin, Delannoy and lace path step vectors

One notable difference between lace paths and Motzkin paths is the addition of vertical steps. We recall that lace paths will be joined together in an osculating manner. It is therefore important to note that a path formed from $\mathfrak{L}$ with two or more consecutive vertical steps, can only combine with any other path formed from $\mathfrak{L}$ via a transverse intersection. As a consequence, a lace path can not have consecutive vertical steps. That is, $\langle 0, y\rangle$ can not be followed by $\langle 0, y'\rangle$, for $y, y' \in \{-2, -1, 1, 2\}$.

Another consideration for lace paths is that they will be wrapped around to form a cycle. We must therefore be careful to avoid paths that both start and end with a vertical edge. This is accomplished by not allowing lace paths to begin with a vertical step. That is, the first step of a lace path must come from the set $\mathfrak{L}' = \{\langle 1, 1\rangle, \langle 1, 0\rangle, \langle 2, 0\rangle, \langle 1, -1\rangle\} \subset \mathfrak{L}$. Restricting the first step does not omit any valid cycles from our consideration because it is quite arbitrary whether a vertical step appear at the beginning or end of its associated path.

Wrapping around the torus introduces a second interesting requirement. The step vector $\langle 2, 0\rangle$ spans two columns. In order to create a complete set of paths that can be combined in an osculating manner, we must allow any step with a horizontal length of 2 columns to straddle column 0. This is modelled by having two copies of each path that starts with $\langle 2, 0\rangle$ — one that starts at column 0 and one that starts at column -1.

Lace paths are examined in more detail in Chapter 6. In the rest of this chapter we will look at the backtracking algorithm used to combine paths and the output of the algorithm including pattern enumerations and examples of the generated diagrams.

5.1 Backtracking Algorithm

The topological graph embeddings resulting from an osculating joining of lattice paths are guaranteed to be connected, free from contractible cycles, thread conserving and have an associated combinatorial embedding with genus of one. The only real question is how to efficiently generate such unions. Backtracking is often used to enumerate combinatorial objects which are not simple and/or not very uniform. For a detailed discussion of this algorithmic approach we refer the reader to Knuth [45] and Ruskey [73].

Definition (*Backtracking*). Backtracking is an algorithmic approach for generating all sequences $x_1...x_n$ that satisfy some property $P_n(x_1, ..., x_n)$ where x_k belongs to some domain D_k. Further, there exists some intermediate cutoff property $P_l(x_1, ..., x_l)$ for $0 \leq l < n$ such that

$$P_l(x_1, ..., x_l) \text{ is true whenever } P_{l+1}(x_1, ..., x_{l+1}) \text{ is true;}$$
$$P_l(x_1, ..., x_l) \text{ is fairly easy to test, if } P_{l-1}(x_1, ..., x_{l-1}) \text{ holds.}$$

[45, Sec. 7.2.2]

In the current lattice path based generation of tesselace embeddings, the solution set is a representation of $\Delta_1(G^\tau)$ given by an $n \times m$ toroidal array in which each array element is the label of a vertex at lattice position (i, j). The array corresponds to a $n \times m$ lattice grid representing a fundamental rectangle of the ground. Each x_k in D_k is a lace path along with the lattice position of its starting point. The property $P_l(x_1, ...x_l)$ is defined by the following set of rules:

Rules 5.1.1.

1. *No two paths intersect transversely,*
2. *No two paths share a common edge,*
3. *For $l > 1$, x_l has at least one vertex in common with x_{l-1},*
4. *For $l > 2$, there are no vertices in x_{l-1} with out-degree 1*
5. *All vertices with out-degree 2 have a label that is lexicographically greater than the smallest possible label of vertex $(0, 0)$ (if the vertex at $(0, 0)$ has out-degree 2, the smallest possible label is the vertex label, otherwise it is determined by the edges already assigned)*

Lace paths are added to $\Delta_1(G^\tau)$ from bottom to top. As each edge in the path is added, the labels of the associated endvertices in $\Delta_1(G^\tau)$ are updated.

The backtracking process can be described as building a tree in which internal nodes represent incremental partial solutions and leaves represent complete solutions. Each child node at level k of the tree extends the partial solution of its parent by adding a element from the domain set A_k. Failure of a candidate node to satisfy Rules 5.1.1 results in the termination of its associated branch. When a branch terminates at level k, either by finding a complete solution or by early termination, the algorithm walks back up the

tree, returning to the unextended version of the parent node, and attempts to append the next choice of x in A_k. This process continues until all values in A_k have been explored. Backtracking can offer significant performance gains on a brute force search if the size of the tree is kept as small as possible. The number and length of the branches can be minimized through the choice of good early termination conditions and a small set of configurations for each branching point.

For our scenario, significant gains can be made by identifying pairs of lace paths that can be combined sequentially. As a preprocessing step, all possible ordered pairs (A, B) of lace paths in L_n are examined. The distance d_{AB} between $(x_A, 0)$ and $(x_B, 0)$ is calculated where d_{AB} is the distance such that A and B do not intersect transversely but still have at least one common vertex and no common edges. The value d_{AB} is stored along with the path identifiers. If such a distance does not exist or if it is greater than the number of columns in the pattern grid, the pair is discarded. This preprocessing reduces the number of failures due to rules 1,2 and 3 of Rules 5.1.1. Because of wrapping in the vertical direction around the torus, it is still necessary to test rules 1 and 2 at each stage, however, large portions of the search space have been removed by being preemptive.

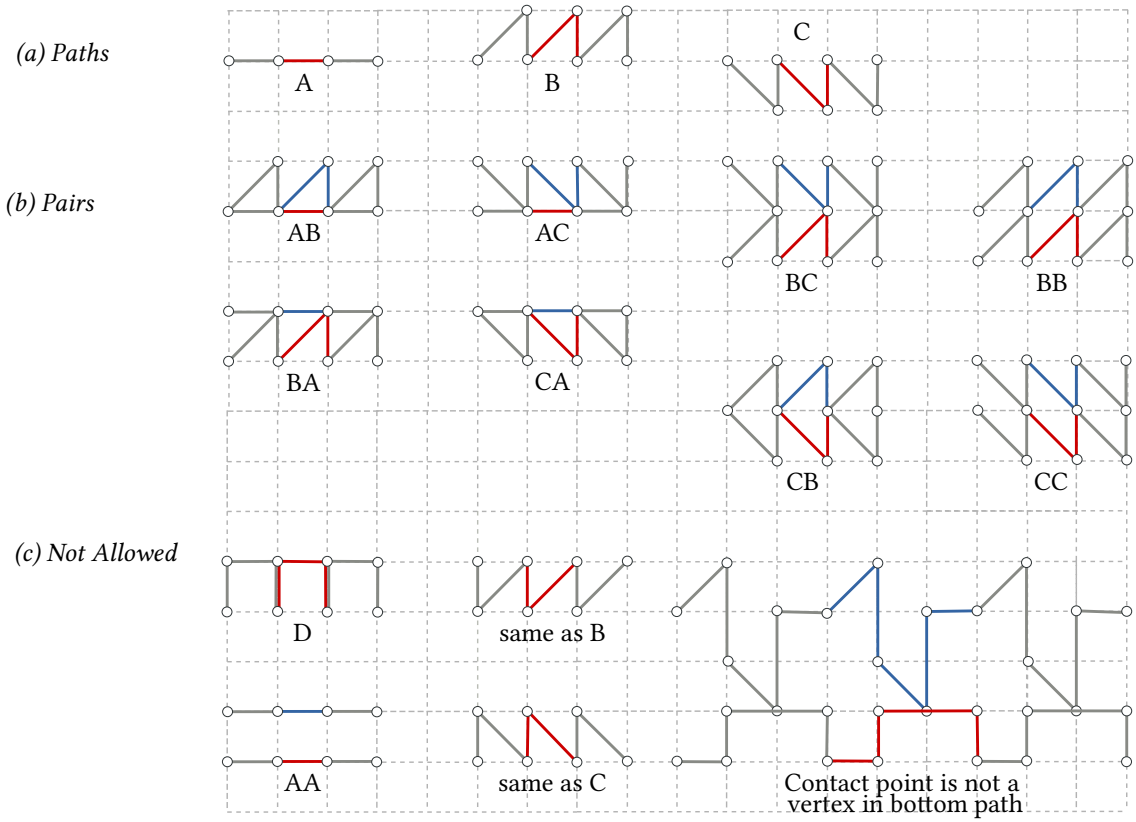


Figure 5.2: Allowed paths and pairs for $n = 1$. Some examples of disallowed paths and pairs

An additional array is used for frequently repeated tests such as the incoming and outgoing degree of a vertex.

When an embedding is successfully completed, it is further tested for isomorphisms and to determine whether it is prime.

5.2 Results

The backtracking algorithm was implemented in Java. Since each branch of the backtracking tree can be evaluated independently, the fixed thread pool service of Java's concurrent Executor class was used to process branches in parallel. The algorithm was executed on a 3.40 GHz machine with eight Intel i-4770 cores, 16 GB RAM and a 64 bit Windows Operating System.

As shown in Table 5.1, the size of L_n, the set of lace paths from $(0,0)$ to $(n,0)$, grows exponentially.

n	$\lvert L_n\rvert$	$\lvert Pairs\rvert$	$\frac{\lvert Pairs\rvert}{\lvert L_n\rvert^2}$	Max	$\frac{Max}{\lvert L\rvert}$	Median	$\frac{Median}{\lvert L\rvert}$
1	3	8	0.89	2	0.666	1	0.333
2	39	1,174	0.77	21	0.538	7	0.179
3	498	196,841	0.79	251	0.504	83	0.167
4	6,667	35,602,650	0.80	3347	0.502	652	0.098

Table 5.1: Statistics for lace paths from $(0,0)$ to $(n,0)$ for small values of n

In preprocessing, $|L_n|^2$ ordered pairs are examined of which approximately 80% are found to meet rules 1,2 and 3 from the Rules 5.1.1. The key result from preprocessing is the reduction in the number of paths that need to be considered at each branching point in the backtracking tree. As indicated by the data shown in Table 5.1, in the worst case, with each node having the maximum number of successful pairings, the number of children per node is halved. More typically internal nodes will have the median number of branches with each node having roughly one tenth the number of children as n approaches 5.

column/rows	1	2	3	4	5
1	1	1	1	2	3
2	3	7	26	112	535
3	4	26	277	3,527	53,132
4	16	176	4,308	$137,273^*$	$5,296,686^\dagger$
5	42	1,090	70,200		

* Backtracking tree contained 7×10^8 nodes and completed in 18 days

† Backtracking tree contained 9×10^9 nodes and completed in 43 days

Table 5.2: Enumeration of prime tesselace embeddings using lattice paths

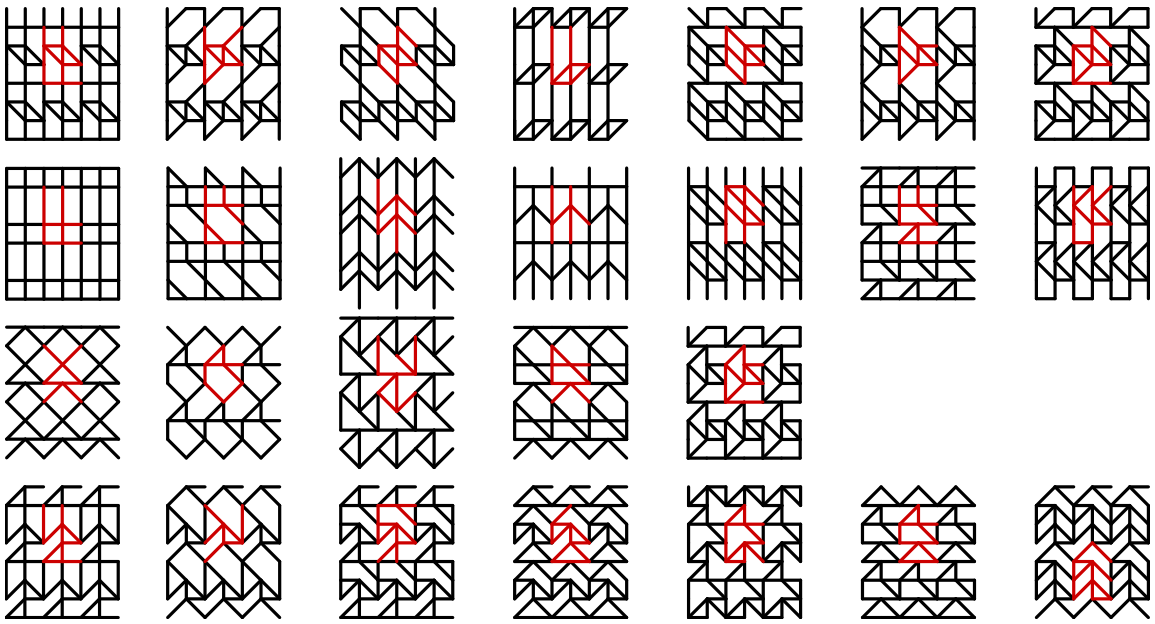


Figure 5.3: Complete results for 3×2. Fundamental graph is shown in red. (Note: patterns are rotated 90° counter-clockwise from direction of lace working)

As anticipated, Table 5.2 shows exponential growth both as the number of columns and as the number of rows in the period rectangle increases. Increasing the number of columns accelerates growth more rapidly than increasing the number of rows. Two factors can be associated with this difference: 1) The number of columns determines the length of the lace paths available as building blocks. 2) Due to the restriction on consecutive vertical edges, the edges in $\Delta_1(G^\tau)$ are biased in the horizontal direction.

The performance of our backtracking approach can be compared to brute force generation. Consider a brute force method in which all possible vertex labels are evaluated. There are $|C|^{nm}$ scenarios to consider where C is the set of distinct vertex labels for the edges allowed by $\mathfrak{L}$. The size of C is $|C| = 1,809$. For $n = 4, m = 5$, a vertex label brute force approach would need to evaluate 1.4×10^{65} potential solutions. Alternatively, consider a brute force method that uses combinations of lace paths. Each row of the pattern can accomodate zero, one or two paths. Therefore, for a period rectangle with n columns and m rows, there are $(1 + |L_n| + |L_n|^2)^m$ combinations to evaluate. For $n = 4, m = 5$, a lace path brute force method must examine 1.7×10^{38} combinations. In contrast, the number of nodes examined in our backtracking tree for 4×5 was 9×10^9 thus demonstrating a very significant performance improvement.

Several samples of the solutions in Table 5.2 have been made into lace using a variety of action sequences and cotton thread. Three of these are shown in Figure 5.4. For each tesselace embedding, three different ζ mappings were used.

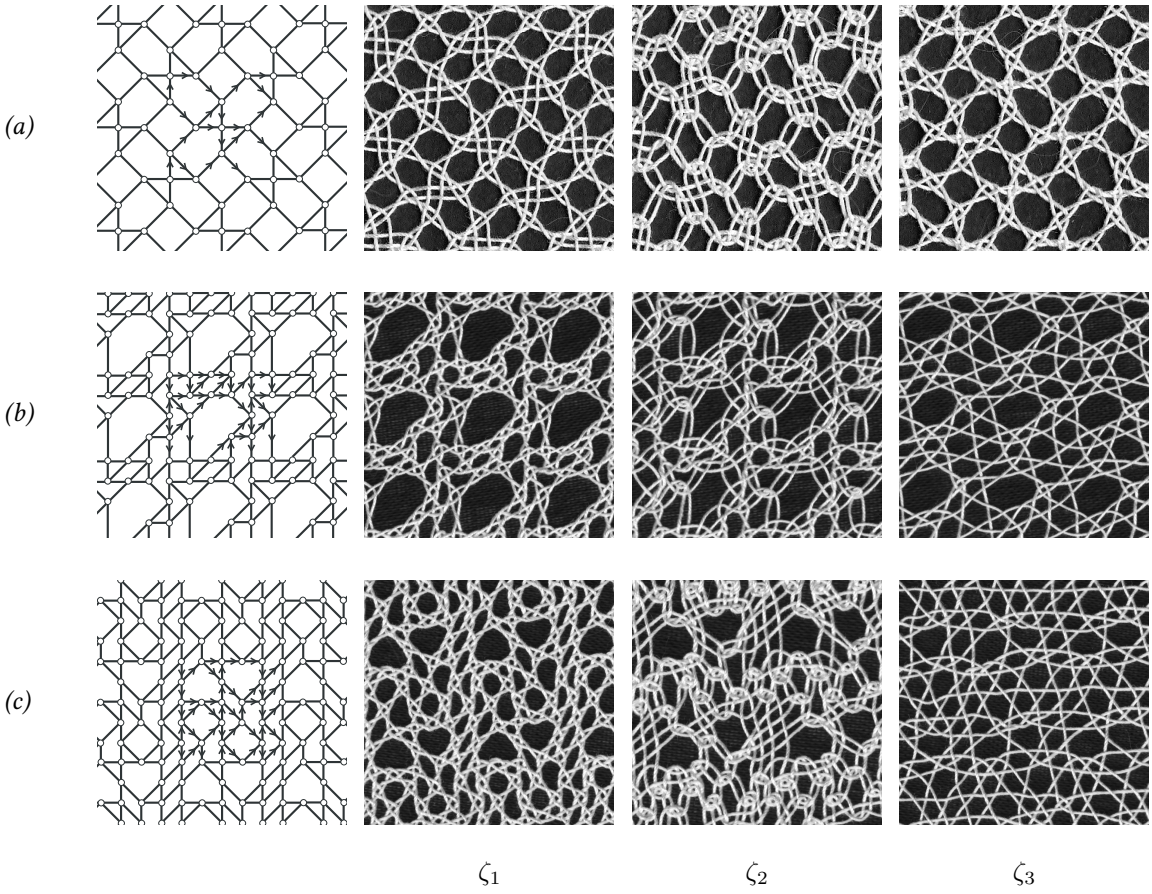


Figure 5.4: Examples of new lace grounds discovered via algorithm worked in thread using three different ζ mappings. As far as the author is aware, these patterns have not been previously documented. The patterns and finished lace are rotated $90°$ counter-clockwise.

The size of the period rectangle we can explore with the lattice path approach is limited by both memory and performance. All lace paths, as well as data about pair spacing, are kept in memory. With exponential growth in the number of lace paths, the size of this data becomes untenable above 5 columns. From a performance perspective, 10^{10} is the limit as far as the number of nodes that can be processed in a reasonable amount of time.

Finally, we note that using isometry as an equivalence relation results in some patterns with trivial differences. For example, Figure 5.5 shows three patterns that are visually distinct but represent variations that are quite easy for the lacemaker to design on the fly. Patterns in this family are related by an operation we shall call accordion scaling. *Accordion scaling* is a scaling transformation that applies to a single pair of rows or a single pair of columns. In the horizontal direction, if all edges between columns x_i and x_{i+1} are of the type $\langle 1, 0\rangle$ (no diagonal edges) and there are no edges of type $\langle 2, 0\rangle$ originating on column x_i or terminating on column $x_i + 1$, then the pattern can be stretched horizontally by transforming all $\langle 1, 0\rangle$ edges originating on column $x_i + 1$ into $\langle 2, 0\rangle$ edges. All edges to the right of x_i are translated right by one column. In a similar manner, two columns exclusively spanned by $\langle 2, 0\rangle$ edges can be horizontally compressed by one column. In the vertical direction, the same operations apply. Fortunately, an examination of the results shows that the members of this family represent a very small subset of the generated patterns.

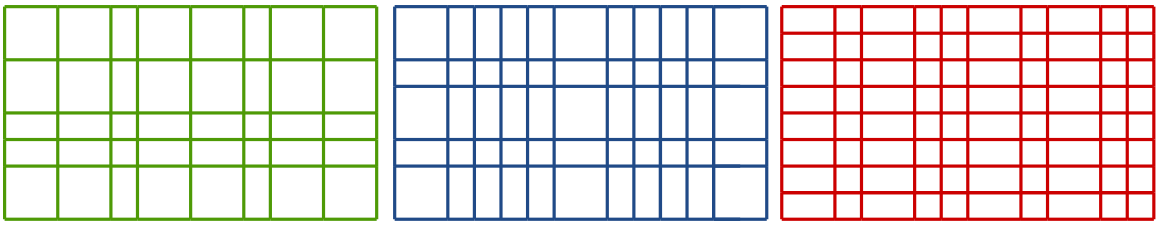


Figure 5.5: Examples of tesselace embeddings related by accordion scaling

Chapter 6

Vertically Constrained Lattice Paths

The lace paths considered in the previous chapter have eight different step vectors, two in each of the vertical and horizontal directions, making their analysis fairly complex. In this chapter, we will take an in depth look at the type of lattice path used into generate lace embeddings but we will simplify things slightly by reducing the number of step vectors to just one in each of the vertical and horizontal directions.

Lattice path enumeration, a classic part of combinatorial enumeration, has been explored for over a century [2]. Research in this area has uncovered some very well known classes of lattice paths such as the Dyck, Motzkin, Schröder and Delannoy paths [22, 78, 6]. Motzkin paths, for example, are lattice paths restricted to the quarter plane. They start and end on the x-axis and are constructed from the finite set of step vectors $\mathfrak{M} = \{\langle 1, 0\rangle, \langle 1, 1\rangle, \langle 1, -1\rangle\}$). The Motzkin triangle, described in 1977 by Donaghey and Shapiro [22], gives a broader view of Motzkin paths by enumerating all quarter plane paths constructed from $\mathfrak{M}$ that terminate a horizontal distance n and vertical distance m from the origin.

Many variations on the lattice paths have been examined in the literature, a few of which are summarised here: applying a weight to each step [29], using step vectors of variable length (e.g., $\{\langle x, 1\rangle, \langle x, -1\rangle, \langle y, 0\rangle\}$, or $\{\langle 1, b\rangle, b \in \mathbb{Z}\}$) [7, 48, 80, 61], applying boundary restrictions such as "elevated" paths that only touch the boundary at start and end vertices [80], and paths restricted to different regions such as the quarter plane, half plane, wedge and slit [11, 86, 69].

As described in Chapter 5, tesselace embeddings can be generated using lattice paths very similar to the Motzkin paths but with the addition of vertical steps. An example of one of these *partially directed* (weakly monotonic and self avoiding) paths is shown in Figure 5.1(c). Partially directed lattice paths have had considerable attention, particularly in the area of polymer modelling.

The addition of north and south step vectors generates, in the absence if any additional constraints, an infinite number of paths between the origin and a termination point. In most investigations, the number of partially directed paths is restricted by considering only paths with a specified number of steps (see for example the "slow walk" example of Niederhausen and Heinrich [66]). Another alternative is to constrain the path to a certain region of space such as a wedge or a slit [86]. The partially directed lattice paths encountered in bobbin lace are not constrained by either of these conditions. Instead, the number of lace paths is bounded by the constraint that vertical steps $(\uparrow, \downarrow)$ cannot be consecutive.

In this chapter we will examine vertically constrained lattice paths on the discrete Cartesian plane $\mathbb{Z}\times\mathbb{Z}$ constructed from the finite set of step vectors $\mathfrak{A} = \{\langle 1,0\rangle, \langle 1,1\rangle, \langle 1,-1\rangle, \langle 0,1\rangle, \langle 0,-1\rangle\}$. Here $\mathfrak{A}$ is the union of the set of step vectors used in Motzkin paths with the set of up and down step vectors. We use the term *vertically constrained* to indicate paths that do not have consecutive vertical steps. That is, the following sequences are not permitted: $\uparrow\uparrow$, $\downarrow\downarrow$, as well as the self intersecting sequences $\uparrow\downarrow$ and $\downarrow\uparrow$.

In Section 6.1 and Section 6.2, we describe the recurrence relation and derive a generating function for $\mathfrak{A}$ paths in the half plane. In Section 6.3, we identify a bijection between $\mathfrak{A}$ paths and Motzkin paths and use this bijection to derive a generating function for $\mathfrak{A}$ paths in the quarter plane. The same approach is applied in Section 6.4 to extend both the Dyck and Schröder class of paths. We conclude with some general properties for this family of partially directed, vertically constrained lattice paths.

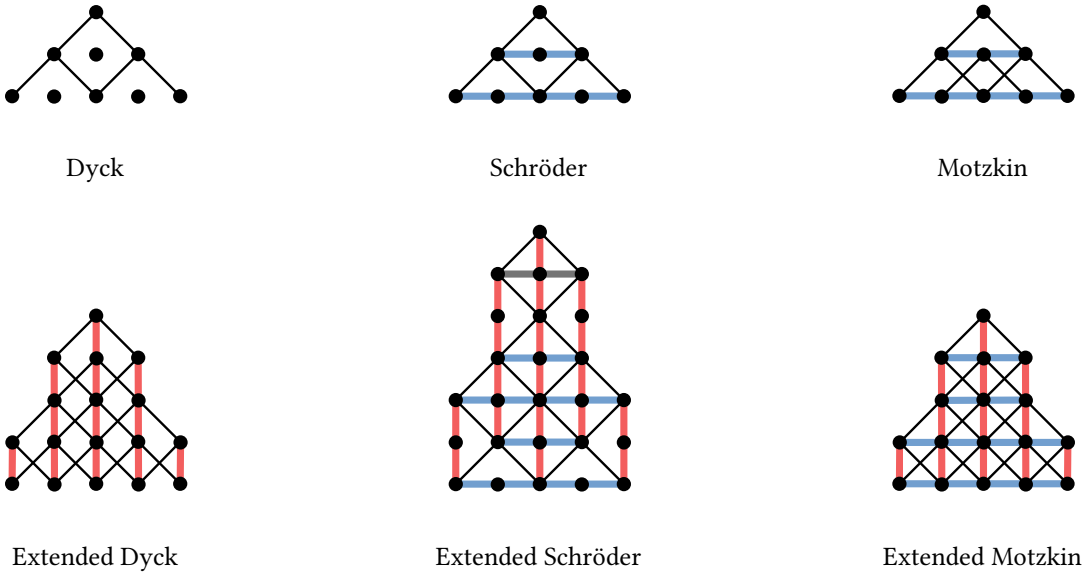


Figure 6.1: Underlying graphs induced by the complete set of paths in the quarter plane for some well known directed lattice path classes and their vertically constrained extensions

6.1 Recurrence Relations

As shown in Table 6.1, the lattice paths can be divided into four classes using two boolean properties:

1. Can the path extend below the x-axis? That is, is the path restricted to the quarter plane (Q) or can it travel anywhere in the half plane (H)?
2. Can the leading step be a vertical step vector? In other words, is the leading step, e_1, chosen from the set $\mathfrak{A}$ (a leading vertical step is allowed) or from the set $\mathfrak{M}$ (leading step is not vertical)?

The second property was mentioned in Chapter 5 with regard to wrapping of lace paths around the torus and not wishing to introduce consecutive vertical edges when closing the path. It will also play an important role in the bijection ϕ which will be discussed in Section 6.3.

Name	Quarter Plane	Leading Step	Description
$\mathbf{A^H}$	False	$e_1 \in \mathfrak{A}$	Leading vertical steps allowed
$\mathbf{A^H_R}$	False	$e_1 \in \mathfrak{M}$	No leading vertical steps
$\mathbf{A^Q}$	True	$e_1 \in \mathfrak{A}$	Restricted to quarter plane, leading vertical steps allowed
$\mathbf{A^Q_R}$	True	$e_1 \in \mathfrak{M}$	Restricted to quarter plane, no leading vertical steps

Table 6.1: Four classes of vertically constrained lattice paths

Define $\mathbf{A^H}$ to be the set of partially directed, vertically constrained lattice paths in the half plane created from step vectors $\mathfrak{A}$. The number of paths of type $\mathbf{A^H}$ starting at point $(0,0)$ and terminating at point (n,m) is given by $A^H(n,m)$.

For proposition P, let $[\![P]\!]$ be 1 if P is true and 0 otherwise.

Lemma 6.1.1. *$A^H(n,m)$ satisfies the recurrence relation:*

$$A^H(0,m) = [\![m \in \{-1,0,1\}]\!], \tag{6.1}$$

otherwise, for $n > 0$,

$$\begin{aligned} A^H(n,m) = {} & A^H(n-1,m+2)+2A^H(n-1,m+1)+3A^H(n-1,m) \\ & +2A^H(n-1,m-1)+A^H(n-1,m-2). \end{aligned} \tag{6.2}$$

Following the technique used by Donaghey and Shapiro to describe the Motzkin triangle [22], equation (6.2) is obtained by examining all of the ways in which an $\mathbf{A^H}$ path can terminate at a lattice point as illustrated in Figure 6.2. The same technique is used to derive all recurrence relations in this chapter.

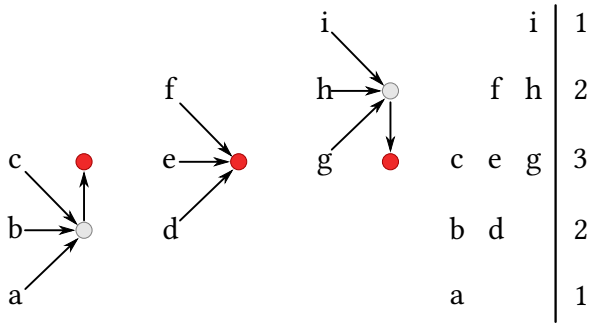


Figure 6.2: All possible ways in which a vertically constrained path with steps from the set $\mathfrak{A}$ can terminate at a lattice point (indicated by red vertex)

In a similar manner, $\mathbf{A}_{\mathbf{R}}^{\mathbf{H}}$ is the set of partially directed, vertically constrained lattice paths in the half plane created from step vector $\mathfrak{A}$ in which the leading step is restricted to $\mathfrak{M}$. The number of paths that travel from the origin to point (n, m), is represented by $A_R^H(n, m)$. The initial condition is $A_R^H(0, m) = [\![m = 0]\!]$. Equation (6.2) can be rewritten for $A_R^H(n, m)$ by substituting $A_R^H(a, b)$ for $A^H(a, b)$.

Paths that are restricted to the quarter plane follow a similar pattern but require special handling for positions close to the x-axis. We define $\mathbf{A}^{\mathbf{Q}}$ as the set of partially directed, vertically constrained lattice paths restricted to the quarter plane, created from step vectors $\mathfrak{A}$. The count of $\mathbf{A}^{\mathbf{Q}}$ paths terminating at (n, m) is given by $A^Q(n, m)$.

Lemma 6.1.2. *The recurrence relation for $A^Q(n, m)$ is as follows:*

$$A^Q(0, m) = [\![m \in \{0, 1\}]\!], \tag{6.3}$$

otherwise, for $n > 0$,

$$A^Q(n, 0) = A^Q(n-1, 2)+2A^Q(n-1, 1)+2A^Q(n-1, 0), \tag{6.4}$$

$$A^Q(n, 1) = A^Q(n-1, 3)+2A^Q(n-1, 2)+3A^Q(n-1, 1)+2A^Q(n-1, 0), \tag{6.5}$$

and, for $n > 0, m > 1$,

$$\begin{aligned} A^Q(n, m) = {} & A^Q(n-1, m+2)+2A^Q(n-1, m+1)+3A^Q(n-1, m) \\ & +2A^Q(n-1, m-1)+A^Q(n-1, m-2). \end{aligned} \tag{6.6}$$

The recurrence relation for the set of $\mathbf{A}_{\mathbf{R}}^{\mathbf{Q}}$ paths in which the leading step is restricted to $\mathfrak{M}$ has initial condition $A_R^Q(0, m) = [\![m = 0]\!]$. Equations (6.4),(6.5) and (6.6) can be rewritten for $\mathbf{A}_{\mathbf{R}}^{\mathbf{Q}}$ paths by substituting $A_R^Q(a, b)$ for $A^Q(a, b)$.

6.2 Generating Functions from Recurrence Relations

Theorem 6.2.1. *The generating function $a^H(x,y)$ for the triangle of values of $A^H(n,m)$ is*

$$\begin{aligned} a^H(x,y) &= \sum_{n\geq 0}\sum_{m} A^H(n,m)x^n y^m \\ &= \frac{y\,(1+y+y^2)}{y^2 - x\,(1+y+y^2)^2}. \end{aligned} \tag{6.7}$$

Proof. Expand terms in the general summation using the recurrence relation of Lemma 6.1.1.

$$\begin{aligned} a^H(x,y) &= \sum_{n\geq 0}\sum_{m} A^H(n,m)x^n y^m \\ &= \sum_{m} A^H(0,m)y^m + \sum_{n\geq 1}\sum_{m}(A^H(n-1,m+2) + 2A^H(n-1,m+1) \\ &\quad +3A^H(n-1,m) + 2A^H(n-1,m-1) + A^H(n-1,m-2))x^n y^m \\ &= y^{-1} + 1 + y \\ &\quad +xy^{-2}\sum_{n\geq 1}\sum_{m} A^H(n-1,m+2)x^{n-1}y^{m+2} \\ &\quad +2xy^{-1}\sum_{n\geq 1}\sum_{m} A^H(n-1,m+1)x^{n-1}y^{m+1} \\ &\quad +3x\sum_{n\geq 1}\sum_{m} A^H(n-1,m)x^{n-1}y^{m} \\ &\quad +2xy\sum_{n\geq 1}\sum_{m} A^H(n-1,m-1)x^{n-1}y^{m-1} \\ &\quad +xy^2\sum_{n\geq 1}\sum_{m} A^H(n-1,m-2)x^{n-1}y^{m-2} \\ &= y^{-1} + 1 + y + xy^{-2}a^H(x,y) + 2xy^{-1}a^H(x,y) + 3xa^H(x,y) \\ &\quad +2xya^H(x,y) + xy^2a^H(x,y) \end{aligned}$$

Rearrange and solve for $a^H(x,y)$. □

One can similarly prove Theorem 6.2.2 using the recurrence relation of Lemma 6.1.2:

Theorem 6.2.2. *The generating function for the $A_R^H(n,m)$ triangle, represented as $a_R^H(x,y)$, is*

$$\begin{aligned} a_R^H(x,y) &= \sum_{n\geq 0}\sum_{m} A_R^H(n,m)x^n y^m \\ &= \frac{y^2}{y^2 - x\left(1+y+y^2\right)^2}. \end{aligned} \tag{6.8}$$

6.3 An Explicit Bijection

Motzkin family	Relationship	OEIS[77]
Motzkin Paths	$A_R^Q(n,0) = M^Q(2n)$	A026945
	$A^Q(n,0) = M^Q(2n+1)$	A099250
Grand Motzkin Paths	$A_R^H(n,0) = M^H(2n)$	A082758
	$A^H(n,0) = M^H(2n+1)$	Bisection of A002426
Motzkin Triangle	$A_R^Q(n,m) = \triangle_M(2n,m)$	Row Bisection of A026300
	$A^Q(n,m) = \triangle_M(2n+1,m)$	Row Bisection of A026300
Triangle of Trinomial Coefficients	$A_R^H(n,m) = \triangle_T(2n,m)$	Row Bisection of A027907
	$A^H(n,m) = \triangle_T(2n+1,m)$	Row Bisection of A027907

Table 6.2: Correlation between vertically constrained paths and their Motzkin counterparts.

Definition (*Bisection*). A bisection is a mapping to alternating (even or odd) elements of a sequence. The elements may be individual values or rows of values, as specified.

As indicated in Table 6.2, there is a relationship between the vertically constrained $\mathfrak{A}$ lattice paths and a bisection of the Motzkin paths. We prove this relationship by providing an explicit bijection.

We define the mapping ϕ, illustrated in Figure 6.3, in which the following substitutions are performed on alternating steps of the Motzkin path:

$$\begin{aligned}\phi: &\quad \langle 1,1\rangle &&\to \langle 0,1\rangle\\ \phi: &\quad \langle 1,-1\rangle &&\to \langle 0,-1\rangle\\ \phi: &\quad \langle 1,0\rangle &&\to \text{remove step}\end{aligned}$$

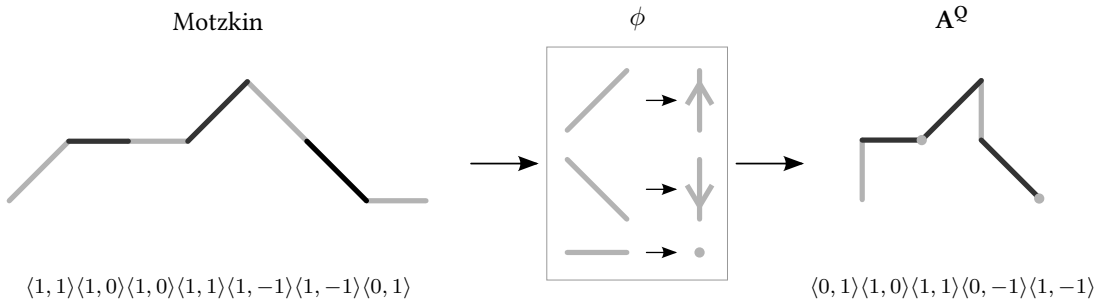


Figure 6.3: Conversion of Motzkin path ($n = 7$) to $\mathbf{A}^{\mathbf{Q}}$ path ($n = 3$). Grey steps are modified using the substitution rules of ϕ. Black steps are unchanged.

Lemma 6.3.1. *The mapping ϕ applied to alternating steps starting at the first step of a Motzkin (Grand Motzkin) path terminating at $(2n+1, m)$ produces a vertically constrained path of type $\mathbf{A}^{\mathbf{Q}}$ ($\mathbf{A}^{\mathbf{H}}$) terminating at (n, m).*

Proof. Step vectors in the Motzkin paths belong to $\mathfrak{M}$ which is a subset of $\mathfrak{A}$. All substituted step vectors in the mapping ϕ belong to $\mathfrak{A}$. Therefore the resulting path contains only step vectors from $\mathfrak{A}$. Only alternating steps of the Motzkin path are modified therefore consecutive vertical step vectors cannot be introduced by ϕ. The resulting path is therefore a vertically constrained path of type $\mathbf{A}^{\mathbf{Q}}$ ($\mathbf{A}^{\mathbf{H}}$). At each substitution, the horizontal length of the path is decreased by 1. The mapping starts with a path of length $2n + 1$ and horizontally collapses $n + 1$ steps resulting in a path of length n. Only horizontal displacement is affected by the substitutions of ϕ leaving the height of endvertices along the path unchanged. Therefore, an operand restricted to one quadrant produces a result restricted to one quadrant and an operand that terminates at height m produces a result that terminates at height m. □

Lemma 6.3.2. *The mapping ϕ applied to alternating steps starting at the second step of a Motzkin (Grand Motzkin) path terminating at $(2n, m)$ produces a path of type $\mathbf{A}^{\mathbf{Q}}_{\mathbf{R}}$ ($\mathbf{A}^{\mathbf{H}}_{\mathbf{R}}$) (restricted leading step) terminating at (n, m).*

Proof. The same arguments used in Lemma 6.3.1 apply. Because the mapping starts with the second step of the Motzkin path, the first step of the resulting path belongs to $\mathfrak{M}$ and n steps are collapsed horizontally. □

The inverse relationship, taking a vertically constrained $\mathfrak{A}$ path to a Motzkin path, can be represented by the mapping ψ, illustrated in Figure 6.4, in which the following substitutions are performed on steps of the vertically constrained $\mathfrak{A}$ path centered at a horizontal distance i from the origin (that is, the midpoint of the step is at a horizontal distance i from the origin).

$$\begin{aligned} \psi: &\quad \langle 0,1\rangle &&\rightarrow \langle 1,1\rangle \\ \psi: &\quad \langle 0,-1\rangle &&\rightarrow \langle 1,-1\rangle \\ \psi: &\quad \text{no step} &&\rightarrow \text{insert } \langle 1,0\rangle \end{aligned}$$

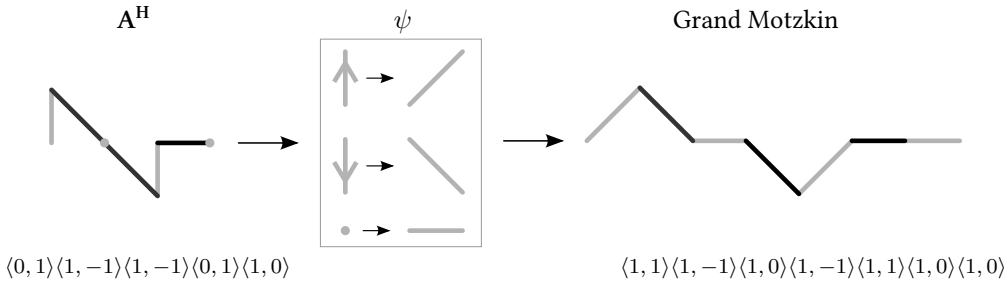


Figure 6.4: Conversion of $\mathbf{A^H}$ path ($n = 3$) to Grand Motzkin path ($n = 7$)

Lemma 6.3.3. *The mapping ψ applied at a horizontal distance i from the origin for all $0 \leq i \leq n$ to a vertically constrained $\mathbf{A^Q}$ ($\mathbf{A^H}$) path terminating at (n, m) produces a Motzkin (Grand Motzkin) path terminating at $(2n + 1, m)$.*

Proof. The mapping ψ replaces all vertical steps with steps belonging to $\mathfrak{M}$. The resulting path is therefore a Motzkin or Grand Motzkin path. The substitutions do not affect the height of endvertices along the path, therefore, an operand restricted to a quadrant produces a result restricted to a quadrant and an operand that terminates at height y produces a result that terminates at height y. At each substitution, the horizontal length of the path is increased by 1. Since $n + 1$ substitutions are performed on an operand of length n, the resulting path has length $2n + 1$. □

Lemma 6.3.4. *The mapping ψ applied at a horizontal distance i from the origin for all $0 < i \leq n$ to a vertically constrained $\mathbf{A}_{\mathbf{R}}^{\mathbf{Q}}$ ($\mathbf{A}_{\mathbf{R}}^{\mathbf{H}}$) path terminating at (n, m) produces a Motzkin (Grand Motzkin) path terminating at $(2n, m)$.*

Proof. The same arguments used in Lemma 6.3.3 apply. Because the substitutions begin at $i = 1$, the number of substitutions is n and the length of the resulting path is $2n$. □

Theorem 6.3.5. *The Triangle of Trinomial Coefficients is described by the following recurrence relation [3, 22]:*

$$\triangle_T(0, m) = [\![m = 0]\!] \text{ if } n = 0, \text{ otherwise} \tag{6.9}$$

$$\triangle_T(n, m) = \triangle_T(n-1, m-1)+\triangle_T(n-1, m)+\triangle_T(n-1, m+1). \tag{6.10}$$

Theorem 6.3.6. *There is a bijection between Motzkin (Grand Motzkin) paths terminating at $(2n + 1, m)$ and vertically constrained paths of type $\mathbf{A}^{\mathbf{Q}}$ ($\mathbf{A}^{\mathbf{H}}$) terminating at (n, m).*

Proof. First we prove that the mapping ϕ is an injection. The operand for ϕ is a member of the well known set of Motzkin paths. Through ϕ, each step vector in the operand is either copied directly to the output or uniquely mapped to a step vector that is not an element in $\mathfrak{M}$. The uniqueness of the operand is preserved by ϕ, therefore, ϕ is an injection from a Motzkin (Grand Motzkin) path to a vertically constrained $\mathbf{A}^{\mathbf{Q}}$ ($\mathbf{A}^{\mathbf{H}}$) path.

The mapping ϕ is also a surjection. To prove this, we demonstrate that the cardinality of the set of vertically constrained paths of type $\mathbf{A}^{\mathbf{H}}$ terminating at (n, m) is equal to the cardinality of the set of Grand Motzkin paths terminating at $(2n+1, m)$. The equivalence of the cardinality of the two path sets can be proven by induction using Lemma 6.1.1 and Theorem 6.3.5.

It is clear that $A^H(0, m) = \triangle_T(1, m)$ by the initial conditions described in Equations (6.1) and (6.9). Assuming that $A^H(n, m) = \triangle_T(2n + 1, m)$, we now show that $A^H(n + 1, m) = \triangle_T(2n + 3, m)$.

Using the recurrence relation described in Equation (6.10), $\triangle_T(2n+3, m)$ is expanded:

$$\begin{aligned}\triangle_T(2n+3, m) = {} & \triangle_T(2n + 1, m-2)+2\triangle_T(2n + 1, m-1)+3\triangle_T(2n + 1, m) \\ & +2\triangle_T(2n + 1, m+1)+\triangle_T(2n + 1, m+2)\end{aligned} \tag{6.11}$$

Next, Equation (6.2) is applied to $A^H(n+1,m)$ and the substitution $A^H(n,m) = \triangle_T(2n+1,m)$ is performed:

$$\begin{aligned} A^H(n{+}1,m) &= A^H(n,m{-}2){+}2A^H(n,m{-}1){+}3A^H(n,m) \\ &\quad {+}2A^H(n,m{+}1){+}A^H(2n,m{+}2) \\ A^H(n{+}1,m) &= \triangle_T(2n{+}1,m{-}2){+}2\triangle_T(2n{+}1,m{-}1){+}3\triangle_T(2n{+}1,m) \\ &\quad {+}2\triangle_T(2n{+}1,m{+}1){+}\triangle_T(2n{+}1,m{+}2) \\ &= \triangle_T(2n{+}3,m) \text{ by Equation (6.11)} \end{aligned} \tag{6.12}$$

This proves that ϕ is bijection between Grand Motzkin paths and $\mathbf{A^H}$ paths in the half-plane. In a similar fashion, the recurrence relation for the Motzkin Triangle and Lemma 6.1.2 can be used to prove that ϕ is a bijection between Motzkin paths and $\mathbf{A^Q}$ paths restricted to one quadrant. □

Theorem 6.3.7. *There is a bijection from Motzkin (Grand Motzkin) paths of length $2n$ to vertically constrained paths of type $\mathbf{A^Q_R}$ ($\mathbf{A^H_R}$) and length n.*

Proof. The proof follows from similar arguments to those used in Theorem 6.3.6. □

In Figure 6.5, it is obvious that vertically constrained paths with a restricted leading step are a subset of the unrestricted vertically constrained paths of the same horizontal length. By the bijection mapping ϕ, this relationship also exits between even and odd length Motzkin paths. If we prepend a horizontal step, Motzkin paths of length $2n$ form a unique subset of Motzkin paths of length $2n+1$.

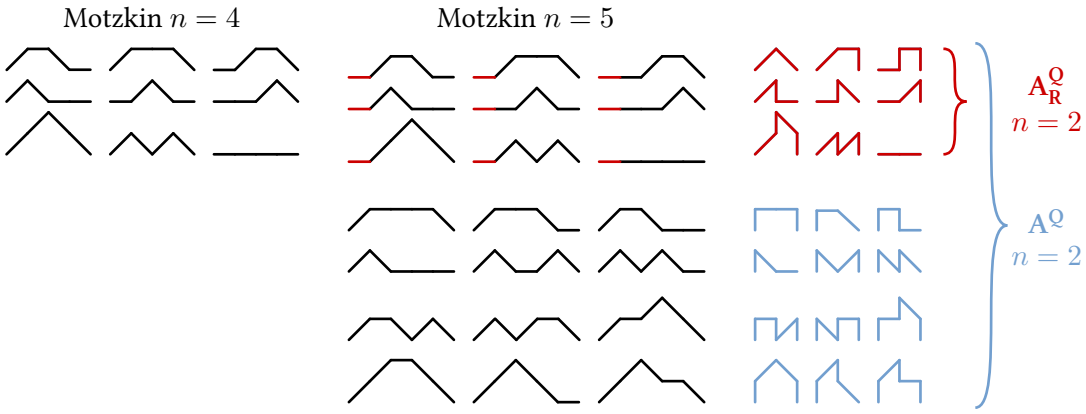


Figure 6.5: Motzkin paths ($n = 4$ and $n = 5$) and corresponding $\mathbf{A^Q_R}$ and $\mathbf{A^Q}$ paths ($n = 2$).

6.3.1 Generating Functions Derived from Bijection

Theorem 6.3.8. *Let $a^H(x,0)$ and $a_R^H(x,0)$ be the generating functions for the subset of $\mathbf{A^H}$ and $\mathbf{A_R^H}$ paths which terminate on the x-axis, respectively. The ordinary univariate generating functions are as follows:*

$$a^H(x,0) := \sum_{n\geq 0} A^H(n,0)x^n = \frac{1}{2\sqrt{x}}\left(\frac{1}{D(x)} - \frac{1}{E(x)})\right)$$

and

$$a_R^H(x,0) := \sum_{n\geq 0} A_R^H(n,0)x^n = \frac{1}{2}\left(\frac{1}{D(x)} + \frac{1}{E(x)}\right)$$

where

$$D(x) := \sqrt{1-2\sqrt{x}-3x}$$
$$E(x) := \sqrt{1+2\sqrt{x}-3x}.$$

Proof. The generating functions are derived by bisecting the generating function for the corresponding Grand Motzkin paths ($m^H(x,0)$) [6].

$$m^H(x,0) = \frac{1}{\sqrt{1-2x-3x^2}} \tag{6.13}$$

□

Theorem 6.3.9. *Let $a^Q(x,y)$ and $a_R^Q(x,y)$ be the generating functions for the $\mathbf{A^Q}$ and $\mathbf{A_R^Q}$ paths, respectively. The ordinary bivariate generating functions are as follows:*

$$a^Q(x,y) = \frac{1}{2\sqrt{x}}\left(F(x,y) - G(x,y)\right)$$
$$a_R^Q(x,y) = \frac{1}{2}\left(F(x,y) + G(x,y)\right)$$

where

$$F(x,y) = m^Q(\sqrt{x},y) = \frac{1-\sqrt{x}(1+2y)-\sqrt{1-2\sqrt{x}-3x}}{2\sqrt{x}\left(\sqrt{x}(1+y+y^2)-y\right)}$$
$$G(x,y) = m^Q(-\sqrt{x},y) = \frac{1+\sqrt{x}(1+2y)-\sqrt{1+2\sqrt{x}-3x}}{2\sqrt{x}\left(\sqrt{x}(1+y+y^2)+y\right)}$$

Proof. Generating functions are derived by bisecting the generating function for the corresponding Motzkin triangle ($m^{Q}(x,y)$) derived by Barcucci et al. [6] and simplifying.

$$m^{Q}(x,y) = \frac{1-x-2xy-\sqrt{1-2x-3x^2}}{2x(xy^2+x+xy-y)} \tag{6.14}$$

□

Theorem 6.3.10. *Let $a^{Q}(x,0)$ and $a^{Q}_{R}(x,0)$ be the generating functions for the subset of $\mathbf{A}^{\mathbf{Q}}$ and $\mathbf{A}^{\mathbf{Q}}_{\mathbf{R}}$ paths terminating on the x-axis, respectively. The ordinary univariate generating functions are as follows:*

$$a^{Q}(x,0) = \frac{-2\sqrt{x}-\sqrt{1-2\sqrt{x}-3x}+\sqrt{1+2\sqrt{x}-3x}}{4x\sqrt{x}}$$
$$a^{Q}_{R}(x,0) = \frac{2-\sqrt{1-2\sqrt{x}-3x}-\sqrt{1+2\sqrt{x}-3x}}{4x}$$

Proof. Generating functions are obtained by bisecting the generating function for Motzkin paths [6] and simplifying.

$$m^{Q}(x,0) = \frac{1-x-\sqrt{1-2x-3x^2}}{2x^2} \tag{6.15}$$

□

6.4 Extension to Other Lattice Paths

Our approach can be used to explore vertically constrained counterparts to other well known lattice paths such as Dyck, Schröder and Delannoy paths as shown in Figure 6.1.

6.4.1 Dyck Paths with Vertical Steps

Dyck paths are composed from the step vectors $\mathfrak{D} = \{\langle 1,1\rangle, \langle 1,-1\rangle\}$. We now consider vertically constrained lattice paths with the vector step set $\mathfrak{B} = \{\langle 1,1\rangle, \langle 1,-1\rangle, \langle 0,1\rangle, \langle 0,-1\rangle\}$. The four types of $\mathfrak{B}$ paths and their relationship to known OEIS integer sequences are presented in Tables 6.3 and 6.4.

Name	Quarter Plane	Leading Step
$\mathbf{B}^{\mathrm{H}}$	False	$e_1 \in \mathfrak{B}$
$\mathbf{B}^{\mathrm{H}}_{\mathrm{R}}$	False	$e_1 \in \mathfrak{D}$
$\mathbf{B}^{\mathrm{Q}}$	True	$e_1 \in \mathfrak{B}$
$\mathbf{B}^{\mathrm{Q}}_{\mathrm{R}}$	True	$e_1 \in \mathfrak{D}$

Table 6.3: Vertically constrained lattice paths with vector step set $\mathfrak{B}$.

Family	Relationship	OEIS[77]
Motzkin, no flat steps at odd indices	$B^{Q}(n,0) = M^{Q}_{\mathbb{Q}}(2n+1,0)$	Bisection of A214938
Motzkin, no flat steps at even indices	$B^{Q}_{R}(n,0) = M^{Q}_{\mathbb{Q}}(2n,0)$	Bisection of A214938
Grand Motzkin, no flat steps odd indices	$B^{H}(n,0) = M^{H}_{\mathbb{Q}}(2n+1,0)$	Bisection of A026520
Grand Motzkin, no flat steps even indices	$B^{H}_{R}(n,0) = M^{H}_{\mathbb{Q}}(2n,0)$	Bisection of A026520
Grand Motzkin, no flat steps odd indices	$B^{H}(n,m) = M^{H}_{\mathbb{Q}}(2n+1,m)$	Column bisection of A026519
Grand Motzkin, no flat steps even indices	$B^{H}_{R}(n,m) = M^{H}_{\mathbb{Q}}(2n,m)$	Column bisection of A026519

Table 6.4: Correlation between vertically constrained paths of type $\mathfrak{B}$ and subsets of the Motzkin family of paths

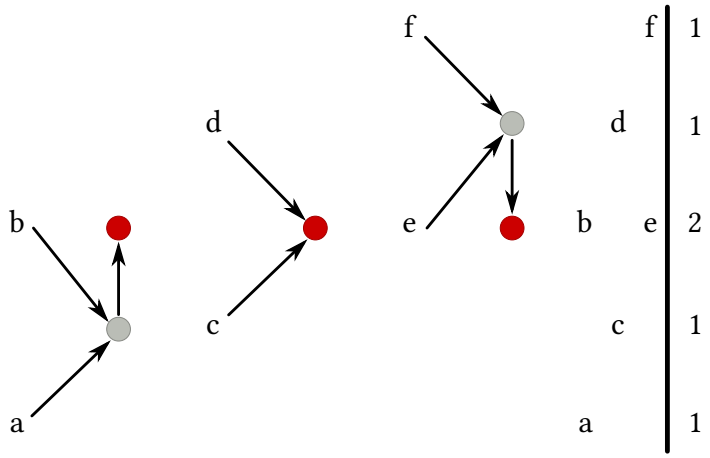


Figure 6.6: All possible ways a path using steps from the set $\mathfrak{B}$ and without consecutive vertical steps, can terminate at a lattice point (indicated by red dot)

As before, we determine the recurrence relation by considering all ways that vertically constrained paths constructed from the set of step vectors $\mathfrak{B}$ can terminate at a lattice point.

Lemma 6.4.1. *Let $B^H(n,m)$ be the number of vertically constrained paths in the half plane created from step vectors $\mathfrak{B}$ and extending from $(0,0)$ to (n,m). The recurrence relation for $B^H(n,m)$ is*

$$B^H(0,m) = [\![m \in \{-1,0,1\}]\!], \tag{6.16}$$

otherwise, for $n > 0$,

$$\begin{aligned} B^H(n,m) &= B^H(n-1,m+2)+B^H(n-1,m+1)+2B^H(n-1,m) \\ &\quad +B^H(n-1,m-1)+B^H(n-1,m-2). \end{aligned} \tag{6.17}$$

For $\mathbf{B}^{\mathbf{H}}_{\mathbf{R}}$ paths, in which the first step is restricted to the set $\mathfrak{D}$, the initial condition is $B^H_R(0,m) = [\![m = 0]\!]$. Equations 6.16 and 6.17 can be rewritten for $\mathbf{B}^{\mathbf{H}}_{\mathbf{R}}$ by substituting $B^H_R(a,b)$ for $B^H(a,b)$.

The recurrence relation for $\mathbf{B^Q}$ and $\mathbf{B^Q_R}$, paths restricted to the quarter plane, is very similar but with additional conditions concerning the first two rows of the triangle:

Lemma 6.4.2. *Let $B^Q(n,m)$ be the number of vertically constrained paths in the quarter plane created from step vectors $\mathfrak{B}$ and extending from $(0,0)$ to (n,m). The recurrence relation for $B^Q(n,m)$ is*

$$B^Q(0,m) = [\![m \in \{0,1\}]\!],$$

otherwise, for $n > 0$,

$$B^Q(n,0) = B^Q(n-1,2)+B^Q(n-1,1)+B^Q(n-1,0), \tag{6.18}$$

$$B^Q(n,1) = B^Q(n-1,3)+B^Q(n-1,2)+2B^Q(n-1,1)+B^Q(n-1,0) \tag{6.19}$$

and for $n > 0$, $m > 1$,

$$\begin{aligned} B^Q(n,m) &= B^Q(n-1,m+2)+B^Q(n-1,m+1)+2B^Q(n-1,m) \\ &\quad +B^Q(n-1,m-1)+B^Q(n-1,m-2). \end{aligned} \tag{6.20}$$

For $\mathbf{B^Q_R}$ paths, in which the first step is restricted to the set $\mathfrak{D}$, the initial condition is $B^Q_R(0,m) = [\![m = 0]\!]$. Equations 6.18, 6.19 and 6.20 can be rewritten for $\mathbf{B^Q_R}$ by substituting $B^Q_R(a,b)$ for $B^Q(a,b)$.

The generating functions for $\mathbf{B^H}$ and $\mathbf{B^H_R}$ paths can be determined by expanding terms in the general summation using the recurrence relation:

Theorem 6.4.3. *The ordinary bivariate generating function for $B^H(n,m)$ is*

$$\begin{aligned} b^H(x,y) &:= \sum_{n\geq 0}\sum_{m} B^H(n,m)x^n y^m \\ &:= \frac{y\,(1+y+y^2)}{y^2 - x\,(1+y^2)\,(1+y+y^2)} \end{aligned}$$

Theorem 6.4.4. *The ordinary bivariate generating function for $B^H_R(n,m)$ is*

$$\begin{aligned} b^H_R(x,y) &= \sum_{n\geq 0}\sum_{m} B^H_R(n,m)x^n y^m \\ &= \frac{y^2}{y^2 - x\,(1+y^2)\,(1+y+y^2)} \end{aligned}$$

6.4.2 Explicit Bijection for Dyck Paths with Vertical Steps

The OEIS integer sequences A026520 and A214938 referenced in Table 6.3 correspond to a subset of Motzkin paths in which horizontal steps appear only at alternating indices. The bijection between this Motzkin subset and the vertically constrained Dyck paths can be described using the same ϕ and ψ mappings outlined in Section 6.3.

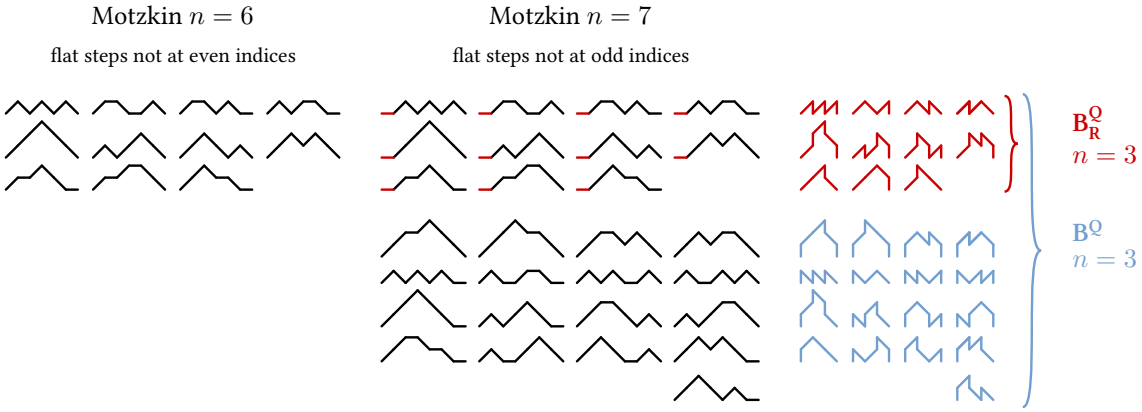


Figure 6.7: Motzkin paths with flat steps restricted to odd or even indices and corresponding $\mathbf{B}_{\mathbf{R}}^{\mathbf{Q}}$ or $\mathbf{B}^{\mathbf{Q}}$ paths respectively

Theorem 6.4.5. *There is a bijection between the subset of Motzkin (Grand Motzkin) paths terminating at* $(2n+1, m)$ *that avoid flat steps at* ***odd*** *indices and the vertically constrained paths of type* $\mathbf{B}^{\mathbf{Q}}$ *(*$\mathbf{B}^{\mathbf{H}}$*) terminating at* (n, m).

Proof. This bijection can be described using the same ϕ and ψ mappings outlined in Section 6.3. Start with a Motzkin path (or Grand Motzkin path) of length $2n+1$ in which the step vector $\langle 1, 0\rangle$ occurs only at **odd** indices. The application of ϕ to the first step, e_1, and subsequent steps of odd index, removes all horizontal steps during the substitution phase and inserts non-consecutive vertical steps. The resulting path contains only steps found in $\mathfrak{B}$ and is of type $\mathbf{B}^{\mathbf{Q}}$ or $\mathbf{B}^{\mathbf{H}}$. Inversely, if we start with a $\mathbf{B}^{\mathbf{Q}}$ or $\mathbf{B}^{\mathbf{H}}$ path and apply ψ, all vertical steps are removed and horizontal steps, when added, occur only at odd indices producing a Motzkin path (or Grand Motzkin path) of the desired subset. □

Theorem 6.4.6. *There is a bijection between the subset of Motzkin (Grand Motzkin) paths terminating at* $(2n, m)$ *that avoid flat steps at* ***even*** *indices and the vertically constrained paths of type* $\mathbf{B}^{\mathbf{Q}}$ *(*$\mathbf{B}^{\mathbf{H}}$*) terminating at* (n, m).

Proof. The proof of this theorem follows from that of Theorem 6.4.5. By applying ϕ to the second step and alternating steps thereafter, the first step cannot be vertical. Similarly, ψ removes all horizontal steps at even indices in the resulting Motzkin path. □

It is important to note that if we reverse a Motzkin path of even length with horizontal steps not allowed at **even** indices we obtain a Motzkin path with horizontal steps not allowed at **odd** indices.

Generating functions for $\mathbf{B}^{\mathbf{Q}}$ and $\mathbf{B}_{\mathbf{R}}^{\mathbf{Q}}$ paths (restricted to the quarter plane) have been partially determined by expanding terms in the general summation using the recurrence relations.

Theorem 6.4.7. *The ordinary bivariate generating function for $B^Q(n,m)$, in terms of its first two columns, is*

$$\begin{aligned} b^Q(x,y) &:= \sum_{n\geq 0}\sum_{m\geq 0} B^Q(n,m)x^n y^m \\ &= \frac{y^2+y^3-x\,(1+y+y^2)\,R_0(x) - xyR_1(x)}{y^2 - x\,(1+y^2)\,(1+y+y^2)}, \end{aligned}$$

where

$$R_i(x) := \sum_{n\geq 0} B^Q(n,i)x^n, \quad i \in 0,1.$$

Theorem 6.4.8. *The ordinary bivariate generating function for $B_R^Q(n,m)$, in terms of its first two columns, is*

$$b_R^Q(x,y) := \sum_{n\geq 0}\sum_{m\geq 0} B_R^Q(n,m)x^n y^m \tag{6.21}$$

$$= \frac{y^2 - x\,(1+y+y^2)\,S_0(x) - xyS_1(x)}{y^2 - x\,(1+y^2)\,(1+y+y^2)} \tag{6.22}$$

where

$$S_i(x) := \sum_{n\geq 0} B_R^Q(n,i)x^n, \quad i \in 0,1 \tag{6.23}$$

To find a full explicit form for these generating functions, we will make use of the bijection relation between vertically constrained $\mathbf{B}^{\mathfrak{Q}}$ paths and Motzkin paths. A generating function for Motzkin paths that avoid flat steps at odd indices was proposed by Hanna [77, A214938]. It makes use of InverseSeries$(f(x))$ which is the Taylor series expansion of the inverse of function $f(x)$ given by the Lagrange inversion theorem:

Conjecture 6.4.9. *[77, A214938] The ordinary univariate generating function for Motzkin paths terminating on the x-axis in which flat steps are not permitted at odd indices is*

$$m^{Q}_{\mathfrak{Q}}(x,0) := \sum_{n\geq 0} M^{Q}_{\mathfrak{Q}}(n,0)x^n = J(x^2) + xK(x^2) \tag{6.24}$$

where

$$J(x) := \frac{1}{x}\text{InverseSeries}\left(\frac{x(1-x)(1-2x)^2}{1-4x+5x^2-2x^3+x^4}\right),$$
$$K(x) := \frac{1}{x}\text{InverseSeries}\left(\frac{x}{1+2x+3x^2+2x^3+2x^6(Cat(-x^2))^3}\right), \text{ and}$$
$$Cat(x) := \frac{1-\sqrt{1-4x}}{2x}.$$

The expression for $K(x)$ can be simplified to

$$K(x) = \frac{1}{x}\text{InverseSeries}\left(\frac{x\sqrt{1+4x^2}-2x^2}{1+x^2}\right).$$

The following lemma is required by the proof of Theorem 6.4.12.

Lemma 6.4.10. *The set of $\mathbf{B}^{\mathfrak{Q}}$ paths terminating at $(n,1)$ is equinumerous with the set of $\mathbf{B}^{\mathfrak{Q}}_{\mathbf{R}}$ paths terminating at $(n+1,0)$, i.e.,*

$$B^{Q}(n,1) = B^{Q}_{R}(n+1,0).$$

Proof. Consider the set of paths of type $\mathbf{B}^{\mathfrak{Q}}_{\mathbf{R}}$ terminating at $(n+1,0)$. Because the first step is restricted to $\mathfrak{D}$ and the path is restricted to one quadrant, the first step in the path must be $\langle 1,1\rangle$. By removing the first step and reversing the direction of the entire path, a path of type $\mathbf{B}^{\mathfrak{Q}}$ terminating at $(n,1)$ is created. □

Corollary 6.4.11. *The number of* $\mathbf{B}^{\mathbf{Q}}$ *paths terminating at* $(n, 1)$ *can be expressed in terms of the number of Motzkin paths terminating on the* x*-axis with flat steps not allowed at odd indices, i.e.,*

$$B^Q(n, 1) = M^Q_{\diagdown\!\!\diagdown}(2n + 2, 0).$$

We can now derive generating functions for $\mathbf{B}^{\mathbf{Q}}$ paths in terms of the Lagrange inversion:

Conjecture 6.4.12. *The first two columns of the* $\mathbf{B}^{\mathbf{Q}}$ *triangle are*

$$R_0(x) := \sum_{n\geq 0} B^Q(n, 0)x^n = K(x)$$
$$R_1(x) := \sum_{n\geq 0} B^Q(n, 1)x^n = \frac{J(x)-1}{x}$$

and therefore the ordinary bivariate generating function for $B^Q(n, m)$ *is*

$$b^Q(x, y) = \frac{y(1 + y + y^2)-yJ(x)-x(1 + y + y^2)K(x)}{y^2-x\,(1+y^2)\,(1+y+y^2)}.$$

Proof. The solution for $R_0(x)$ is derived by bisecting Equation 6.24 to extract the even powers of x: $R_0(x) = (m^Q_{\diagdown\!\!\diagdown}(\sqrt{x}, 0) - m^Q_{\diagdown\!\!\diagdown}(-\sqrt{x}, 0))/(2\sqrt{x})$. The solution for $R_1(x)$ is obtained from Corollary 6.4.11. □

For $\mathbf{B}^{\mathbf{Q}}_{\mathbf{R}}$ paths, we require an additional relation between Motzkin paths given by the following lemma:

Lemma 6.4.13. *The number of Motzkin paths that terminate at* $(n, 1)$ *can be expressed in terms of the number of Motzkin paths terminating on the* x*-axis.*

$$M^Q(n, 1) = M^Q(n + 1, 0) - M^Q(n, 0).$$

Proof. Given a Motzkin path terminating at $(n, 1)$, the only way this path can be extended to a Motzkin path that terminates at $(n+1, 0)$ is by appending $\langle 1, -1\rangle$. The problem thus becomes one of finding the number of Motzkin path that terminate at $(n + 1, 0)$ with a final step of $\langle 1, -1\rangle$. There are only two possible ways in which a Motzkin path restricted to one quadrant can terminate: either by a $\langle 1, -1\rangle$ step or by a $\langle 1, 0\rangle$ step. If the path terminates at $(n+1, 0)$ with a $\langle 1, 0\rangle$ step, removing this final step will result in a Motzkin path terminating at $(n, 0)$. The set of paths terminating with a $\langle 1, -1\rangle$ step at $(n + 1, 0)$ can therefore be obtained by subtraction. □

Conjecture 6.4.14. *The first two columns of the* $\mathbf{B}_{\mathbf{R}}^{\mathbf{Q}}$ *triangle are*

$$S_0(x) := \sum_{n\geq 0} B_R^Q(n, 0)x^n = J(x)$$
$$S_1(x) := \sum_{n\geq 0} B_R^Q(n, 1)x^n = K(x) - J(x)$$

and therefore the ordinary bivariate generating function for $B_R^Q(n, m)$ *is*

$$b_R^Q(x, y) = \frac{y^2 - x(1 + y^2)J(x) - xyK(x)}{y^2 - x(1+y^2)(1+y+y^2)}$$

Proof. The solution for $S_0(x)$ is derived by bisecting Equation 6.24 to extract the even powers of x: $S_0(x) = (m_{\not\square}^Q(\sqrt{x}, 0) + m_{\not\square}^Q(-\sqrt{x}, 0))/2$. The solution for $S_1(x)$ is obtained by noting that Lemma 6.4.13 also applies for the subset of Motzkin paths of even length in which flat steps are not allowed at odd indices and taking the bisection of this expression to extract the even powers of x. □

Corollary 6.4.15. *The ordinary bivariate generating function for Motzkin paths in which flat steps are not permitted at odd indices is*

$$\begin{aligned} m_{\not\square}^Q(x, y) &= b_R^Q(x^2, y) + xb^Q(x^2, y) \\ &= \frac{x(x + y + xy^2)J(x^2) + (y + x(1 + y + y^2))(x^2K(x^2) - y)}{x^2(1+y^2)(1+y+y^2) - y^2}. \end{aligned}$$

6.4.3 Schröder and Delannoy Paths with Vertical Steps

Schröder and Delannoy paths are composed from the step vectors $\mathfrak{S} = \{\langle 1,1\rangle, \langle 1,-1\rangle, \langle 2,0\rangle\}$. Our vertically constrained variant is created from the step set $\mathfrak{C} = \{\langle 1,1\rangle, \langle 1,-1\rangle, \langle 2,0\rangle, \langle 0,2\rangle, \langle 0,-2\rangle\}$ where the height of the vertical step vector is equal to the width of the horizontal step vector. Table 6.5 outlines the four cases.

Name	Quarter Plane	Leading Step
$\mathbf{C}^{\mathbf{H}}$	False	$e_1 \in \mathfrak{C}$
$\mathbf{C}^{\mathbf{H}}_{\mathbf{R}}$	False	$e_1 \in \mathfrak{S}$
$\mathbf{C}^{\mathbf{Q}}$	True	$e_1 \in \mathfrak{C}$
$\mathbf{C}^{\mathbf{Q}}_{\mathbf{R}}$	True	$e_1 \in \mathfrak{S}$

Table 6.5: Vertically constrained lattice paths with vector step set $\mathfrak{C}$.

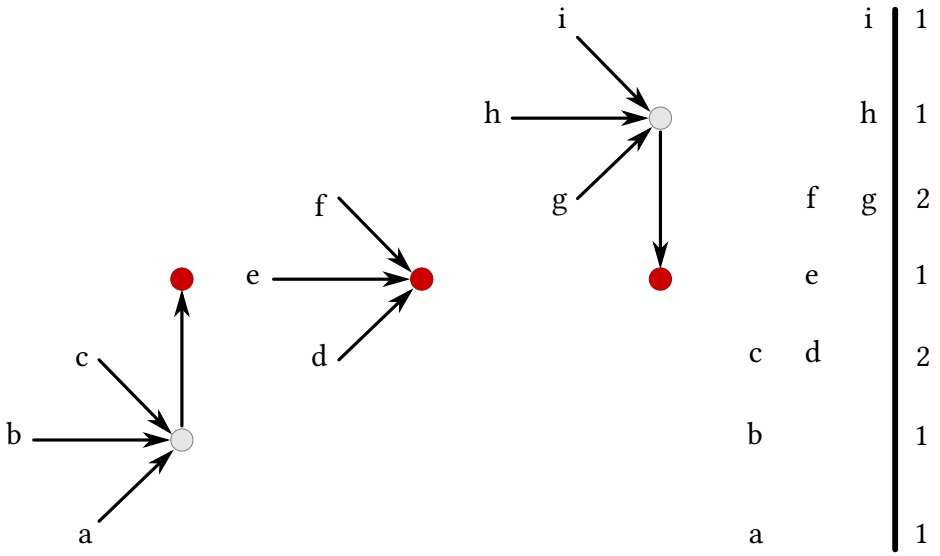


Figure 6.8: All possible ways a path can terminate at a lattice point (red dot) using step set $\mathfrak{C}$ and without consecutive vertical steps

The recurrence relation for $\mathbf{C}^{\mathbf{H}}$ paths can again be determined by considering all ways the path can terminate at a point as shown in Figure 6.8.

Lemma 6.4.16.

$$C^H(0,m) \;= [\![m \in \{-2,0,2\}]\!],$$

otherwise, for $n > 0$*,*

$$\begin{aligned} C^H(n,m) \;&= C^H(n{-}1,m{+}3){+}C^H(n{-}2,m{+}2){+}2C^H(n{-}1,m{+}1){+}C^H(n{-}2,m) \\ &\quad {+}2C^H(n{-}1,m{-}1){+}C^H(n{-}2,m{-}2){+}C^H(n{-}1,m{-}3) \end{aligned}$$

The recurrence relation for $\mathbf{C^Q}$ (restricted to the quarter plane) is very similar but with additional conditions concerning the first three rows of the triangle:

Lemma 6.4.17.

$$\begin{aligned} C^Q(p,m) \;&= 0, \text{ for } p < 0 \\ C^Q(0,m) \;&= [\![m \in \{0,2\}]\!] \end{aligned}$$

otherwise, for $n > 0$*,*

$$\begin{aligned} C^Q(n,0) \;&= C^Q(n{-}1,3){+}C^Q(n{-}2,2){+}2C^Q(n{-}1,1){+}C^Q(n{-}2,0) \text{ for } n > 0 \\ C^Q(n,1) \;&= C^Q(n{-}1,4){+}C^Q(n{-}2,3){+}2C^Q(n{-}1,2){+}C^Q(n{-}2,1){+}C^Q(n{-}1,0) \\ C^Q(n,2) \;&= C^Q(n{-}1,5){+}C^Q(n{-}2,4){+}2C^Q(n{-}1,3){+}C^Q(n{-}2,2){+}2C^Q(n{-}1,1) \\ &\quad {+}C^Q(n{-}2,0) \end{aligned}$$

and, for $n > 0, m > 2$*,*

$$\begin{aligned} C^Q(n,m) \;&= C^Q(n{-}1,m+3){+}C^Q(n{-}2,m+2){+}2C^Q(n{-}1,m+1){+}C^Q(n{-}2,m) \\ &\quad {+}2C^Q(n{-}1,m-1){+}C^Q(n{-}2,m-2){+}C^Q(n{-}1,m-3) \end{aligned}$$

The generating function for the $\mathbf{C^H}$ and $\mathbf{C^H_R}$ paths, expressed as $c^H(x,y)$ and $c^H_R(x,y)$, can be determined by expanding terms in the general summation using the recurrence relation.

Theorem 6.4.18. *Generating function for* $\mathbf{C^H}$*:*

$$\begin{aligned} c^H(x,y) &= \sum_{n\geq 0}\sum_{m} C^H(n,m)x^n y^m \\ &= \frac{y\,(1{+}y^2{+}y^4)}{y^3{-}x\,(1{+}xy{+}y^2)\,(1{+}y^2{+}y^4)} \end{aligned}$$

Theorem 6.4.19. *Generating function for* $\mathbf{C_R^H}$*:*

$$\begin{aligned} c_R^H(x,y) &= \sum_{n\geq 0}\sum_{m} C_R^H(n,m)x^n y^m \\ &= \frac{y^3}{y^3 - x\,(1+xy+y^2)\,(1+y^2+y^4)} \end{aligned}$$

The mappings ϕ and ψ do not apply to the vertically constrained variants of the Schröder-Delannoy paths. In $\mathfrak{C}$, the $\rightarrow$ step vector has twice the horizontal displacement of the $\nearrow$ and $\searrow$ step vectors, a property that causes substitutions of the type used in ϕ and ψ to produce paths of varying length. The OEIS does not contain entries for objects equinumerous with members of the $\mathfrak{C}$ path family.

There does, however, exist a bijection to a subset of Schröder-Delannoy paths consisting of paths that are *smooth* (do not change direction) at a regular interval of 3 columns. Examples of these paths are shown in Figure 6.11. For conciseness, we shall refer to this subset of Schröder/Delannoy paths as the Smooth paths.

We define a mapping γ, illustrated in Figure 6.9, in which the following substitutions are performed on a Smooth path to either a single step or a pair of steps centered at a regular interval of every 3 columns on the integer lattice:

$$\begin{aligned} \gamma: &\quad \langle 1,1\rangle\langle 1,1\rangle &&\rightarrow \langle 0,2\rangle \\ \gamma: &\quad \langle 1,-1\rangle\langle 1,-1\rangle &&\rightarrow \langle 0,-2\rangle \\ \gamma: &\quad \langle 2,0\rangle\langle 2,0\rangle &&\rightarrow \langle 2,0\rangle \\ \gamma: &\quad \langle 2,0\rangle &&\rightarrow \text{ remove step} \end{aligned}$$

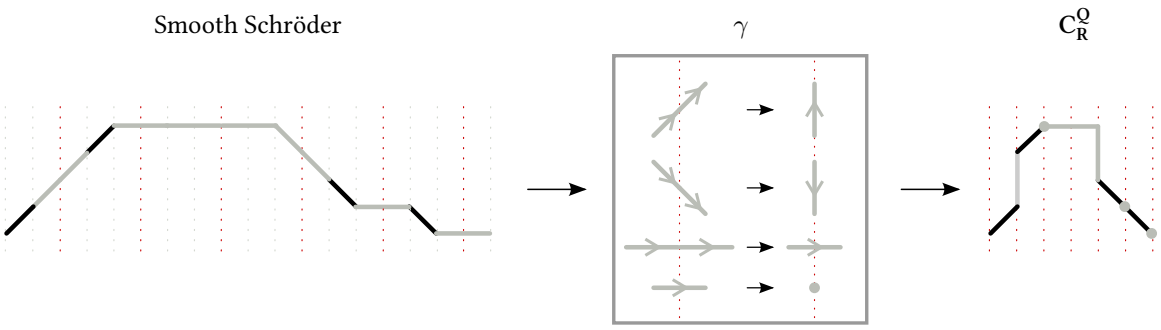


Figure 6.9: Bijective mapping from a Schröder path (length $= 18$) that is smooth at horizontal distances $3i + 2$ from the origin (indicated by red dashed lines) for $0 \leq i < 6$ to a $\mathbf{C}_\mathbf{R}^\mathbf{Q}$ path (length $= 6$).

The mapping γ is similar to ϕ but maintains a constant horizontal width by replacing vertical steps in the $\mathfrak{C}$ paths with *two* diagonal steps and replacing horizontal steps in $\mathfrak{C}$ with *two* horizontal steps.

We define the inverse mapping λ, illustrated in Figure 6.10, in which the following substitutions are performed on steps that are centered at a regular interval of every column in the integer lattice:

$$\begin{aligned}
\lambda: \quad \langle 0, 2\rangle \quad &\to \langle 1, 1\rangle\langle 1, 1\rangle \\
\lambda: \quad \langle 0, -2\rangle \quad &\to \langle 1, -1\rangle\langle 1, -1\rangle \\
\lambda: \quad \langle 2, 0\rangle \quad &\to \langle 2, 0\rangle\langle 2, 0\rangle \text{ otherwise} \\
\lambda: \quad \text{no step} \quad &\to \text{insert } \langle 2, 0\rangle
\end{aligned}$$

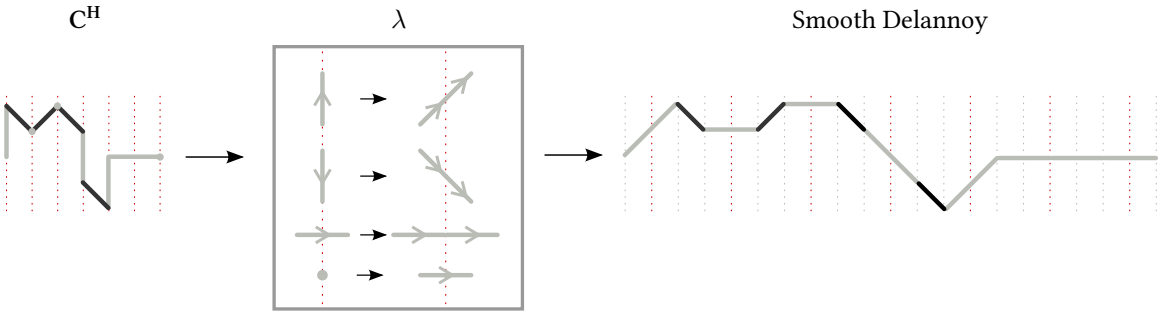


Figure 6.10: Bijective mapping from a $\mathbf{C}^\mathbf{H}$ path (length $= 6$) to a Delannoy path (length $= 20$) that is smooth at horizontal distances $3i+1$ from the origin (indicated by red dashed lines) for $0 \leq i \leq 6$.

Theorem 6.4.20. *The mapping γ is a bijection from Schröder (Delannoy) paths terminating at $(3n+2, m)$ that are smooth at each horizontal distance $3i+1$ from the origin for $0 \leq i \leq n$ to vertically constrained paths of type $\mathbf{C}^{\mathbf{Q}}$ ($\mathbf{C}^{\mathbf{H}}$) terminating at (n, m).*

Proof. We start by proving that γ applied to a Smooth Schröder (Delannoy) path always produces a vertically constrained $\mathbf{C}^{\mathbf{Q}}$ ($\mathbf{C}^{\mathbf{H}}$) path. The set of step vectors $\mathfrak{S}$ is a subset of $\mathfrak{C}$ and all substitutions in γ are from the set $\mathfrak{C}$, therefore the resulting path is made up of steps from $\mathfrak{C}$. Because there is a spacing of three columns in the pre-image between γ substitutions, there is at least one column between vertical steps in the image satisfying the non-consecutive condition of vertically constrained $\mathfrak{C}$ paths. Each γ substitution reduces the horizontal path length by 2 resulting, after $n+1$ substitutions, in a path of length n. As in the Motzkin path bijections of Section 6.3, there is no change in the height of step endvertices ensuring path images remain in the same half or quadrant as their pre-image.

Next we will prove that the vertically constrained $\mathfrak{C}$ paths and Smooth Schröder (Delannoy) paths are equinumerous. There are exactly four ways in which a Schröder path can be smooth across a column: two repeated steps of the same type ($\rightarrow\rightarrow$, $\nearrow\nearrow$, and $\searrow\searrow$) or a single horizontal step of length two. All four of these cases are uniquely mapped by γ to steps in $\mathfrak{C}$. Since each Smooth Schröder (Delannoy) path under γ produces a unique vertically constrained $\mathfrak{C}$ path, the number of vertically constrained $\mathfrak{C}$ paths is greater than or equal to the number of Smooth Schröder (Delannoy) paths. Similarly, the mapping λ applied to a vertically constrained $\mathfrak{C}$ path can be proven to always produce a unique Smooth Schröder (Delannoy) path. From this we can conclude that the number of Smooth Schröder (Delannoy) paths is greater than or equal to the number of vertically constrained $\mathfrak{C}$ paths. The two inequalities imply that the two sets of paths are equinumerous. □

Theorem 6.4.21. *The mapping γ is a bijection from Schröder (Delannoy) paths terminating at $(3n, m)$ that are smooth at a horizontal distance $3i+2$ from the origin for $0 \leq i < n$ to vertically constrained paths of type $\mathbf{C}_{\mathbf{R}}^{\mathbf{Q}}$ ($\mathbf{C}_{\mathbf{R}}^{\mathbf{H}}$) terminating at (n, m).*

Proof. The proof is similar to that of Theorem 6.4.20. The first step of the Smooth Schröder (Delannoy) path is not modified by γ therefore there can be no leading vertical step. There are n substitutions resulting in a vertically constrained path of length n. □

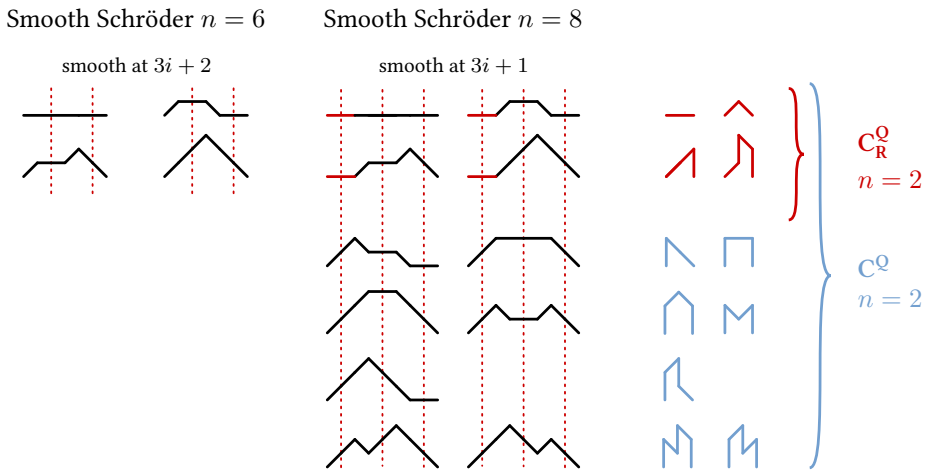


Figure 6.11: Schröder paths ($n = 6$ and $n = 8$) that are smooth at a regular interval of 3 columns and the corresponding $\mathbf{C}_{\mathbf{R}}^{\mathbf{Q}}$ and $\mathbf{C}^{\mathbf{Q}}$ paths ($n = 2$).

Generating functions for $\mathbf{C}^{\mathbf{Q}}$ and $\mathbf{C}_{\mathbf{R}}^{\mathbf{Q}}$ (restricted to the quarter plane) have been partially determined by expanding terms in the general summation using the recurrence relations.

Theorem 6.4.22. *The ordinary bivariate generating function for $C^Q(n, m)$, in terms of its first three columns, is*

$$c^Q(x, y) := \sum_{n\geq 0}\sum_{m\geq 0} C^Q(n, m)x^n y^m$$
$$= \frac{y\,(y^2{+}y^4) - x\,(1{+}xy{+}2y^2{+}y^4)\,V_0(x) - xy(1{+}xy)V_1(x) - xy^2V_2(x)}{y^3 - x\,(1{+}xy{+}y^2)\,(1{+}y^2{+}y^4)},$$

where

$$V_i(x) := \sum_{n\geq 0} C^Q(n, i)x^n, \quad i \in 0, 1, 2.$$

Theorem 6.4.23. *The ordinary bivariate generating function for $C_R^Q(n,m)$, in terms of its first three columns, is*

$$c_R^Q(x,y) := \sum_{n\geq 0}\sum_{m\geq 0} C_R^Q(n,m)x^n y^m$$
$$= \frac{y^3-x\left(1+xy+2y^2+y^4\right)W_0(x)-xy(1+xy)W_1(x)-xy^2W_2(x)}{y^3-x\left(1+xy+y^2\right)\left(1+y^2+y^4\right)},$$

where

$$W_i(x) := \sum_{n\geq 0} C_R^Q(n,i)x^n, \quad i \in 0,1,2.$$

To find the full explicit form for Theorems 6.4.22 and 6.4.23, expressions for the first three columns of the triangle are required. Based on the results for Motzkin and Dyck path extensions, it is quite likely that these six expressions are very closely related. For example, applying the same kind of logic used in Lemma 6.4.13, it is possible to show that $C^Q(n,1) = C_R^Q(n+1,0) - C^Q(n-1,0)$ from which it is possible to express $V_1(x)$ in terms of $V_0(x)$ and $W_0(x)$.

Open Problem 6.4.1. *What are the explicit forms of $V_i(x)$ for $i \in 0,1,2$?*

Open Problem 6.4.2. *What are the explicit forms of $W_i(x)$ for $i \in 0,1,2$?*

6.5 Integer Sequence Tables

Tables 6.6 through 6.17 contain enumeration results for all varieties of vertically constrained paths described in this chapter. Where available, corresponding OEIS sequences have been referenced. The values in the tables have been verified through both the generating functions and recurrence relations described in this chapter, as well as through exhaustive generation of the paths using backtracking.

$n\backslash m$	0	|1|	|2|	|3|	|4|	|5|	|6|	|7|	|8|	|9|	|10|	|11|	|12|	|13|	|14|	|15|	|16|
0	1																
1	3	2	1														
2	19	16	10	4	1												
3	141	126	90	50	21	6	1										
4	1107	1016	784	504	266	112	36	8	1								
5	8953	8350	6765	4740	2850	1452	615	210	55	10	1						
6	73789	69576	58278	43252	28314	16236	8074	3432	1221	352	78	12	1				
7	616227	585690	502593	388752	270270	168168	93093	45474	19383	7098	2184	546	105	14	1		
8	5196627	4969152	4343160	3465840	2520336	1665456	996216	536640	258570	110448	41328	13328	3620	800	136	16	1
OEIS	A082758																

Table 6.6: Number of paths terminating at point (n, m) for $\mathbf{A}_{\mathbf{R}}^{\mathbf{H}}$

$n\backslash m$	0	|1|	|2|	|3|	|4|	|5|	|6|	|7|	|8|	|9|	|10|	|11|	|12|	|13|	|14|	|15|	|16|	|17|
0	1	1																
1	7	6	3	1														
2	51	45	30	15	5	1												
3	393	357	266	161	77	28	7	1										
4	3139	2907	2304	1554	882	414	156	45	9	1								
5	25653	24068	19855	14355	9042	4917	2277	880	275	66	11	1						
6	212941	201643	171106	129844	87802	52624	27742	12727	5005	1651	442	91	13	1				
7	1787607	1704510	1477035	1161615	827190	531531	306735	157950	71955	28665	9828	2835	665	120	15	1		
8	15134931	14508939	12778152	10329336	7651632	5182008	3198312	1791426	905658	410346	165104	58276	17748	4556	952	153	17	1
OEIS	Bisect A002426																	

Table 6.7: Number of paths terminating at point (n, m) for $\mathbf{A^H}$

$n\backslash m$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1																
1	2	2	1														
2	9	12	9	4	1												
3	51	76	69	44	20	6	1										
4	323	512	518	392	230	104	35	8	1								
5	2188	3610	3915	3288	2235	1242	560	200	54	10	1						
6	15511	26324	29964	27016	20240	12804	6853	3080	1143	340	77	12	1				
7	113634	196938	232323	220584	177177	122694	73710	38376	17199	6552	2079	532	104	14	1		
8	853467	1503312	1822824	1800384	1524120	1128816	737646	426192	217242	97120	37708	12528	3484	784	135	16	1
OEIS	A026945																

Table 6.8: Number of paths terminating at point (n, m) for $\mathbf{A_R^Q}$

$n\backslash m$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	1																
1	4	5	3	1														
2	21	30	25	14	5	1												
3	127	196	189	133	70	27	7	1										
4	835	1353	1422	1140	726	369	147	44	9	1								
5	5798	9713	10813	9438	6765	4037	2002	814	264	65	11	1						
6	41835	71799	83304	77220	60060	39897	22737	11076	4563	1560	429	90	13	1				
7	310572	542895	649845	630084	520455	373581	234780	129285	62127	25830	9163	2715	650	119	15	1		
8	2356779	4179603	5126520	5147328	4453320	3390582	2292654	1381080	740554	352070	147356	53720	16796	4403	935	152	17	1
OEIS	A099250																	

Table 6.9: Number of paths terminating at point (n, m) for $\mathbf{A}^{\mathbf{Q}}$

$n\backslash m$	0	$\|1\|$	$\|2\|$	$\|3\|$	$\|4\|$	$\|5\|$	$\|6\|$	$\|7\|$	$\|8\|$	$\|9\|$	$\|10\|$	$\|11\|$	$\|12\|$	$\|13\|$	$\|14\|$	$\|15\|$	$\|16\|$	$\|17\|$
0	1																	
1	2	1	1															
2	8	6	5	2	1													
3	38	33	27	16	9	3	1											
4	196	180	150	104	65	32	14	4	1									
5	1052	990	845	635	430	251	130	55	20	5	1							
6	5774	5502	4797	3786	2721	1752	1016	516	231	86	27	6	1					
7	32146	30863	27377	22344	16793	11543	7252	4117	2107	952	378	126	35	7	1			
8	180772	174456	156900	131264	102102	73592	49064	30088	16913	8632	3976	1624	582	176	44	8	1	
OEIS	A026520																	

Table 6.10: Number of paths terminating at point (n, m) for $\mathbf{B}^{\mathbf{H}}_{\mathbf{R}}$

$n\backslash m$	0	|1|	|2|	|3|	|4|	|5|	|6|	|7|	|8|	|9|	|10|	|11|	|12|	|13|	|14|	|15|	|16|	|17|
0	1	1																
1	4	4	2	1														
2	20	19	13	8	3	1												
3	104	98	76	52	28	13	4	1										
4	556	526	434	319	201	111	50	19	5	1								
5	3032	2887	2470	1910	1316	811	436	205	80	26	6	1						
6	16778	16073	14085	11304	8259	5489	3284	1763	833	344	119	34	7	1				
7	93872	90386	80584	66514	50680	35588	22912	13476	7176	3437	1456	539	168	43	8	1		
8	529684	512128	462620	390266	306958	224758	152744	96065	55633	29521	14232	6182	2382	802	228	53	9	1
OEIS	A026520																	

Table 6.11: Number of paths terminating at point (n, m) for $\mathbf{B}^{\mathbf{H}}$

$n\backslash m$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1																	
1	1	1	1															
2	3	4	4	2	1													
3	11	17	18	13	8	3	1											
4	46	76	85	72	51	28	13	4	1									
5	207	355	415	384	300	196	110	50	19	5	1							
6	977	1716	2076	2034	1705	1236	785	430	204	80	26	6	1					
7	4769	8519	10584	10801	9541	7426	5145	3165	1729	826	343	119	34	7	1			
8	23872	43192	54798	57672	53038	43504	32151	21456	12937	7008	3394	1448	538	168	43	8	1	
OEIS	Bisect A214938																	

Table 6.12: Number of paths terminating at point (n, m) for $\mathbf{B}_{\mathbf{R}}^{\mathbf{Q}}$

$n\backslash m$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	1																
1	2	3	2	1														
2	7	11	10	7	3	1												
3	28	46	48	39	24	12	4	1										
4	122	207	233	208	151	92	45	18	5	1								
5	562	977	1154	1099	880	606	356	179	74	25	6	1						
6	2693	4769	5826	5815	4975	3726	2451	1419	714	310	112	33	7	1				
7	13288	23872	29904	30926	27768	22112	15736	10039	5720	2898	1288	496	160	42	8	1		
8	67064	121862	155662	165508	154214	128693	97111	66544	41401	23339	11850	5380	2154	749	219	52	9	
OEIS	Bisect A214938	Bisect A214938																

Table 6.13: Number of paths terminating at point (n, m) for $\mathbf{B}^{\mathbf{Q}}$

$n\backslash m$	0	$\|1\|$	$\|2\|$	$\|3\|$	$\|4\|$	$\|5\|$	$\|6\|$	$\|7\|$	$\|8\|$	$\|9\|$	$\|10\|$	$\|11\|$	$\|12\|$	$\|13\|$	$\|14\|$	$\|15\|$	$\|16\|$	$\|17\|$
0	1																	
1		2		1														
2	11		9		4		1											
3		58		41		20		6		1								
4	343		300		200		99		35		8		1					
5		1943		1561		1000		503		193		54		10		1		
6	11837		10794		8167		5094		2588		1051		331		77		12	
7		69670		59357		42968		26278		13453		5686		1944		520		102
8	430819		401490		324653		227151		136849		70470		30692		11136		3277	

Table 6.14: Number of paths terminating at point (n, m) for $\mathbf{C}_{\mathbf{R}}^{\mathbf{H}}$

$n\backslash m$	0	|1|	|2|	|3|	|4|	|5|	|6|	|7|	|8|	|9|	|10|	|11|	|12|	|13|	|14|	|15|	|16|	|17|
0	1		1															
1		5		3		1												
2	29		24		14		5		1									
3		157		119		67		27		7		1						
4	943		843		599		334		142		44		9		1			
5		5447		4504		3064		1696		750		257		65		11		1
6	33425		30798		24055		15849		8733		3970		1459		420		90	
7		198697		171995		128603		82699		45417		21083		8151		2556		612
8	1233799		1156962		953294		688653		434470		238012		112290		45078		14997	

Table 6.15: Number of paths terminating at point (n, m) for $\mathbf{C^H}$

$n\backslash m$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1																	
1		1		1														
2	4		5		3		1											
3		19		23		14		5		1								
4	70		110		107		66		27		7		1					
5		439		626		525		320		141		44		9		1		
6	1684		2942		3344		2657		1591		734		256		65		11	
7		11977		18503		18006		13560		8068		3830		1441		418		88
8	47083		86936		107498		97295		70074		41378		20083		7968		2529	

Table 6.16: Number of paths terminating at point (n, m) for $\mathbf{C_R^Q}$

$n\backslash m$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1		1															
1		3		3		1												
2	11		15		12		5		1									
3		59		77		54		25		7		1						
4	221		364		378		260		125		42		9		1			
5		1463		2155		1921		1276		636		236		63		11		1
6	5666		10120		11893		9948		6408		3259		1297		395		88	
7		41417		65179		65562		51697		32727		16850		7041		2338		583
8	163799		306433		386557		360673		270875		168856		87798		37971		13366	

Table 6.17: Number of paths terminating at point (n, m) for $\mathbf{C}^{\mathbf{Q}}$

Chapter 7

Symmetry

As demonstrated by the results in Section 5.2, it is clear that the number of tesselace embeddings grows at least exponentially with the number of vertices. For a 4×5 period rectangle, we have generated over 5 million tesselace embeddings - enough to keep an army of lacemakers employed for several life times. It would be nice if we could identify a short list of patterns to be considered first.

In addition, despite the large number of patterns identified, there are some real gems that were not discovered. For example, the traditional pattern shown in Figure 2.4 has 74 vertices and was not found by the lattice path algorithm, which becomes intractable for graphs with more than 20 vertices.

Is there a way that we can address both of these issues? A way to pull out patterns of interest and simultaneously extend our search space to much larger graph sizes? The answer is yes, by taking advantage of symmetry, both for its aesthetic appeal and its ability to condense information.

7.1 Isometries of the Euclidean Plane

In Section 3.2 we introduced the idea of symmetry as set of isometric transformations that map a set of tiles onto itself. All tesselace embeddings are periodic, meaning that they possess the symmetry of translation in two non-parallel directions. In this chapter we will look at symmetries in addition to translation, namely, rotation, reflection and glide reflection. The group actions of rotation and reflection are quite familiar and need no further explanation. A glide reflection is less common so we shall provide a definition.

Definition (*Glide reflection*). A glide reflection is a reflection about a line L combined with a translation through a distance d parallel to L. We shall refer to L as the glide line. [32, p. 27]

A surprising result in the application of symmetry to a set of points in the Euclidean plane is that only a finite number of rotation operations are allowed.

Lemma 7.1.1 (Crystallographic restriction). *A rotation contained in a planar periodic pattern can only be of order 1,2,3,4, or 6 [54, p. 63].*

Of particular note is the absence of 5-fold rotations.

The restricted number of rotation types gives rise to the following theorem:

Theorem 7.1.2 (Fedorov-Schönflies Theorem). *There are exactly 17 distinct symmetry groups in planar periodic patterns.*

This result was first discovered by Fedorov and Shönflies in 1891 and rediscovered by Pólya in 1924. A detailed proof with a case-by-case analysis is presented by Lyndon in [54, p. 74]. An approachable and aesthetically pleasing discussion of the proof and its implications is given by Conway et al. in [16]. For a summary of the isometries of each symmetry group, refer to [32, Tab. 1.4.2].

Several notations exist for labelling the planar symmetry groups. We will use the orbifold notation of Conway [16] which provides a topological description of the symmetry. The set of all points that are the same under a symmetry operation is called an *orbit*. An *orbifold* is "folded" by identifying all points in the same orbit.

In orbifold notation, unique rotational (or gyrational, to uses Conway's term) symmetries are listed first by specifying the number of repetitions required to rotate a point back to its original position. The point group $n\bullet$ (which corresponds to C_n in Schönflies notation) is the group of rotations σ such that $\sigma^n = 1$, where 1 is the identity and n is the smallest number of rotations that returns a point to its original position. Reflection (or kaleidoscope) symmetries are preceded by a $*$ and are represented by the number of mirror lines that meet at a point. The point group $*n\bullet$ (D_n in Schönflies notation) is the dihedral group of order n. A single mirror line is represented by a solo $*$; a glide line is indicated by $\times$. For example, $*632$ indicates a pattern with three distinct reflection point subgroups of types $*6\bullet$, $*3\bullet$ and $*2\bullet$; 2222 indicates four distinct reflection point subgroups each of type $2\bullet$; $4*2$ represents a mix of rotations and reflections, namely, $4\bullet$ and $*2\bullet$; and $*\times$ indicates a mirror reflection and a glide reflection.

It is apparent from looking at traditional bobbin lace patterns that symmetry plays a large role in its design. Mirror and glide reflections feature prominently. In Section 7.4 we will prove by demonstration the following theorem:

Theorem 7.1.3. *There exist tesselace embeddings for each of the seventeen periodic symmetry groups in the plane.*

7.2 Preliminaries

The exhaustive enumeration of tesselace embeddings according to their symmetry group produces some incredibly beautiful results, however, the algorithm requires careful attention to many details on a case-by-case basis. In this section we will outline the properties that must considered and their specific details for each of the 16 non-trivial planar periodic symmetry groups.

As in Chapter 5, we will follow the traditional practices of lace design and look for tesselace patterns on a lattice. Because we are considering symmetry in this chapter, we will be more specific about the geometry of the lattice. We will restrict our attention to two lattice types: the hexagonal and square lattices. Other lattice types have been identified but hexagonal and square lattices are sufficient to cover all 17 planar periodic symmetry groups. Exploration of other types of lattices could be carried out in the same way.

In the case of the hexagonal lattice we will actually be working with a rectangular lattice created from two copies of the hexagonal lattice; the second copy is translated to the right by the width of a column and superimposed on the first as shown in Figure 7.2(a).

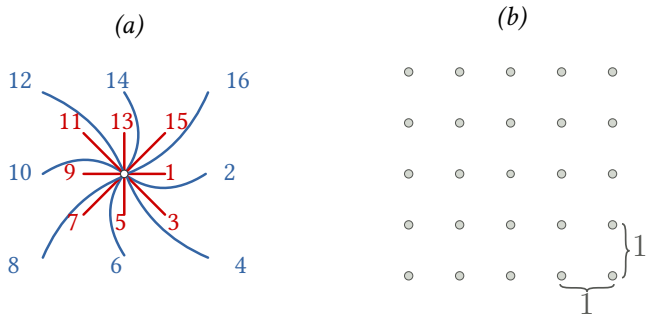


Figure 7.1: Square lattice: (a) Steps vectors (b) Lattice points

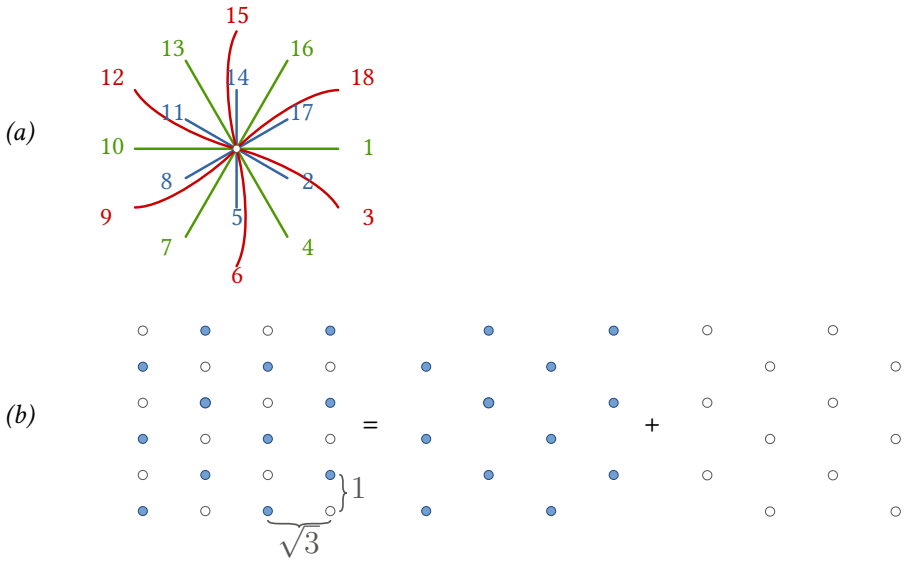


Figure 7.2: Hexagonal lattice: (a) Steps vectors (b) Lattice points

The edges of a tesselace embedding are oriented and from this orientation a partial order can be assigned to the vertices. If we apply symmetry generators to the directed edges of a tesselace embedding, the only transformations allowed are reflection and glide along a vertical line. All other symmetry generators will produce contractible cycles. The effect of edge orientation on the appearance of the finished lace depends on the ζ mapping used. In most cases, it is possible to choose cross and twist combinations that closely follow the symmetry of the edges independent of orientation. For the purpose

of exploring as many of the 17 wallpaper groups as possible, we have chosen to apply symmetry transformations to the underlying undirected edges of the graph associated with the tesselace embedding and then determine whether there exists an orientation of the edges that will produce a workable pattern.

In a tesselace embedding, vertices of the graph are mapped to lattice points. As a result, transforming an edge of a tesselace embedding is permissable only if it takes the endvertices of an edge from one pair of lattice points to another pair of lattice points. There are a finite number of locations to place a particular symmetry group generator in order to comply with this condition. Let Γ be the set of 17 periodic planar symmetry groups. For each $\gamma \in \Gamma$, there is a commonly used set of generators [18]. In order to manipulate edges on a lattice according to the symmetry generators of a group, we will assign a fixed position to each generator in the plane using the triple (σ, θ, δ). Here σ is a group generator. If the generator has an axis, θ is the clockwise angle between that axis and a horizontal line, otherwise θ is null. If the generator belongs to a point group, δ is the location of the center of the point group (not necessarily a lattice point). For a single mirror or glide reflection, δ is a point on the mirror or glide line.

An exhaustive list of the allowed orientations and locations for the symmetry generators of Γ is given in Figures 7.3 to 7.8. The locations are shown relative to the neighbouring vertices. The dashed lines in these figures show how an example edge would be successfully transformed on the lattice; the grey dashed lines are transformed copies of the red edge. The significance of the red vertex in these figures will be explained later on in this section. At the bottom of the figures we give some examples of failed generator positions. These failure cases are presented merely as examples and should not be considered an exhaustive list.

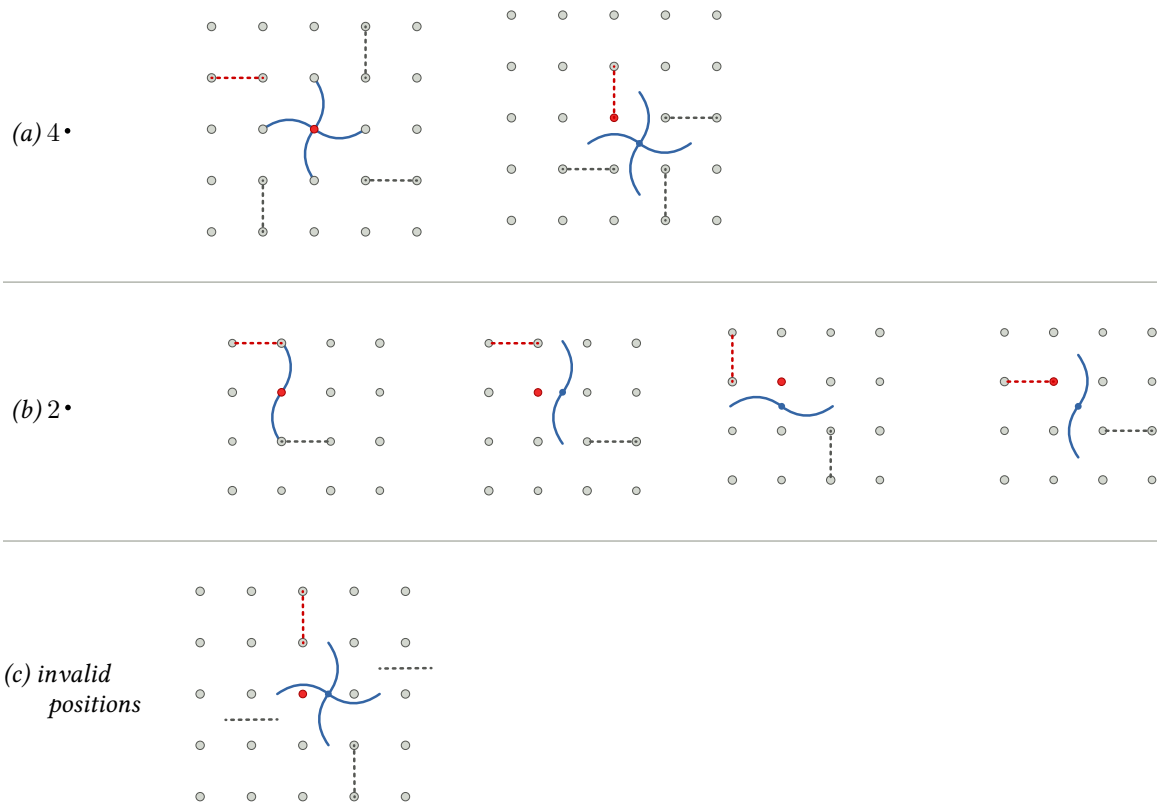


Figure 7.3: Square lattice: (a) and (b) All possible locations for the centre of a $n\cdot$ point group where $n = 2$ or 4 respectively (c) Example of invalid locations for center of $n\cdot$. Gray dashed line segments are the images of the red dashed line segments under rotation by $n\cdot$. Red dot is the lattice position used as root of lexicographic labels.

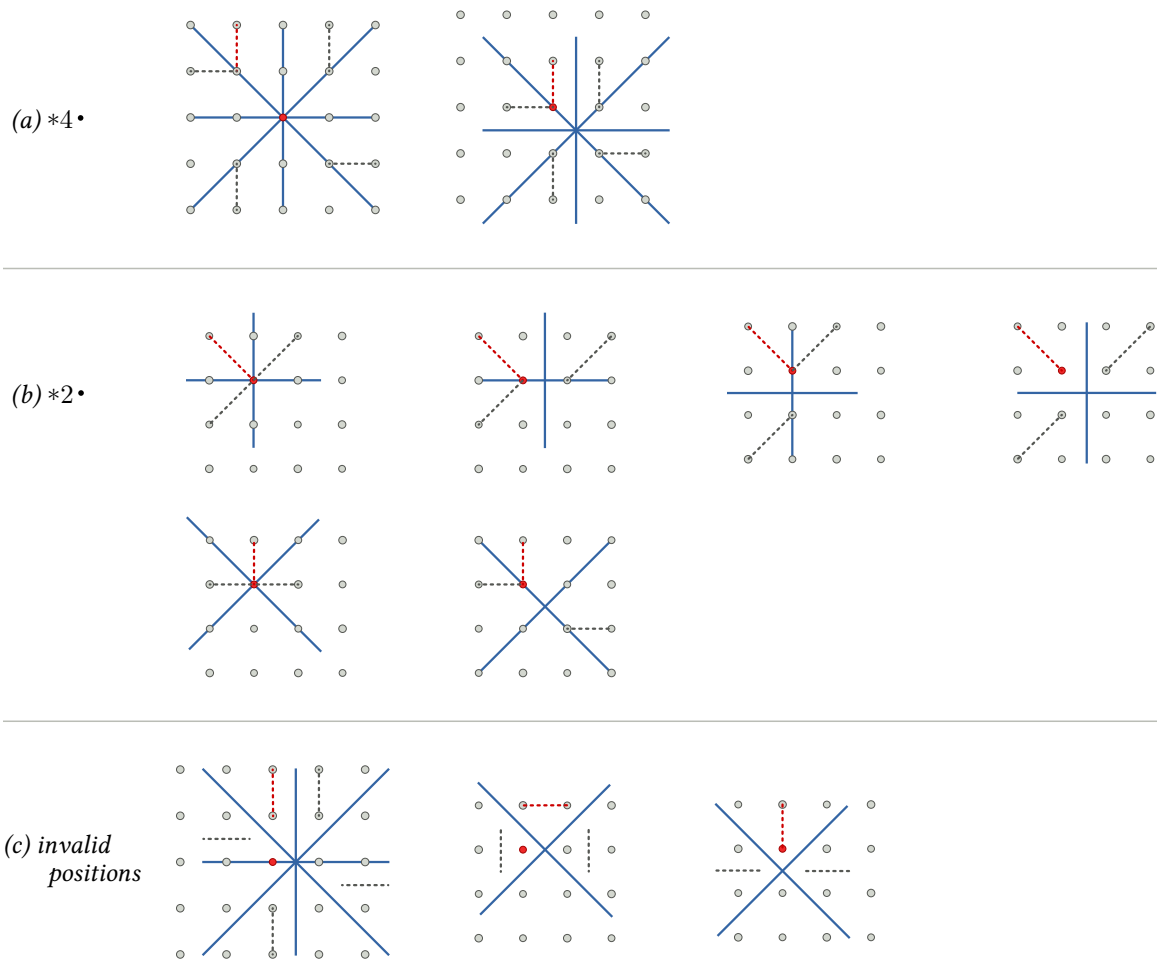


Figure 7.4: Square lattice: (a) and (b) All possible locations for the centre of a $*n\cdot$ point group where $n = 2$ or 4 respectively. The mirror lines of $*2\cdot$ can be vertical/horizontal or diagonal. (c) Examples of invalid locations for center of $*n\cdot$. Blue lines are mirror lines. Gray dashed line segments are the images of the red dashed line segments under reflection by $*n\cdot$. Red dot is the lattice position used as root of lexicographic labels.

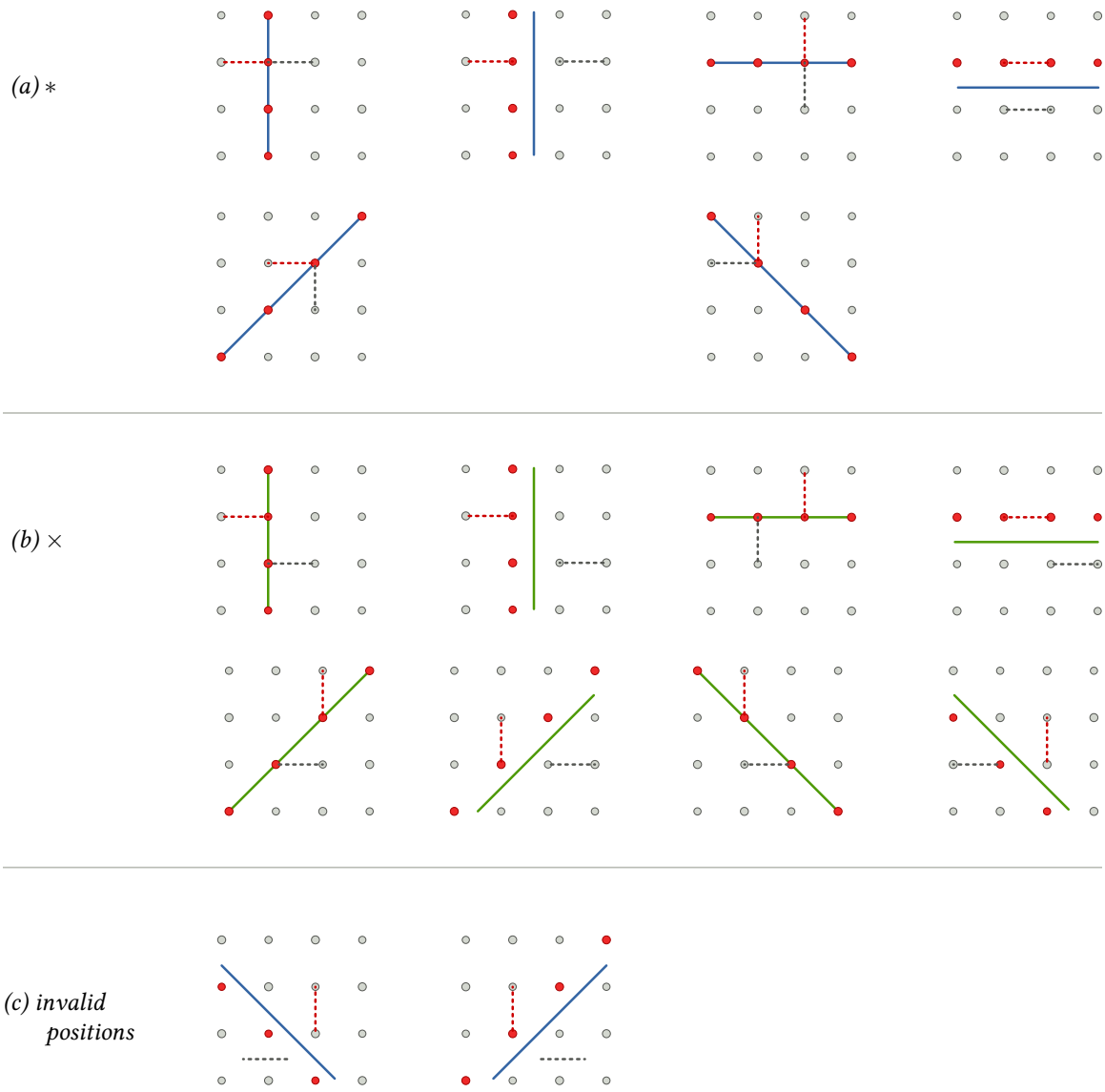


Figure 7.5: Square lattice: All possible locations for (a) mirror lines and (b) glide lines. The respective lines can be at an angle of $0°$, $45°$, $90°$ or $135°$ from horizontal. (c) Examples of invalid locations for mirror and glide lines. Blue lines are mirror lines. Green lines are glides lines. Gray dashed line segments are the images of the red dashed line segments under $*$ or $\times$. Red dots are the set of lattice positions used as roots of lexicographic labels.

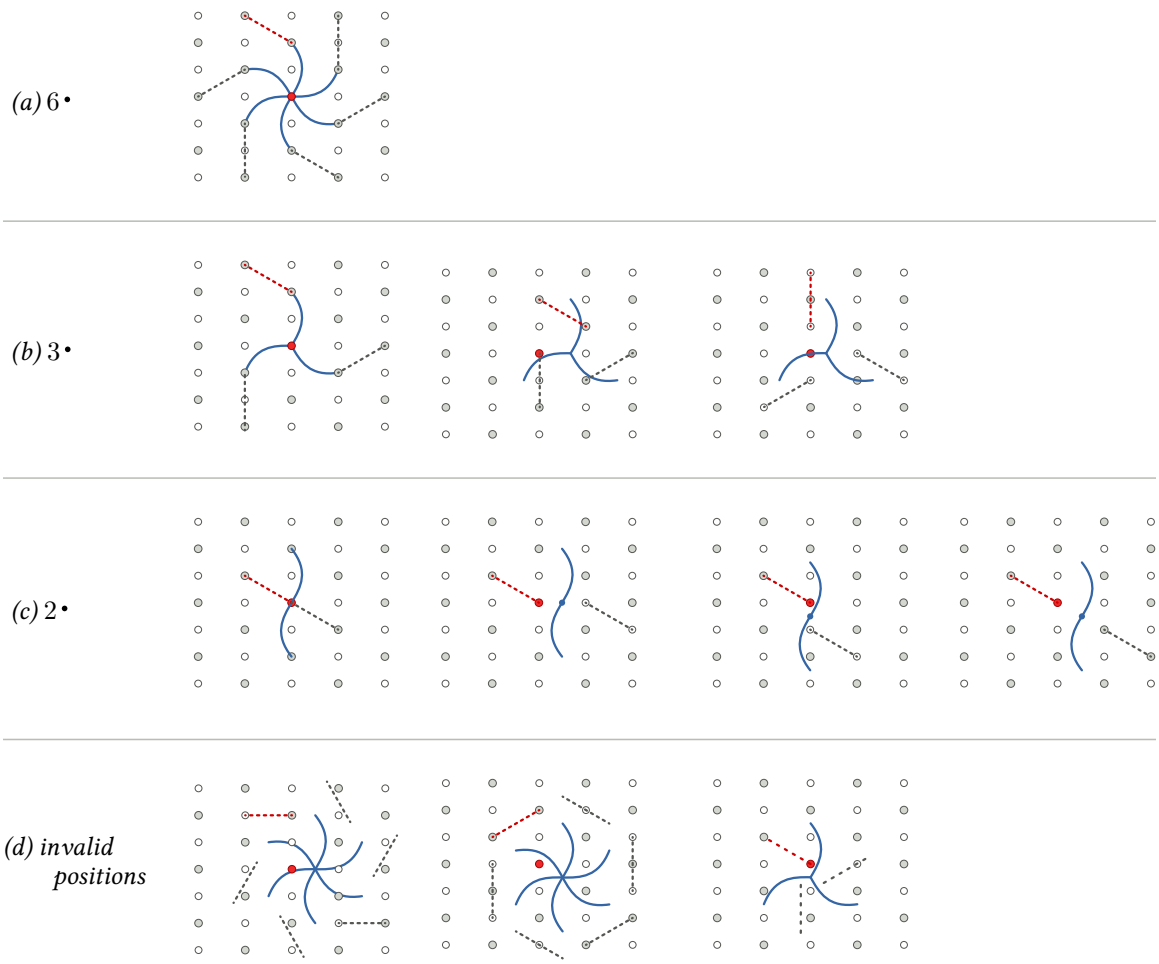


Figure 7.6: Hexagonal lattice: (a),(b) and (c) All possible locations for the centre of a $n\cdot$ point group where $n = 6, 3$ or 2 respectively. (d) Examples of invalid locations for center of $n\cdot$. Gray dashed line segments are the images of the red dashed line segments under rotation by $n\cdot$. Red dot is the lattice position used as root of lexicographic labels.

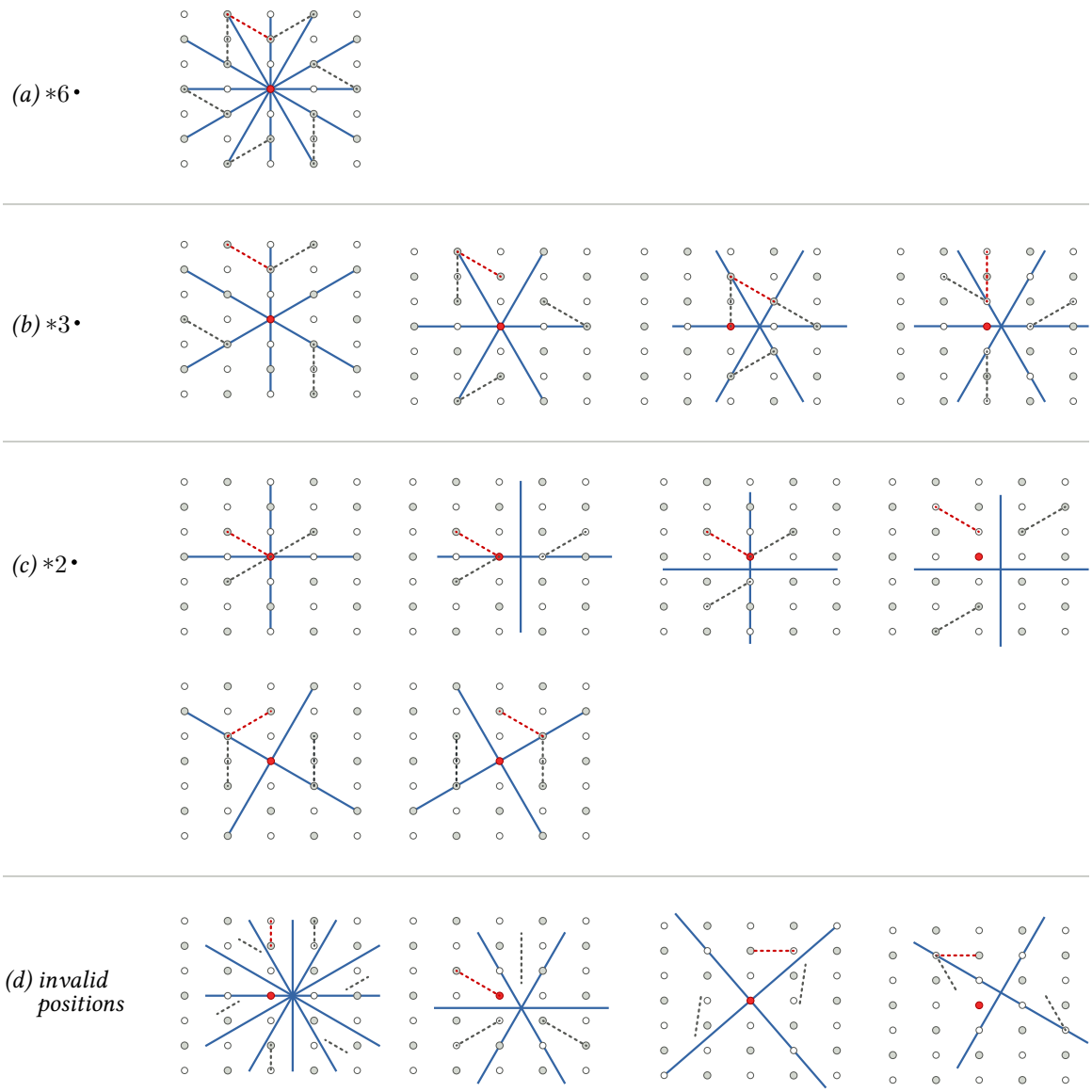


Figure 7.7: Hexagonal lattice: (a), (b) and (c) All possible locations for the centre of a $*n\cdot$ point group where $n = 6, 3$ or 2 respectively. (d) Examples of invalid locations for center of $*n\cdot$. For $*3\cdot$ the set or mirror lines may include either a vertical or a horizontal line. Blue lines are mirror lines. Gray dashed line segments are the images of the red dashed line segments under reflection by $*n\cdot$. Red dot is the lattice position used as root of lexicographic labels.

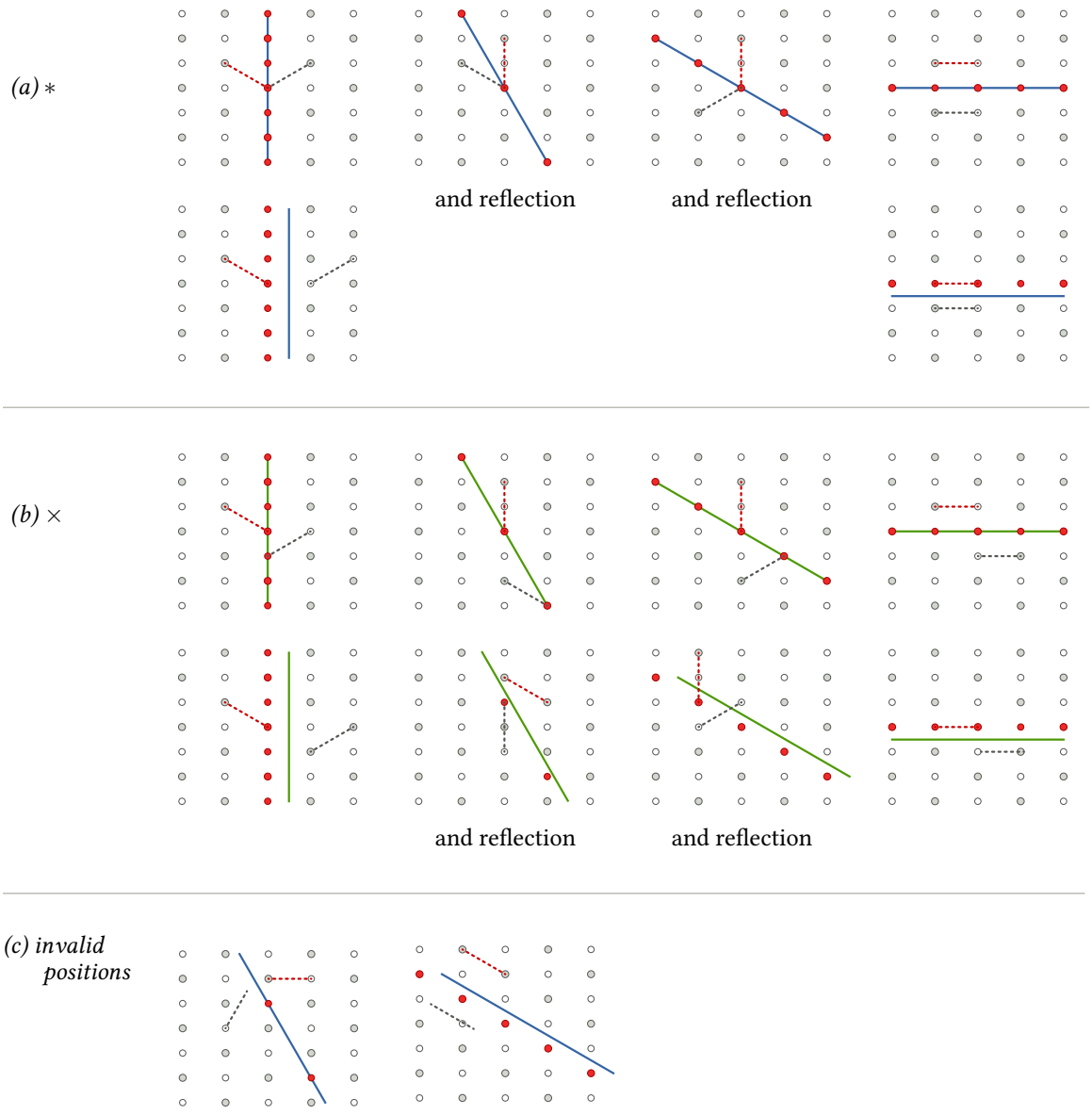


Figure 7.8: Hexagonal lattice: All possible locations for (a) mirror lines and (b) glide lines. Angles of $0°$, $30°$, $60°$ and $90°$ from horizontal are allowed. (c) Examples of invalid locations for mirror and glide lines. Blue lines are mirror lines. Green lines are glides lines. Gray dashed line segments are the images of the red dashed line segments under $*$ or $\times$. Red dots are the set of lattice positions used as roots of lexicographic labels.

To apply γ to line segments on the lattice, we need to specify the complete set of generators for γ along with their position and orientation. We shall represent a configuration of the generators of γ as the set $R(\gamma) = \{(\sigma_1, \theta_1, \delta_1), ..., (\sigma_k, \theta_k, \delta_k)\}$, where each of the generators used to define γ appears in R exactly once. The configuration $R(\gamma)$ is not unique so, for completeness, we must specify all valid combinations which we shall represent as $\mathcal{R}(\gamma) = \{R_1, R_2, ..., R_r\}$.

In the following discussion, a lattice point (col, row) is *near* a point (x, y) if $(col, row) = (\lfloor x \rfloor, \lfloor y \rfloor)$. Similarly, a lattice point (col, row) is near a non-horizontal line l if given a point (x, row) on l then $col = \lfloor x \rfloor$. If l is horizontal then given a point (col, y) on l, a lattice point (col, row) is near l if $row = \lfloor y \rfloor$.

For improved performance, we shall restrict the elements of $\mathcal{R}(\gamma)$ to configurations that generate the least number of graphs from the same isomorphism class. This is done by choosing the least common, ideally unique, generator from the symmetry group γ and placing it at or near the lattice point $(0, 0)$.

In cases where there is a single unique point group generator, such as $*2\bullet$ in $*442$, the choice is clear. In cases where there are multiple unique point group generators, such as $*6\bullet$, $*3\bullet$ and $*2\bullet$ in $*632$, we choose the generator with the least number of valid positions on the lattice; in the case of $*632$ this is $*6\bullet$ as shown in Figure 7.7(b). In cases where there are multiple point group generators of the same type, such as 2222, each one of the generators is placed at or near $(0, 0)$ in turn and lexicographic comparison of the corresponding labels is used to find the representative of the class. For 2222, this is a comparison of four different labels which is still a significant gain over the $n \times m$ comparisons required in the lattice path approach. In Figures 7.3, 7.4, 7.6, and 7.7, the lattice point near the center of a point symmetry group is indicated by a red dot.

For symmetry groups with single mirror reflections or glide reflections, we must compare the labels rooted at any lattice point $a \in A$ where A is the set of lattice points in the period rectangle that intersect the mirror or glide line l. If l does not intersect any lattice points, then A is the set of lattice points in the period rectangle that are near l. In Figures 7.5 and 7.8, lattice points in A are indicated by red dots.

For clarity, let us walk through an example. Consider $\gamma = *442$. Because the $*2^\bullet$ generator is unique it serves as the anchor for $\mathcal{R}$. As shown in Figure 7.4(b), there are six possible placements for $*2^\bullet$ so the size of $\mathcal{R}$ is 6; i.e. $|\mathcal{R}(*442)| = 6$. The $*442$ symmetry group always forms a square as shown in Figure 7.16. Therefore, on an $n \times n$ grid, we have the following exhaustive set of configurations:

$$
\begin{aligned}
R_1(*442) &= \{(*2^\bullet, 0^\circ, (0,0)), \quad (*4^\bullet, \varnothing, (0, \tfrac{n}{2})), \quad (*4^\bullet, \varnothing, (\tfrac{n}{2}, 0))\},\\
R_2(*442) &= \{(*2^\bullet, 0^\circ, (\tfrac{1}{2}, 0)), \quad (*4^\bullet, \varnothing, (\tfrac{1}{2}, \tfrac{n}{2})), \quad (*4^\bullet, \varnothing, (\tfrac{n+1}{2}, 0))\},\\
R_3(*442) &= \{(*2^\bullet, 0^\circ, (0, \tfrac{1}{2})), \quad (*4^\bullet, \varnothing, (0, \tfrac{n+1}{2})), \quad (*4^\bullet, \varnothing, (\tfrac{n}{2}, \tfrac{1}{2}))\},\\
R_4(*442) &= \{(*2^\bullet, 0^\circ, (\tfrac{1}{2}, \tfrac{1}{2})), \quad (*4^\bullet, \varnothing, (\tfrac{1}{2}, \tfrac{n+1}{2})), \quad (*4^\bullet, \varnothing, (\tfrac{n+1}{2}, \tfrac{1}{2}))\},\\
R_5(*442) &= \{(*2^\bullet, 45^\circ, (0,0)), \quad (*4^\bullet, \varnothing, (\tfrac{n}{4}, \tfrac{n}{4})), \quad (*4^\bullet, \varnothing, (\tfrac{n}{4}, \tfrac{3n}{4}))\},\\
R_6(*442) &= \{(*2^\bullet, 45^\circ, (\tfrac{1}{2}, \tfrac{1}{2})), \quad (*4^\bullet, \varnothing, (\tfrac{n+2}{4}, \tfrac{n+2}{4})), \quad (*4^\bullet, \varnothing, (\tfrac{n+2}{4}, \tfrac{3n+2}{4}))\}.
\end{aligned}
$$

To generate tesselace patterns with the symmetry of a group γ, we start by positioning the symmetry generators of γ on the lattice as described above. As each edge is added to the graph, copies of the edge are transformed by the generators thereby creating the desired symmetry. But that is not the complete story. In addition to the symmetry that we are looking for, it is possible for additional types of symmetry to creep in, changing the pattern from one planar periodic symmetry group to another. For example, when generating a 632 pattern, in addition to the desired rotation symmetries, it is possible for mirror reflections to appear in the graph, resulting in a $*632$ pattern. Whenever there is the possibility of additional, unwanted group structure, it is necessary to test that a potential solution contains only the desired symmetries.

Let us look at all the scenarios where additional unwanted symmetry might pop up. There is a subgroup relationship between a cyclic symmetry group of order n and any cyclic group of order m where $n \backslash m$. With respect to the 17 planar periodic symmetry groups, the cyclic subgroup relationships of interest are $2^\bullet < 4^\bullet$, $2^\bullet < 6^\bullet$ and, $3^\bullet < 6^\bullet$. The same divisibility property and subgroup structure apply between dihedral groups. Further, $n^\bullet$ is a subgroup of the dihedral symmetry group of order n so we have $n^\bullet < *n^\bullet$. Between glide and mirror reflections there is also a relationship: a mirror reflection is equivalent to a glide reflection with a translation distance of 0.

Tables 7.1 to 7.16 provide information specific to each of the 16 non-trivial symmetry groups in Γ for the following categories:

- *Lattice* - Bobbin lace patterns are designed on a regular grid which can be either a $45°$ square lattice or a $60°$ hexagonal lattice.
- *Generators* - Identify the minimum set of symmetry generators required to create patterns of the desired symmetry group.
- *Orientation* - List of all ways in which a symmetry group can be oriented on the grid.
- *Placement* - Describe all ways in which the generators can be positioned relative to the lattice points of the grid.
- *Label* - Describe the criteria used to generate a unique label for each class of patterns within a given symmetry group.
- *Tests* - List the minimum set of accidental symmetries for which we must test in order to ensure patterns are of the desired symmetry group.

Lattice:	Hexagonal lattice
Generators:	$*6\bullet$, $*3\bullet$ and $*2\bullet$ centers
Orientation:	All three generator centers align on a vertical *or* horizontal mirror line
Placement:	$*6\bullet$ center must be on a lattice point
Label:	Root is $*6\bullet$ center
Tests:	None

Table 7.1: Configuration details for $*632$

Lattice:	Hexagonal lattice
Generators:	$6\bullet$, $3\bullet$ and $2\bullet$ centers
Orientation:	All three generator centers align on a vertical *or* horizontal line
Placement:	$6\bullet$ center must be on a lattice point
Label:	Root is $6\bullet$ center
Tests:	Six-fold rotation center has no reflections; i.e., it is $6\bullet$ and not $*6\bullet$

Table 7.2: Configuration details for 632

Lattice:	Hexagonal lattice
Generators:	Three distinct $*3\bullet$ centers
Orientation:	All three centers align on a vertical *or* horizontal mirror line
Placement:	Vertical orientation: $*3\bullet$ must be on a lattice point. Horizontal orientation: $*3\bullet$ center can be on lattice point *or* row aligned and $1/3$ or $2/3$ of distance between columns as shown in Figure 7.7.
Label:	Root is lexicographically least of three $*3\bullet$ centers
Tests:	No pairs of $*3\bullet$ centers are isomorphic

Table 7.3: Configuration details for $*333$

Lattice:	Hexagonal lattice
Generators:	Three distinct $3\bullet$ centers
Orientation:	Three $3\bullet$ centers align on a vertical *or* horizontal line
Placement:	Vertical orientation: $3\bullet$ must be on a lattice point. Horizontal orientation: $3\bullet$ center can be on lattice point *or* row aligned and $1/3$ or $2/3$ of distance between columns.
Label:	Root is lexicographically least of three $3\bullet$ centers
Tests:	No pairs of $3\bullet$ centers are isomorphic. At least two of the three $3\bullet$ centers are not also centers of $*3\bullet$.

Table 7.4: Configuration details for 333

Lattice:	Hexagonal lattice
Generators:	One $*3\bullet$ center and one $3\bullet$ center
Orientation:	Two $3\bullet$ centers align on a vertical *or* horizontal axis
Placement:	Vertical orientation: $*3\bullet$ must be on a lattice point. Horizontal orientation: $*3\bullet$ center can be on lattice point *or* row aligned and $1/3$ or $2/3$ of distance between columns.
Label:	Root is $*3\bullet$ center
Tests:	Three-fold rotational symmetry has no reflections; i.e. it is a $3\bullet$ and not a $*3\bullet$ center

Table 7.5: Configuration details for $3*3$

Lattice:	Square lattice
Generators:	Two $*4\bullet$ centers and one $*2\bullet$ center
Orientation:	Axes of $*2\bullet$ center can be vertical/horizontal *or* two diagonals
Placement:	$*2\bullet$ center can be placed on a lattice point *or* row aligned and half way between columns *or* column aligned and half way between rows *or* midway between two rows and two columns.
Label:	Root is $*2\bullet$ center
Tests:	None

Table 7.6: Configuration details for $*442$

Lattice:	Square lattice
Generators:	Two $4\bullet$ centers and one $2\bullet$ center
Orientation:	$2\bullet$ and $4\bullet$ centers can both lie on vertical/horizontal lines *or* on two diagonal lines
Placement:	$2\bullet$ center can be place on a lattice point *or* row aligned and half way between columns *or* column aligned and half way between rows *or* midway between two rows and two columns.
Label:	Root is $2\bullet$ center
Tests:	Two-fold rotational symmetry has no reflections; i.e. it is a $2\bullet$ and not a $*2\bullet$ center

Table 7.7: Configuration details for 442

Lattice:	Square lattice
Generators:	One $4\bullet$ center and one $*2\bullet$ center
Orientation:	Axes of $*2\bullet$ center can be vertical/horizontal *or* two diagonals
Placement:	$4\bullet$ center can be place on a lattice point *or* midway between two rows and two columns.
Label:	Root is $4\bullet$ center
Tests:	Four-fold rotational symmetry has no reflections; i.e. it is a $4\bullet$ and not a $*4\bullet$ center

Table 7.8: Configuration details for $4*2$

Lattice:	Square and hexagonal lattices
Generators:	Four $*2\bullet$ centers
Orientation:	Mirrors are vertical/horizontal *or* mirrors are diagonal
Placement:	$*2\bullet$ centers can be on a lattice point *or* half way between lattice points *or* row aligned and halfway between columns *or* column aligned and halfway between rows. The first center is unconstrained (all 4 positions possible). Two remaining centers have one degree of freedom (2 positions each). Final center is complete specified by previous three.
Label:	Root is lexicographically least of four possible $*2\bullet$ centers
Tests:	No pairs of $*2\bullet$ centers are isomorphic. At least three out of four rotation centers are not $*4\bullet$.

Table 7.9: Configuration details for $*2222$

Lattice:	Square and hexagonal lattices
Generators:	One $2\bullet$ center and two $*2\bullet$ centers
Orientation:	Mirrors are perpendicular to one another and can be vertical/horizontal *or* along the two diagonals
Placement:	$2\bullet$ center can be on a lattice point *or* half way between lattice points *or* row aligned and halfway between columns *or* column aligned and halfway between rows
Label:	Root is $2\bullet$ center
Tests:	Two fold rotation is not a $*2\bullet$ center

Table 7.10: Configuration details for $2*22$

Lattice:	Square and hexagonal lattices
Generators:	Two $2\bullet$ centers and one mirror
Orientation:	Mirror can be vertical *or* horizontal *or* one of the diagonals
Placement:	$2\bullet$ centers can be on a lattice point *or* half way between lattice points *or* row aligned and halfway between columns *or* column aligned and halfway between rows. Placement of the mirror is parallel to the line between two $2\bullet$ centers.
Label:	Root is lexicographically least of two possible $2\bullet$ centers
Tests:	Two $2\bullet$ centers are not isomorphic. Two-fold rotational symmetries have no reflections; i.e. they are $2\bullet$ centers and not $*2\bullet$ centers.

Table 7.11: Configuration details for $22*$

Lattice:	Square and hexagonal lattices
Generators:	Two $2\bullet$ centers and glide reflections in two perpendicular directions
Orientation:	Glide lines can be vertical and horizontal *or* along the two diagonals
Placement:	$2\bullet$ centers can be on a lattice point *or* half way between lattice points *or* row aligned and halfway between columns *or* column aligned and halfway between rows. Glide line can be through vertices *or* midway between them.
Label:	Root is lexicographically least of two possible $2\bullet$ centers
Tests:	Two $2\bullet$ centers are not isomorphic. Glide reflection has a nonzero translation distance. Two-fold rotation centers are not $*2\bullet$.

Table 7.12: Configuration details for $22\times$

Lattice:	Square and hexagonal lattices
Generators:	Two parallel single mirrors
Orientation:	Vertical *or* horizontal *or* diagonal
Placement:	Mirrors can pass through a lattice point *or* halfway between adjacent lattice points. Two mirror lines are independent.
Label:	Root is lexicographically least vertex along either one of the two mirror lines
Tests:	Mirrors are not isomorphic. Points along mirror lines are not $*2^\bullet$ or higher.

Table 7.13: Configuration details for $**$

Lattice:	Square and hexagonal lattices
Generators:	Two parallel glide reflections
Orientation:	Vertical *or* horizontal *or* diagonal
Placement:	Glide lines can pass through a lattice point *or* halfway between adjacent lattice points. Positions of the two glide lines are independent.
Label:	Root is lexicographically least vertex along either one of the two glide lines
Tests:	Glide reflections are not isomorphic. Each glide reflection includes a non-zero translation.

Table 7.14: Configuration details for $\times\times$

Lattice:	Square and hexagonal lattices
Generators:	A mirror and a parallel glide reflection
Orientation:	Vertical, horizontal, or diagonal
Placement:	Mirror line and glide line can pass through a vertex or midway between vertices, the two are independent of each other
Label:	Root is lexicographically least vertex along the glide line
Tests:	No points along mirror line are $*2^\bullet$ or higher. Glide reflection includes a non-zero translation distance.

Table 7.15: Configuration details for $*\times$

Lattice:	Square and hexagonal lattices
Generators:	Four different $2^\bullet$ centers
Orientation:	Form a parallelogram with smallest interior angle of $30, 45, 60 or 90$
Placement:	$2^\bullet$ centers can be on a lattice point *or* half way between lattice points *or* row aligned and halfway between columns *or* column aligned and halfway between rows. Position of three centers independent.
Label:	Root is lexicographically least of four centers
Tests:	2-fold centers are not $*2^\bullet$, all four centers are non-isomorphic

Table 7.16: Configuration details for 2222

7.3 Algorithm

The algorithm can be broken into two parts: For each $\gamma \in \Gamma$ (the set of 17 periodic planar symmetry groups), (1) Exhaustively generate topological embeddings of 4-regular undirected graphs on the torus such that for each embedding the vertices are restricted to lattice points, the associated combinatorial embedding has a genus one and curves associated with the edges correspond to the symmetry of γ, and (2) Assign an orientation to the undirected graph and determine whether the result adheres to all of the properties of a tesselace embedding. In our enumeration results, found in Section 7.4, we include at most one digraph for each undirected graph. In some cases, multiple non-isomorphic digraphs exist but only one representative was included in the count.

As in Chapter 5, we will use backtracking to enumerate and generate solutions. For the first stage, generating undirected graph embeddings with symmetry, the solution set is the representation of an undirected toroidal embedding $\Delta_1(G^\tau)$ given by an $n \times m$ toroidal array in which each array element is the label of a vertex at lattice position (i, j). The domain is a set of configurations $D = \{x_1, x_2, ...\}$ in which each configuration x_i consists of 4 edges incident to a vertex v. The far endvertex of an edge in x_i must be the image of

v under the finite set of step vectors specified in Figure 7.1(a) for the square lattice and Figure 7.2(a) for the hexagonal lattice. The edges in a configuration can only intersect at an endvertex; care must be taken when there exist two step vectors with the same angle but different length.

The property $P_\ell(x_1, ..., x_l)$ is defined by the following rules:

Rules 7.3.1.

1. *Each vertex has degree* ≤ 4.
2. *Edges only intersect at endvertices.*
3. *Vertices are mapped to lattice points.*
4. *All transformed copies of an edge under symmetry group* γ *are present in the embedding.*

The backtracking algorithm loops over all $R \in \mathcal{R}$ and recursively adds configurations to the graph in a row-by-row order. When a solution is output from the backtracking algorithm, it is tested to ensure that it is prime, connected, the lexicographically least representative of its isomorphism class and the associated combinatorial embedding has a genus of one.

Algorithm 7.3.1: COMPUTE$(\mathcal{R}(\gamma), n, m)$

```
Input: R(γ), Sets of generator positions for symmetry group γ
Input: n, Number of rows of in lattice
Input: m, Number of columns in lattice
begin
    for R ∈ R(γ) do
        Initialize data with n, m and the values of R
        BACK(1, data)
    end
end
```

Algorithm 7.3.2: BACK$(i, data)$

Input: $data$, $n \times m$ toroidal array
Input: i, index into $data$
begin
  if $i > n \times m$ **then**
    FINISH$(data)$
  end
  for $c \in configs$ **do**
    for $e \in c$ **do**
      ADDEDGE$(e, i, data)$
      for *Transformations* e' *of* e *under* R **do**
        ADDEDGE$(e', i', data)$
      end
      if *All edges add successfully* **then**
        BACK$(data, i + 1)$
      end
      Roll back changes to $data$
    end
  end
end

Algorithm 7.3.3: ADDEDGE$(e, i, data)$

Input: e, An edge
Input: i, A row-by-row index into $data$
Input: $data$, $n \times m$ toroidal array
Output: $True$ if edge successfully added to $data$
Output: Updated $data$
begin
  $u \leftarrow$ vertex at i
  $v \leftarrow$ opposite endvertex of e
  Check degree of u and v
  Check for edge crossings
  Update vertex labels in $data$ for both u and v
end

Algorithm 7.3.4: FINISH$(data)$

Input: $data$, $n \times m$ toroidal array
begin
  Verify $data$ is prime
  Verify $data$ is connected
  Verify $data$ has a genus of 1
  Verify $data$ is the lexicographically least representative of its isomorphism class
  Verify $data$ does not contain any unwanted symmetry
  if *All conditions pass* **then**
    Update enumeration count
    Print data
  end
end

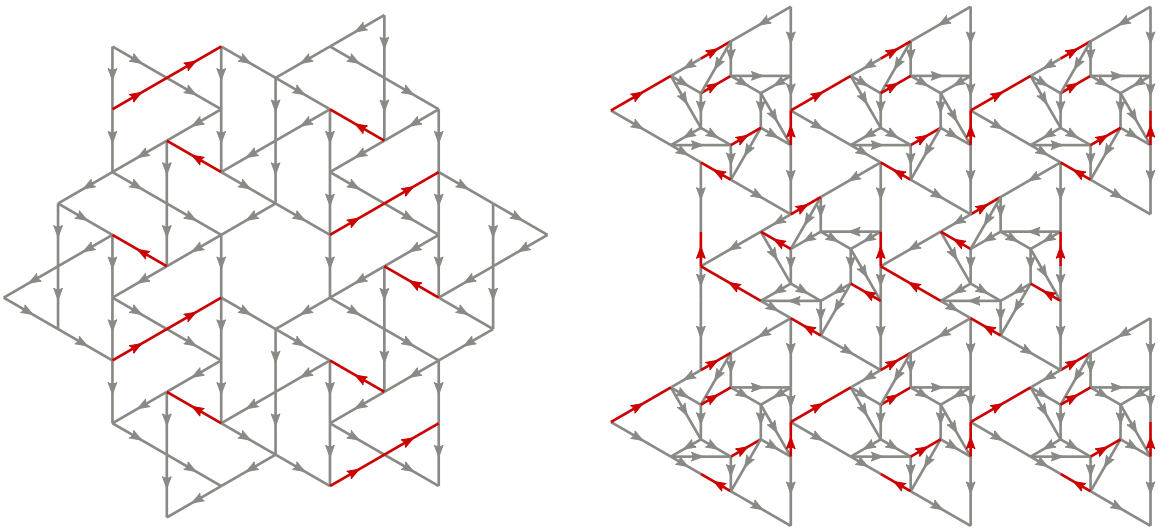
For the second stage, assigning an orientation to each edge in the graph embedding, the solution set is the same $n \times m$ toroidal array used in the first stage with the addition that edges in the rotational order label of a vertex are given an orientation. The property $\tilde{P}_\ell(x_1, ...x_l)$ is defined by the following rules:

1. Each vertex has in-degree ≤ 2 and out-degree ≤ 2
2. Outgoing edges are consecutive in rotational order
3. An oriented edge contributes to the out-degree of one of its endvertices and the in-degree of the other

At each level of the backtracking tree a vertex is selected and all edges incident to the vertex are oriented. Vertices are processed in order of the extent to which direction has been assigned to the edges of a vertex; vertices with a higher percentage of oriented incident edges will be processed first. A lightweight triage of the vertices is all the sorting that is required; vertices are placed in one of five buckets: 0, 1, 2, 3, done. All vertices start out in bucket 0. When an edge is oriented, the two endvertices of the edge move up one bucket. When all edges are in the done bucket or we have exhausted the options for orienting the edges of a vertex, that branch of the tree is terminated. The triage approach significantly reduces the number of nodes in the backtracking tree. If none of the edges incident to a vertex have been oriented, there are four choices to explore equating to four

branch points in the backtracking tree. If three out of four edges incident to a vertex have been assigned a direction there is only one choice for the remaining edge. When all edges have been oriented, the digraph is tested for presence of contractible cycles and the wrapping index of its osculating paths.

At the beginning of our search for all 17 symmetry groups, we allowed only weakly monotonic step vectors, similar to the choice made in 5. For most of the symmetry groups, finding tesselace embeddings with these weakly monotonic edge directions was a straightforward process. However, for two groups, 333 and 632, we came up blank. In the first stage of the algorithm we found undirected graphs with $3\bullet$ symmetry but all attempts to apply direction to the edges failed. The set of weakly monotonic step vectors was too restrictive to allow $3\bullet$ symmetry in the absence of $*3\bullet$ symmetry. While the vast majority of traditional patterns are weakly monotonic, the mathematical model presented in Chapter 3 does not include any limit on the direction of the steps. From the pattern found in Figure 2.4 it is evident that there exist some traditional patterns with steps in an "upward" direction. We therefore expanded the set of step vectors to include outgoing edges at any angle while still maintaining the requirements that (1) outgoing edges must be rotationally consecutive and (2) the tesselace embedding can be partitioned into osculating cycles with a wrapping index of $(1, 0)$. Subsequently, this larger set of step vectors was used for all enumerations of the planar periodic symmetry groups, not just the 333, $3*3$ and 632 cases.



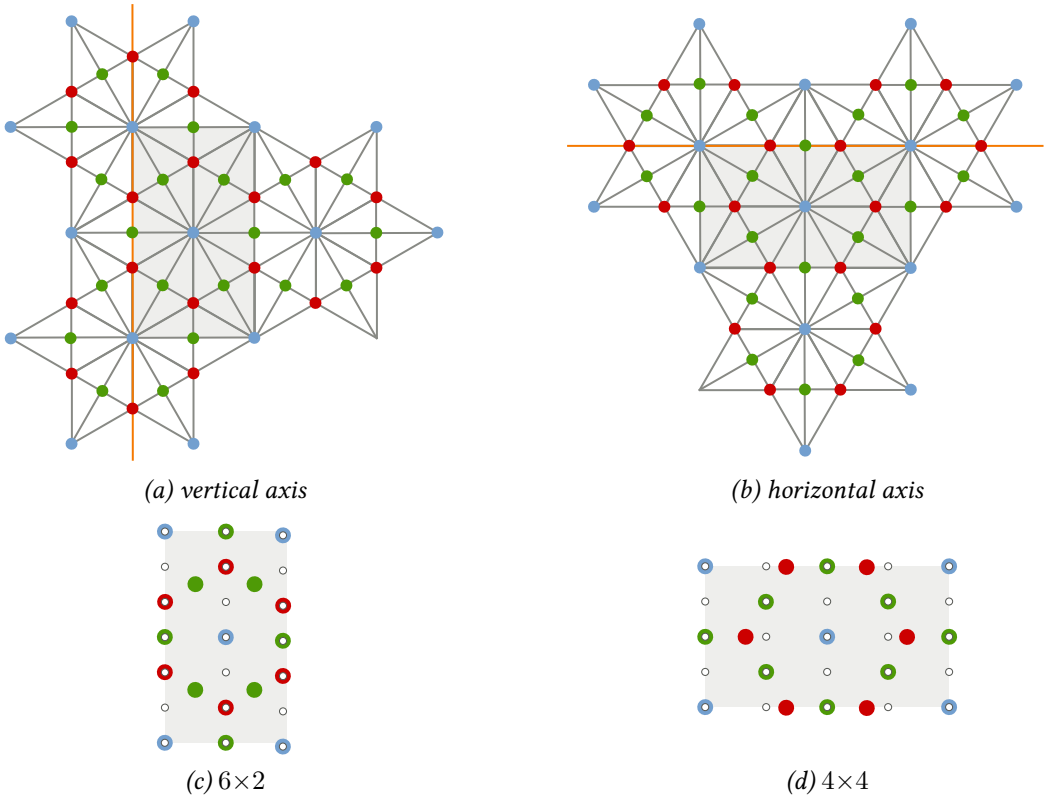
632 333

Figure 7.9: Example tesselace embeddings with edges that travel in an upward direction highlighted in red.

7.4 Results

In the following sections, for each of the 16 non-trivial symmetry groups we present example patterns and enumeration results. The patterns displayed were selected based on their aesthetic interest and because they were not found in any of the catalogues of traditional lace grounds available to the author. Of particular note, the symmetry in traditional patterns is heavily biased towards reflections and glides. Rotational symmetry of $2\bullet$ or $4\bullet$ (independent of $*2\bullet$ or $*4\bullet$) is extremely rare and, to the best of our knowledge, there are no traditional patterns from the 632, 333 or $3*3$ symmetry groups.

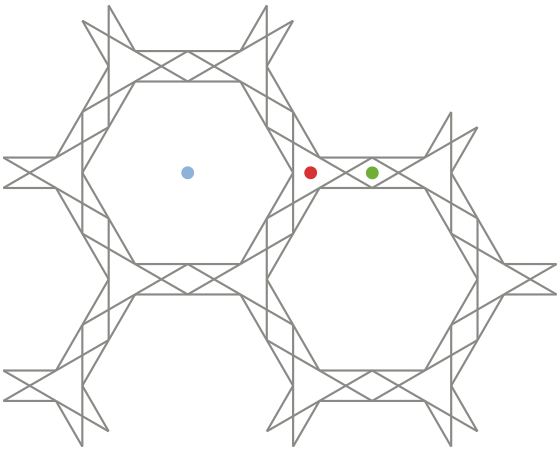
7.4.1 Six-fold Symmetry



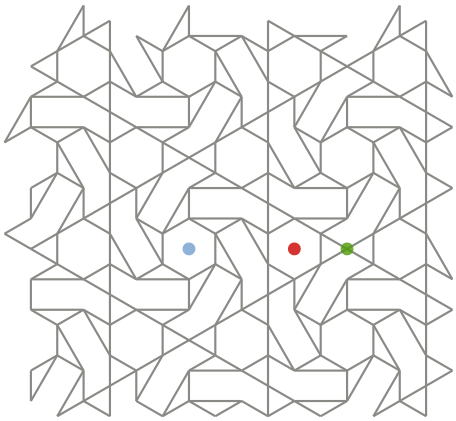
(a) vertical axis *(b) horizontal axis*

(c) 6×2 *(d)* 4×4

Figure 7.10: Six-fold symmetry generator configurations. (a) and (b) Relative position of symmetry generators for $*632$ when all three types of generators are vertically aligned or horizontally aligned. For 632, the positions are the same as for $*632$. (c) and (d) Example of generator positions for a specific period rectangle size.



$*632$
(14×14)



632
(12×12)

Figure 7.11: Six-fold symmetry example patterns

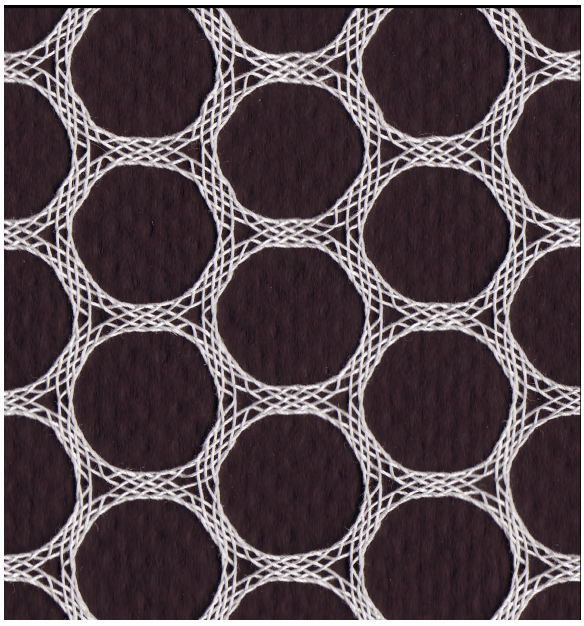


Figure 7.12: A tesselace pattern with $*632$ embedding worked in cotton thread by the author. Exhibited at JMM 2016 [28] as part of a piece called *Speculations*. The tesselace embedding can be seen in Figure 7.11.

Angle	Group	Rows	Columns	Count	Time	Undirected graphs
60	$*632$	4	4	1	0.3s	1
60	$*632$	10	10	5	5s	11
60	$*632$	18	6	4	6s	15
60	$*632$	14	14	≥ 11	DNF	DNF
60	632	8	8	8	5s	11
60	632	10	10	5	5m	103
60	632	18	6	8	10.6h	72
60	632	12	12	≥ 66	DNF	DNF

Table 7.17: Enumeration results for 6-fold symmetry groups on 60° grid

The enumeration results for $\gamma = *632$ and $\gamma = 632$ are shown in 7.17. Only counts with non-zero values are included in the table. The $6^\bullet$ and $*6^\bullet$ centers, which must be aligned with a lattice point, appear both at $(0, 0)$ and in the center of the embedding as shown in Figure 7.10. This forces the embedding to have an even number of rows and columns. The period rectangle has a row to column ratio of $3{:}1$ when all three distinct point group centers are vertically aligned and $1{:}1$ when they are horizontally aligned as shown in Figure 7.10.

While the performance of the algorithm did not allow us to exhaustively count the number of vertically oriented $*632$ patterns for size $14{\times}14$ and above, it did allow us to find partial results for $n = 14, 16$ and 18. We conjecture that $*632$ tesselace embeddings of size $n{\times}n$ exist for all even values of $n \geq 14$. Similarly, we conjecture that 632 tesselace embeddings of size $n{\times}n$ exist for all even values of $n \geq 8$. We were also able to find partial results for horizontally oriented embeddings of size 24×8 for both 632 and $*632$. We conjecture that tesselace embeddings for both six-fold symmetry groups exist for all $3n{\times}n$ grids for all even values of $n \geq 6$.

The single $4{\times}4$ result for $*632$ is the well known *Point de Paris* ground shown in Figure 7.22(b).

7.4.2 Three-fold Symmetry

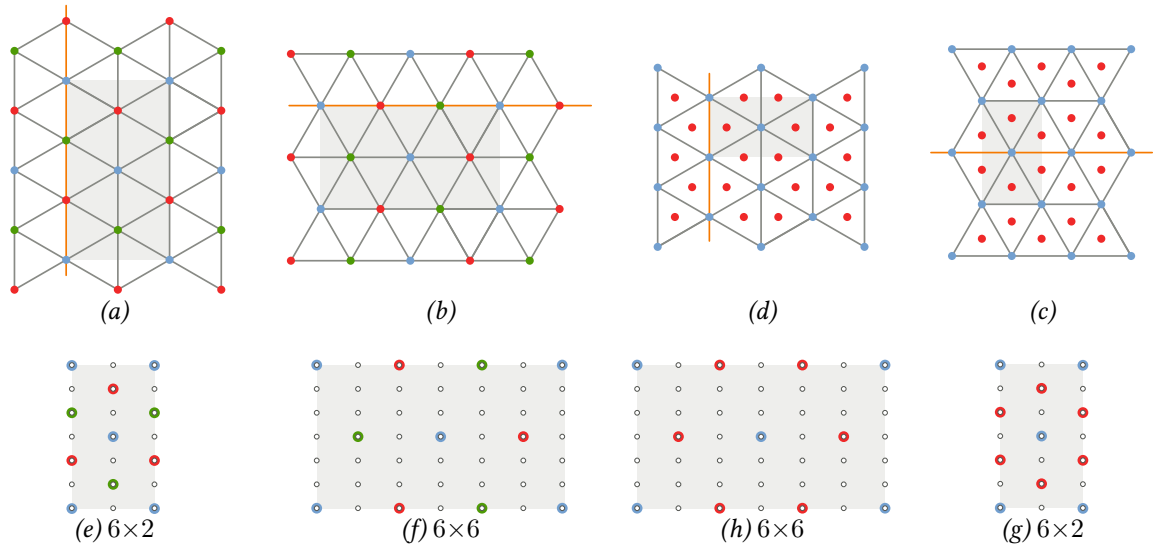


Figure 7.13: Three-fold symmetry generator configurations. (a) and (b) Relative position of ∗333 symmetry generators when there exists either a vertical or horizontal mirror line. For 333, the positions are the same as for ∗333. (c) and (d) Relative position of 3∗3 symmetry generators when there exists either a vertical or horizontal mirror line. (e) through (h) Examples for a specific period rectangle size.

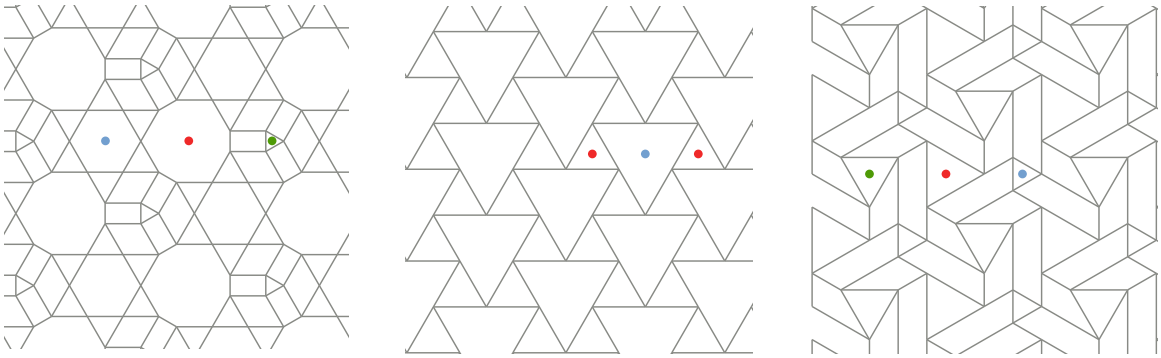


Figure 7.14: Three-fold symmetry example patterns

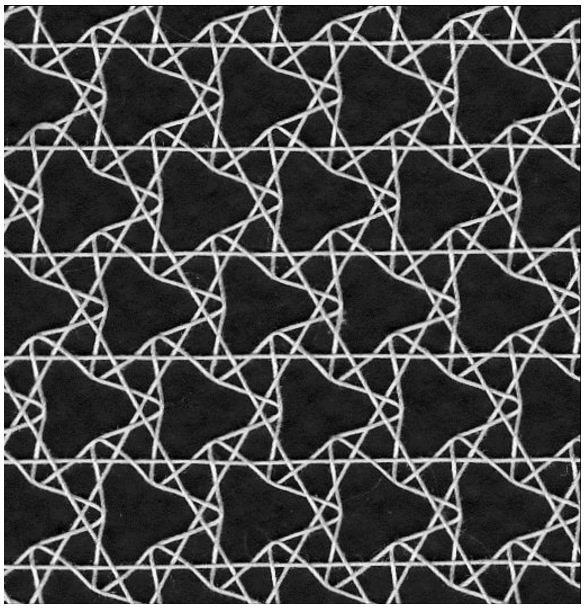


Figure 7.15: A tesselace pattern with $3{*}3$ embedding worked in cotton thread by the author. Tesselace embedding can be seen in Figure 7.13

Angle	Group	Rows	Columns	Count	Time	Undirected graphs
60	$*333$	6	6	1	1s	5
60	$*333$	8	8	2	6s	8
60	$*333$	10	10	8	2m	85
60	$*333$	18	6	9	5m	182
60	333	4	4	1	0.3s	1
60	333	6	6	4	2s	11
60	333	8	8	83	3m	457
60	333	10	10	1,560	3.5h	47,858
60	333	12	12	≥ 104	DNF	DNF
60	333	18	6	≥ 631	DNF	DNF
60	$3*3$	6	6	1	0.3s	1
60	$3*3$	8	8	7	15s	20
60	$3*3$	10	10	18	8m	256
60	$3*3$	12	12	≥ 67	DNF	DNF
60	$3*3$	18	6	28	42m	342

Table 7.18: Enumeration results for 3-fold symmetry groups on 60° grid

In all configurations of three-fold embeddings, there is either a $3\bullet$ or $*3\bullet$ point group centered at a row half-way down the period rectangle. The center for the $3\bullet$ and $*3\bullet$ point groups can not appear at a point between midway between rows. Therefore the number of rows must be an even number. The period rectangle has a row to column ratio of either 3:1 or 1:1 as shown in Figure 7.13. Therefore, if the number of rows is an even number, the number of columns is also even. As a result, for all types of three-fold embeddings, the period rectangle must have an even numbers of rows and columns.

We conjecture that $*333$ and $3*3$ tesselace embeddings of size $n\times n$ exist for all even values of $n \geq 6$ and that 333 tesselace embeddings of size $n\times n$ exist for all even values of $n \geq 4$. We have found exhaustive results for all types of three fold tesselace embeddings of size $3n\times n$ for $n = 6$ and partial results for $n = 8$. We therefore conjecture that $*333$, 333 and $3*3$ tesselace embeddings of size $3n\times n$ exist for all even values of $n \geq 6$.

7.4.3 Four-fold Symmetry

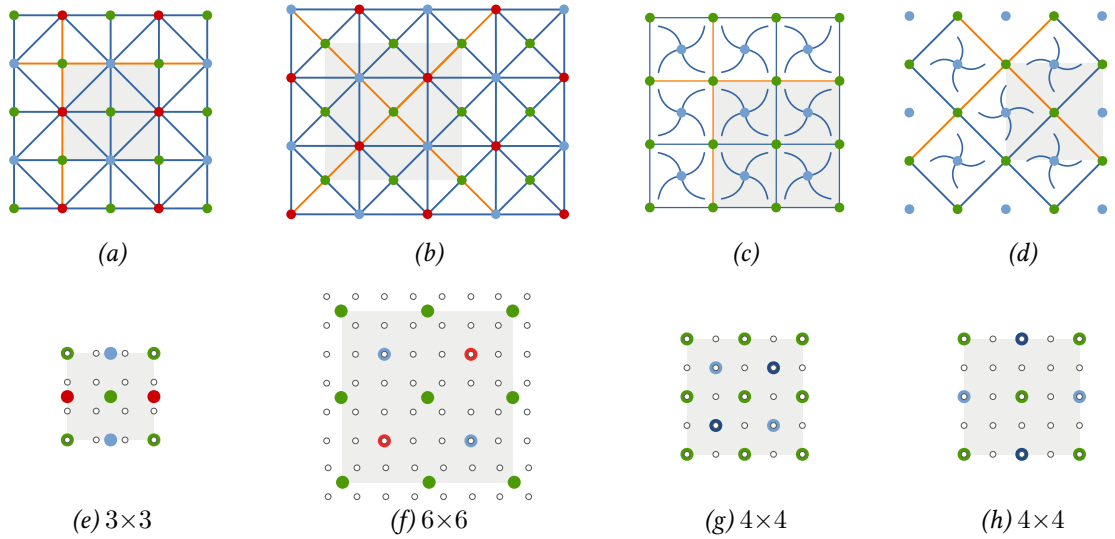
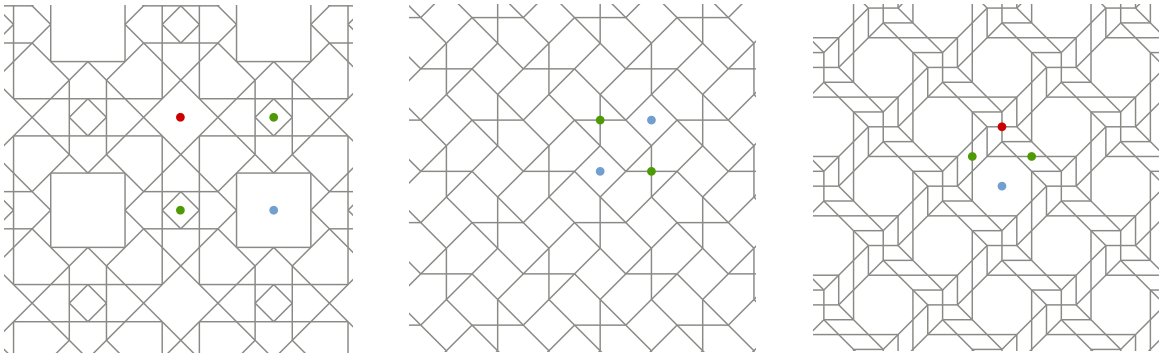
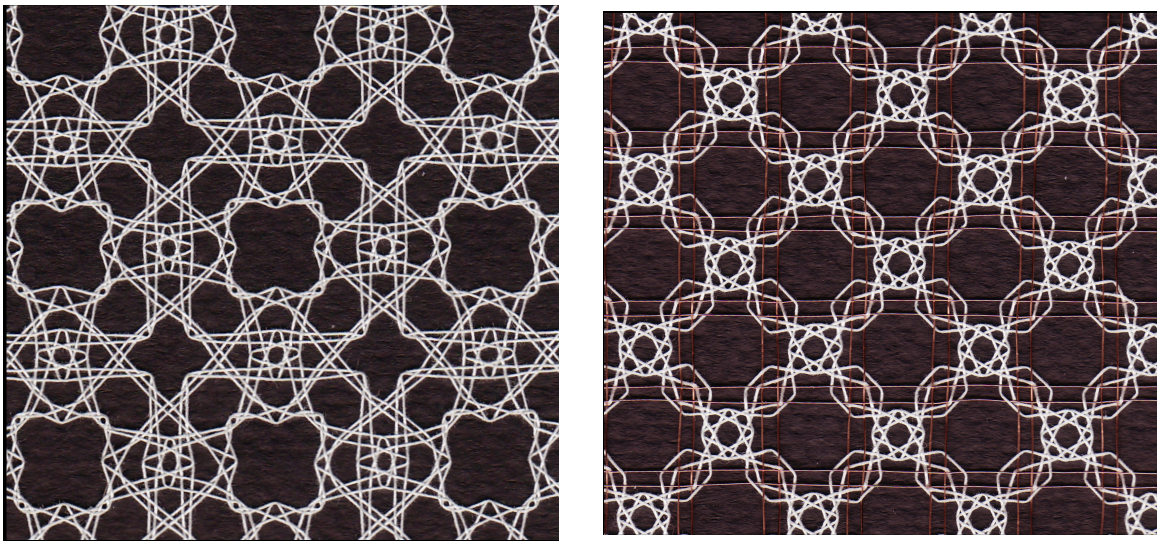


Figure 7.16: Four-fold symmetry generator configurations. (a) and (b) Relative position of $*442$ symmetry generators when 2 fold reflection is either vertical/horizontal or diagonally aligned. For 442, the positions are the same as for $*442$. (c) and (d) Relative position of $4*2$ symmetry generators when 2 fold reflection is either vertical/horizontal or diagonally aligned. (e) through (h) Examples for a specific period rectangle size.



$*442$ (10×10) | $4*2$ (4×4) | 442 (8×8)

Figure 7.17: Four-fold symmetry example patterns



(a) *(b)*

Figure 7.18: Two tesselace patterns with $*442$ embeddings worked in (a) cotton thread and (b) cotton thread and copper wire by the author. The tesselace embedding for (a) can be seen in Figure 7.17

Angle	Group	Rows	Columns	Count	Time	Undirected graphs
45	$*442$	2	2	1	0.2s	2
45	$*442$	3	3	1	0.4s	3
45	$*442$	4	4	2	1s	14
45	$*442$	5	5	3	3.6s	42
45	$*442$	6	6	20	27s	308
45	$*442$	7	7	9	3.5m	1,421
45	$*442$	8	8	345	1.7h	21,446
45	442	3	3	1	0.3s	5
45	442	4	4	7	0.9s	53
45	442	5	5	23	14s	962
45	442	6	6	598	48h	7,870
45	$4*2$	4	4	4	1s	8
45	$4*2$	6	6	95	15h	354

Table 7.19: Enumeration results for 4-fold symmetry groups on $45°$ grid

All tesselace embeddings with four-fold symmetry have a $n \times n$ period rectangle as shown in Figure 7.16.

The symmetry group $4*2$ possesses two perpendicular glide reflections. A glide reflection includes a translation of a distance d along its glide line. Since a glide takes a point back to itself after two repeats, the distance d must equal $n/2$ where n is the length of the period rectangle in the direction of the glide line. The endvertex of an edge must lie on a lattice point, therefore the distance $n/2$ must be an integer value requiring n to be an even number. As a consequence, no $4*2$ tesselace embeddings exist for odd values of n.

The 2×2 tesselace embedding for $*442$ is the embedding for many traditional patterns such as the *Torchon* ground and *half-stitch* ground shown in Figure 3.1.

7.4.4 Two-fold Symmetry

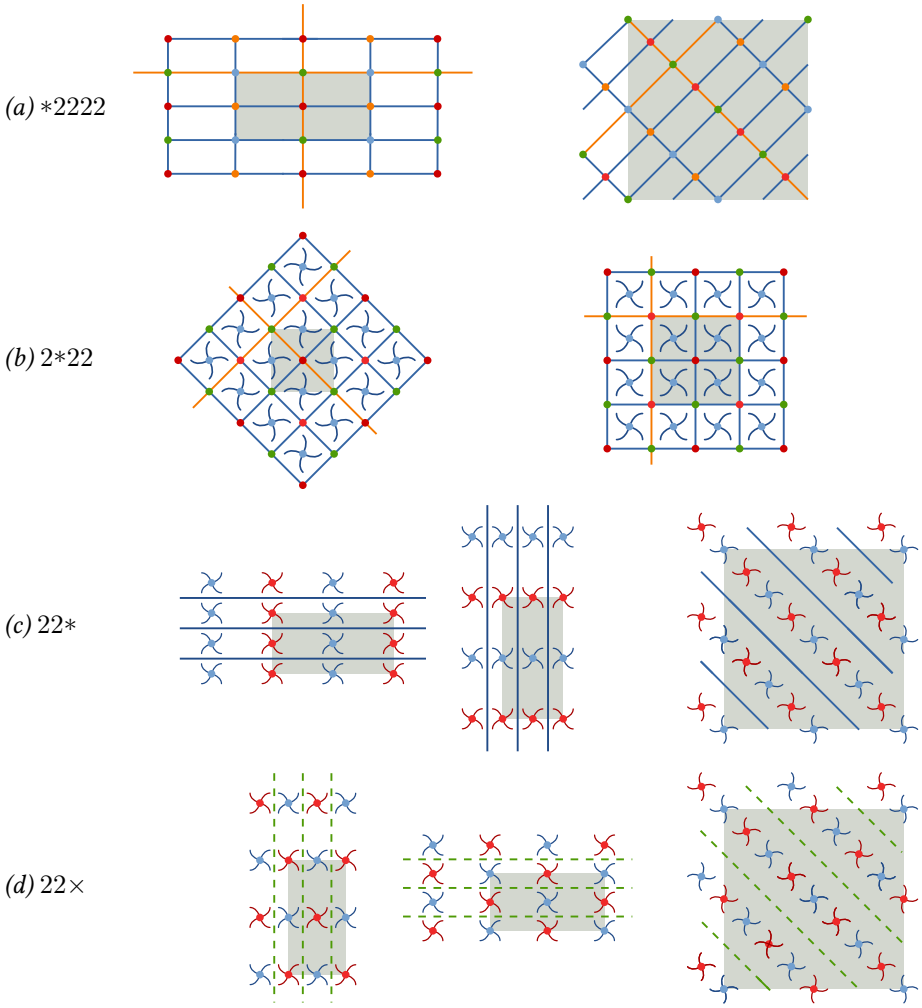


Figure 7.19: Two-fold symmetry generator configurations. (a) and (b) Generator configurations when 2 fold reflections are all either vertical/horizontal or diagonally aligned for ∗2222 and 2∗22, (c) and (d) Vertical, horizontal and diagonal configurations for 22∗ and 22×.

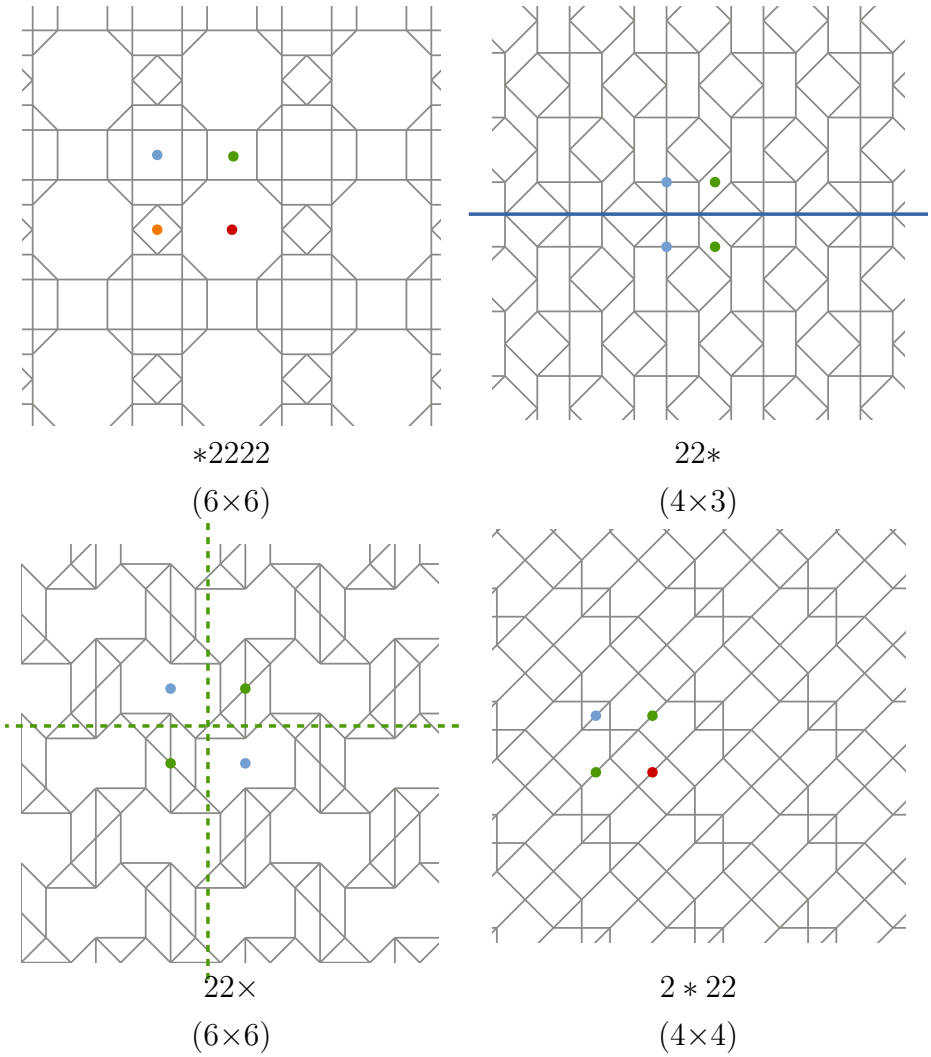


Figure 7.20: Two-fold symmetry example patterns

Rows/Columns	2	3	4	5	6
2	0	0	0	0	1
3	1	1	4	7	11
4	1	2	9	13	77
5	4	8	27	29	122
6	7	18	142	156	1847†

† Completed in 25 minutes

Table 7.20: Enumeration results for ∗2222 on 45° grid

Rows	Columns	Count	Time	Undirected graphs
2	2	2	0.8s	3
3	3	3	0.3s	25
4	4	53	3s	265
5	5	102	1.3m	4,313
6	6	5,328	40h	195,280

Table 7.21: Enumeration results for $2{*}22$ on 45° grid

Rows/Columns	2	3	4	5
2	1	3	6	10
3	3	0	11	0
4	16	48	136	358
5	7	0	75	0

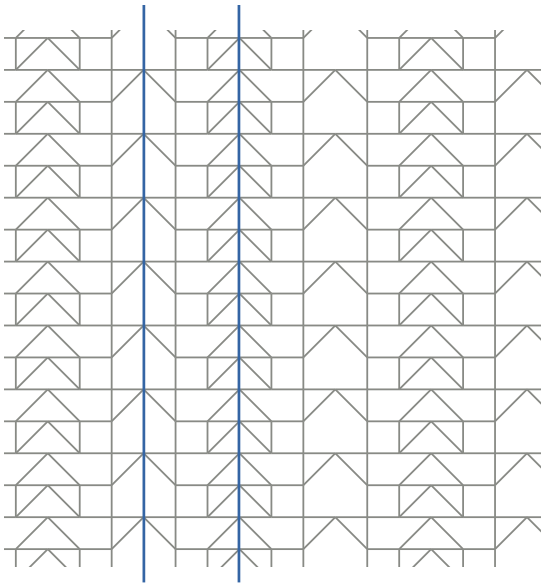
Table 7.22: Enumeration results for $22{*}$ on 45° grid

Rows/Columns	2	4	6
2	0	7	37
4	9	203	$\geq 2,251$
6	49	2,251	

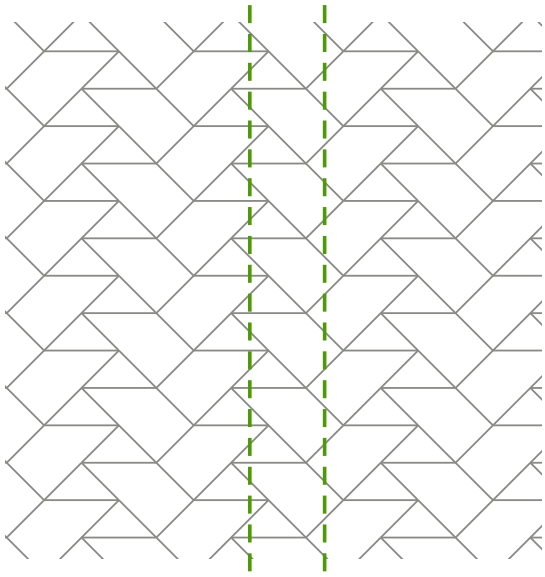
Table 7.23: Enumeration results for $22\times$ on 45° grid

The large variety of configuration options for the $2^\bullet$ and $*2^\bullet$ point groups allows for period rectangles of any integer size for $\gamma = *2222$ and $\gamma = 22*$. Tessellations with symmetry $2{*}22$ have a square or rhombic unit cell requiring a square $n{\times}n$ period rectangle. The symmetry group $22\times$ has two perpendicular glide reflections. As described for $4{*}2$, the length of a period rectangle in the direction of a glide must be an even value.

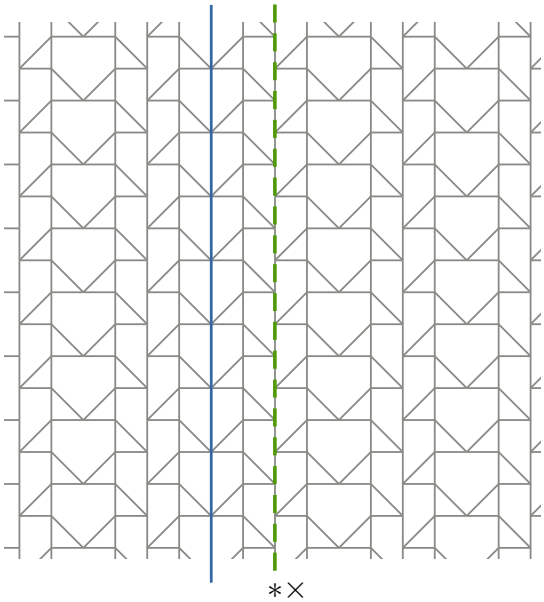
7.4.5 Mirrors and Glides



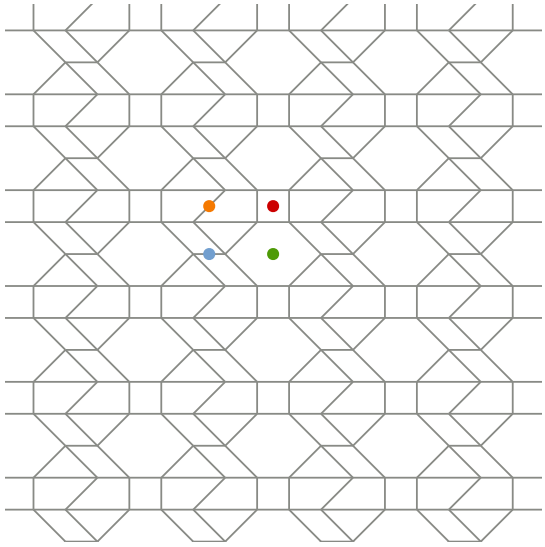
**
(2×6)



××
(2×4)



*×
(2×8)



2222
(3×4)

Figure 7.21: 2222 and parallel mirror symmetry example patterns

Rows/Columns	2	3	4	5
2	3	16	52	165
3	22	111	108	557
4	69	647	6,946	51,003[†]
5	254	3,531		
6	933	21,763		

[†] Completed in 35 hours

Table 7.24: Enumeration results for 2222 on $45°$ grid

Rows/Columns	1	2	3	4	5	6
2	0	0	0	4	0	26
3	0	0	3	28	58	615
4	1	4	47	423	1,986	
5	1	27	152	1,815		
6	10	152	2,321	37,755		

Table 7.25: Enumeration results for $**$ on $45°$ grid

Rows/Columns	2	4	6
2	2	31	344
4	28	4,167	$\geq 45,619$
6	347	$\geq 59,411$	

Table 7.26: Enumeration results for $\times\times$ on $45°$ grid

lattice	Rows/Columns	2	4	6	8
square	2	0	1	4	17
square	4	2	16	120	975
square	6	9	153	2,965	$\geq 13,088$

Table 7.27: Enumeration results for $*\times$ on $45°$ grid

Out of all 16 non-trivial planar periodic symmetry groups, 2222 has the highest degree of freedom for the relative placement of its generators which can be arranged in any parallelogram shape. Not surprisingly then, we find a large number of 2222 tesselace embeddings for all period rectangle sizes. The placement of parallel mirror and glide lines is also quite flexible. The only restriction occurs when glide lines are involved. As mentioned in the previous section, the size of the period rectangle in the direction of the glide line must be an even number. For $*\times$, in the direction perpendicular the mirror and glide lines, a point must reflect, glide, reflect, glide in order to return to its original position. The spacing between mirror and glide lines is therefore $n/4$ where n is the direction perpendicular to the mirror line. Sine reflections and glides can only be positioned on a lattice point or midway between lattice points, n must be an even number.

We can also compare the performance of the backtracking approach against a brute force generation over all possible outgoing labels. For a square lattice using the set of step vectors shown in Figure 7.1, the number of undirected edge configurations per vertex is $c = 1,120$ (all combinations of 4 step vectors from a set of 16 such that no two steps have the same angle). The number of configuration combinations for an $n\times m$ period rectangle is c^{nm}, e.g. For $n = 4, m = 5$, the number of combinations analysed in a brute force approach is 9.6×10^{60}. In our results, for the first stage of the algorithm the number of nodes in the backtracking tree for an undirected graph with $n = 4, m = 5$ and $\gamma = 2222$ is $52,631,384$ which produces $698,556$ undirected graphs resulting after edge orientation in $51,003$ tesselace embeddings. For a $*442$ tesselace embedding with $n = 5, m = 5$ the backtracking tree contained only $1,151$ nodes and generated 42 valid undirected graph embeddings resulting after edge orientation in 3 tesselace embeddings.

7.5 Discussion

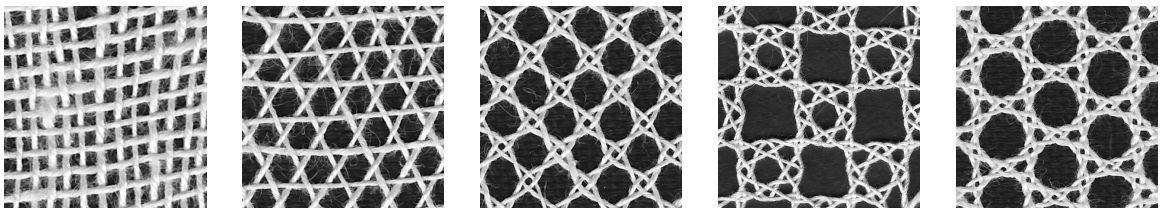
Our focus has been on symmetry in the tesselace embeddings but we can not end this chapter without a discussion of the relationship between symmetry in the tesselace embedding and symmetry at the level of individual threads.

Grünbaum and Shepherd explored the concept of symmetry at the level of individual threads in their seminal work on the geometry of fabrics [33]. Their focus was strictly on fabrics woven on a fixed frame loom which we shall refer to as cloth. While bobbin lace and cloth are both physical manifestations of mathematical braids, they have taken quite opposite directions with regard to creating diversity through braids. The threads in cloth are partitioned into two perpendicular sets which always meet at right angles. There is no variation in the paths taken by individual threads. Variety in cloth is obtained instead by changing the order of over and under crossings. In contrast, bobbin lace is always formed as an alternating braid, the order of over and under crossings always being the same. Variety in bobbin lace is obtained by varying the path of the individual threads and changing which threads are involved in a crossing.

Grünbaum and Shepherd introduced the terms *isonemal*, meaning there exists a symmetry relation that maps any strand in a cloth to any other strand in that cloth, and *mononemal*, meaning that every strand in the fabric has the same order of over and under crossings but one strand can not necessarily be transformed into another by a symmetry operation. They also extended the concept of isonemal patterns to k-*nemal* patterns in which there are k classes of strands. Because the threads follow straight line paths, their exploration of similarity between threads is based entirely on crossing order. For bobbin lace, the path of the thread must be considered.

In bobbin lace, there exist isonemal grounds such as Half-stitch (632), Cloth-stitch $(4{*}2)$, Torchon Ground $(4{*}2)$ and Point de Paris (632). There also exist $2-nemal$ grounds such as double trellis (632) [36, Fig. 104]. For the most part, however, the path of an individual thread is quite different from that of its neighbours. In addition, the alternating braid structure favours rotational symmetry which limits the symmetry groups you will find at the thread level. But lace is rarely examined under a magnifying glass and was intended to be viewed at the macro level. In cloth the over and under crossings affect the appearance of the cloth even from a distance. It is the over and under crossings in cloth that allow us to see the difference between satin and plain cloth. In some cases colour is used to accentuate this difference as in hound's tooth. In bobbin lace, at the thread level Rose ground has 442 symmetry but when viewed in a piece of lace the subtlety of which

thread is on top is lost and the general impression is that of $*442$ symmetry. By selecting combinations of cross and twist in the ζ mapping of a tesselace pattern, it is possible to reflect fairly closely the symmetry structure visible in the tesselace embedding as shown in Figures 7.12, Figure 7.15, and Figure 7.18.



(a) *(b)* *(c)* *(d)* *(e)*

Figure 7.22: Some traditional lace grounds (a) Cloth-stitch ground (also known as plain weave) (b) Half-stitch ground (also known as triaxial weave) (c) Torchon ground (d) Rose ground (e) Point de Paris

In closing we also acknowledge the related work of Grünbaum and Shepherd [32, p. 7.4] in which they explore the types of patterns that can be formed from line segments. The patterns are classified by the type of intersection (no intersection, intersect at endvertices or intersect at interior points) and an equivalence relation they call homeomerism. Homeomerism is a much stricter definition of equality than is used in the planar periodic symmetry groups and can be used to identify isohedral (face transitive), isotaxal (edge transitive) and isogonal (vertex transitive) patterns. For periodic planar patterns like our tesselace embeddings in which line segments intersect only at the endvertices and each endvertex has degree 4, Grünbaum and Shepherd have identified five homeomerism classes.

Chapter 8

Evaluation and Future Work

In this chapter we will reflect on the results obtained so far and compare them to traditional lace ground designs. We shall also discuss ideas for future work; how to extend the range of patterns that can be generated algorithmically and how the results obtained so far can be used as a data resource for further explorations.

8.1 Comparison to Traditional Lace Grounds

Over the past century, fibre art historians have been identifying and documenting the lace found in museums and private collections. Most of the early pieces have been lost either through decay, fire or because when fashions changed the precious metals used in their silver and gold threads were reclaimed for use in new pieces. In many cases, the only records we have of the first patterns come from portrait paintings and merchant records. Pattern books such as *Le Pompe*, published in Venice in 1557, and *Ein Nüw Modelbuch*, published in Zürich around the same time, are invaluable resources. Most of the literature on lace history focuses on identification: determining the method use to create it, the era in which it was produced and place of origin [52, 76]. A few books, such as *The Art of Old Lace* [36] from 1931, include a survey of grounds that are common to a particular style. The selection of grounds mentioned in these texts is chosen because of their key role in the identification of the lace.

In the last 50 years, lace practitioners have published a number of excellent reference catalogues documenting lace grounds in detail [17, 85, 53]. The catalogues cover a large range of time periods and styles but they are not complete. What percentage of historical patterns they represent is unknown. Over time, it is hoped that the records will become more comprehensive although funding and expertise for this task are scarce. The objective of determining whether a particular lace ground is a new discovery is therefore not very practical. We can, however, make some general observations.

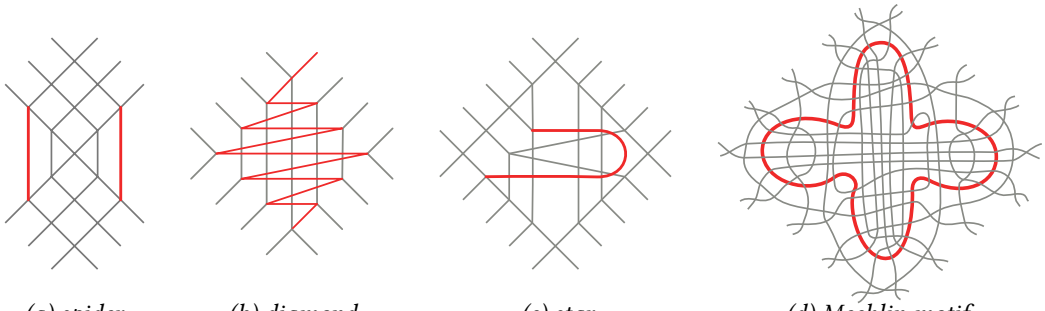
Source	Number of Lace Grounds[1]
The Book of Bobbin Lace Stitches [17]	262
Gründe mit System [85]	449
Viele Güte Gründe [53]	344
Algorithm	$> 5{\times}10^6$ (independent of $\zeta(v)$ mappings)

Table 8.1: Number of lace grounds reported by source

Table 8.1 lists the number of lace grounds reported in three well respected catalogues. There is overlap between the catalogues and in several cases one ground embedding is combined with a variety of cross-twist combinations, each combination being reported as a separate ground. As a result, the total number of distinct embeddings reported is less than the sum of $262 + 449 + 344 = 1,055$. In comparison, the algorithmic count in Table 8.1 strictly enumerates the number of ground embeddings with no assumption about a particular ζ function for the vertices. Based on the sheer numbers, it is quite safe to conclude that the algorithmic approach has identified a large number of new patterns.

A visual examination of the catalogued patterns reveals that the majority of patterns include some form of symmetry (beyond the two perpendicular translations required for tessellation). As in Chapter 7, we will refer to the symmetry of the tesselace embedding rather than the symmetry of individual threads. The most common symmetry groups are $*\times$, $**$ and $*2222$. Quite a few of the larger patterns included in *Viele Güte Grunde* contain four fold reflection ($*442$) or come very close to it. We have found no examples of traditional grounds with either three- or six-fold rotational symmetry which excludes all 632, 333 and $3*3$ patterns. The absence of these symmetries is not surprising. As shown in Tables 7.17 and 7.18 the number of such patterns is quite low for small grid sizes and all require edges with 'upward' step vectors. While upward steps are not unheard of, they are much less intuitive to discover.

While we have expanded the number of known patterns, there are some notable traditional patterns not generated by our restricted set of step vectors. Examples are shown in Figure 8.1. The missing patterns meet all five properties laid out in our mathematical model but they include vertices that do not align with a lattice point. The motifs can be characterized by a dense arrangement of threads, usually due to some type of back and forth weaving. Each motif has a large number of interactions but the layout of the interactions can be described by a small number of intersecting paths. Rather than using step vectors, a more efficient way to represent these elements could be investigated.



(a) spider *(b) diamond* *(c) star* *(d) Mechlin motif [53, A76]*

Figure 8.1: Examples of motifs not generated by algorithm. The number of vertices within each motif are (a) 17, (b) 37 (c) 32 and (d) 151.

The catalogue *Viele Güte Gründe* contains several large *Mechlin* grounds with a period rectangle of size 25×25. These large grounds frequently approximate $*442$ symmetry and include woven motifs of various shapes such as stylized flowers and hearts (see Figure 8.1(d)). In the author's opinion, generating the large, complex *Mechlin* grounds of the 1700's is the holy grail of using computational bobbin lace to generate traditional patterns. Due to the performance limitations of our algorithm, we can not exhaustively generate all the tesselace embeddings with such large period rectangles, although partial results are obtainable in many cases. The current approach could be modified to use a random generation or hill climbing strategy to find individual examples for larger grid sizes.

8.2 Influence on Contemporary Lacemakers

The true success of our research will be measured by its impact on the lacemaking community. To that end, we have shared tesselace patterns with lacemakers worldwide through online discussion forums [38, 4] and our website [41]. In order to make the patterns available in a form useful to designers, we wrote an extension [39] to an open source scalable vector graphics tool called Inkscape. The extension allows designers to fill a rectangle with any one of the tesselace embeddings. Properties such as the scale and angle of the lattice grid can be specified. Once rendered inside Inkscape, the designer can use the tool's full set of drawing capabilities for colouring, resizing, as well as more advanced effects such as bending. The template to define a tesselace embedding is both simple and flexible and could be used to describe any of the alternating braid patterns found in the catalogues.

Feedback from the community has been extremely positive. The author has been invited to conduct workshops introducing the new patterns and to publish articles in the lace magazines *Veulta y Cruz* [40] and *Australian Lace*. Lenka Suchanek, a professional lacemaker who has exhibited world wide, chose a tesselace pattern for her wood and wire sculpture *Waves — an offering to the moon* [79]. Jo Pol, a lace tool developer, has expanded on the Inkscape extensions, adding support for working with polar grids and visualizing the effect of different cross and twist actions [70]. Pol has also been using the tesselace patterns as a base for exploring thread drawing and pin placement.

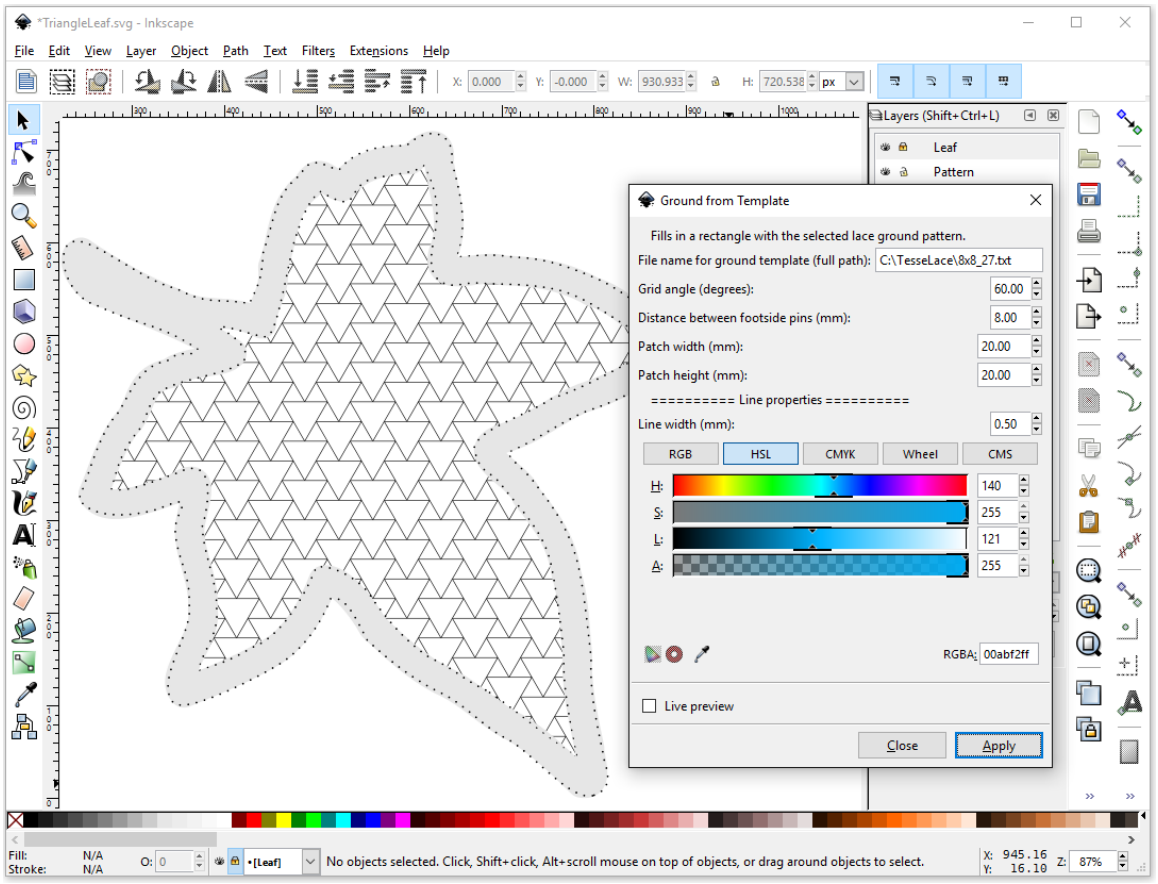


Figure 8.2: A scalable vector graphics tool for designing with tesselace embeddings

The author has created several pieces which have appeared in mathematical art exhibits of the *Joint Math Meetings* (2014/15/16) [50, 27, 28] and the *Bridges Conference on Mathematics and the Arts* [75] as well as a contemporary lace exhibit at the *Guelph Civic Museum*.

8.3 Future Work

8.3.1 Classification of Grounds

Given the large number of patterns in Table 5.2, it would be useful to classify patterns according to characteristics that affect appearance.

A quick and easy first step is to use the combinatorial embedding isomorphism described in Chapter 4. As discussed in Section 4.2, we could use a breadth first search to label the vertices in $\mathcal{O}(|V(G)|^2)$; each vertex is a potential root of the labelling, the first child can be uniquely determined by choosing the far endvertex of the first outgoing edge in clockwise order. For a combinatorial embedding label, we discard the lattice position information but, because tesselace embeddings are multigraphs, we will need to label the edges. Directed edges can be labelled in the order that they are encountered in the BFS.

As a demonstration of what is possible, Table 8.2 shows the number of non-isomorphic combinatorial embeddings obtained by post processing the tesselace embeddings generated in Chapter 5 for up to 10 vertices. Note: The numbers in Table 8.2 do not represent an exhaustive enumeration. As mentioned previously, the set of step vectors used in the algorithm is only a subset of what is possible in bobbin lace. Further, for the set of step vectors used, not all patterns with n vertices have been generated — only those that fall within the range of rectangle sizes examined.

Vertices:	1	2	3	4	5	6	7	8	9	10
Combinatorial Embeddings:	1	2	4	14	22	43	70	139	181	281

Table 8.2: Classification of results in Table 5.2 by number of vertices and combinatorial embedding isomorphism for up to ten vertices

Classification of graph embeddings by isotopy will require a bit more effort as this is a relatively new and active area of research. In 2013 de Verdière and de Mesmay [20] demonstrated a linear time theoretical algorithm for determining whether two graph embeddings on a surface are isotopic. In this case, performance is relative to the sum of the complexity of two embeddings; i.e., the total number of vertices, edges and faces in both graphs. At this time, however, an implementation of their algorithm does not exist so it is not known whether the algorithm is practical. For small numbers of vertices, the isotopy classes can be determined manually from the results of Table 5.2 as shown in Figure 8.3.

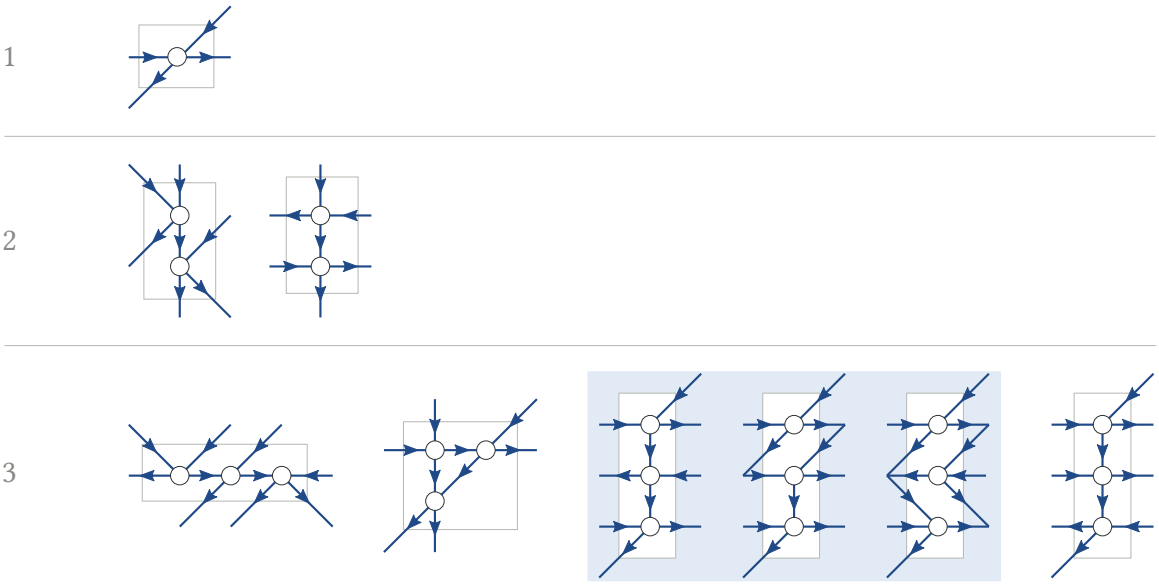


Figure 8.3: Isotopy classes for tesselace embeddings from Table 5.2. Blue box indicates tesselace embeddings with the same combinatorial embedding.

8.3.2 Rendering Tesselace Patterns

A tesselace embedding generated by our algorithm is only half of a tesselace pattern; it needs to be combined with cross and twist actions defined by a ζ mapping in order to be worked as lace. The number of possible ζ mappings depends on the number of vertices in the embedding. If we consider just three basic types of action sequences, CTp, $CTpC$ and $CTpCT$, then a tesselace embedding with 10 vertices can be worked in $59,049$ different ways. Typically one would use the same set of actions on vertices in the same orbit so this would reduce the number of options to consider for patterns with symmetry; however, the number of combinations is still very large. It would greatly speed up the exploration of new patterns if there was a fast way to accurately render the physical behaviour of threads under tension for a given $(\Delta_1(G), \zeta)$ pair. Cantarella, Piatek and Rawdon have developed a library for rendering the tightening of knots [14]. This library, called `ridgerunner`, could be modified to model the behaviour of threads in bobbin lace by adding support for the way in which pins allow threads to resist the tightening forces.

8.3.3 Tesselace Embeddings with Specific Properties

For artistic effect or for an application such as connecting a network of sensors, it may be important to specify the path of a thread (or a set of threads) and look for a pattern that will embed that path in a secure structure. The material used in the application may impose additional constraints such as the number of twists — too many twists may affect its performance — or how close together the twists may be — a metal ribbon that has a significant width may need to be folded rather than twisted. When working with the path of an individual thread, it becomes necessary to consider both the tesselace embedding and the ζ mapping. Such an exploration could start by laying down edges along the path of the thread, then complete the tesselace embedding by adding more edges following the five properties of the mathematical model plus the additional constraint that marked edges belong to a specific pair of threads.

Lace is defined as *a decorative openwork fabric in which the pattern of spaces is as important as the solid areas* [76]. So far we have focused on the threads and where they travel. We should also explore where the threads do not travel; the open spaces. A simple first step would be to classify the results in Table 5.2 by the size (number of vertices bordering) and shape (concave, convex, regular, irregular) of the largest faces in the tesselace embedding. The current backtracking approach, both in Chapter 5 and Chapter 7, could be modified to initialize the embedding with one or more holes, marking edges and vertices incident to the associated face as well as marking internal vertices as isolated, and generate tesselace embeddings with edges filling in the space between the holes. A preliminary step may be to create an exhaustive list of hole shapes. Are there an infinite number of hole shapes for a given set of step vectors or is there a maximum size and finite set? We suspect that the latter is true even if the period rectangle is made indefinitely large.

8.3.4 Breaking the Rules

Now that we have a fixed set of properties for traditional bobbin lace grounds, we can think about exploring in entirely new directions by breaking some of the rules. For example, in Chapter 3 we introduced the property of thread conservation which led to the requirement that tesselace embeddings must be partitionable into osculating circuits with a wrapping index of $(1, 0)$. What would happen if we changed the wrapping index to

$(1, 1)$? The threads would drift to the right (or left) creating a non-right-angled period parallelogram. For example, worked in copper wire, we could create an oblique ribbon that wraps in a spiral around a cylinder. In addition to the new set of patterns this would reveal, the parallelogram period could have advantages for filling in asymmetrical shapes.

Property 2 in our mathematical model is that bobbin lace grounds are periodic; they repeat in two non-parallel directions to cover a region. Is it possible to have an aperiodic lace tessellation? An answer to this question may come from studying the Ammann aperiodic sets of tilings [32].

Notes

1. Grounds borrowed from needlelace which rely on sewings are not included in these counts. Grounds which can only be worked using lazy crossings and therefore do not result in alternating braids are also excluded.

Chapter 9

Conclusion

In this thesis, we take our inspiration from the 500-year-old tradition of bobbin lacemakers and their ability to innovate and adapt. Though some today may think of lace as an ornamental frill or extravagance of a bygone era, its origins were probably very practical; woven cloth has an unfinished edge that will unravel unless it is hemmed or braided. As part of human nature, we have a tendency to go beyond what is merely practical and to create beauty and self-expression out of everyday items. Over time, the braids became more elaborate and started to cover more width than just the edge of the cloth. Simple braids were augmented with *Reticella*, a cutwork style of lace in which threads are pulled out of the cloth and the remaining threads are drawn together with needlework. Ever more elaborate designs were sought out and eventually the needlework began to dominate. At some point the cloth disappeared altogether to be replaced by a temporary backing material that was removed once the lace work was finished. This new style of lace was called *Punto in Aria* — stitches in air. One drawback of the needlework approach is that it is very time consuming and the fashions of the day required vast quantities in short order. Undeterred, lacemakers sought out a faster way to achieve the same effect. It was at this point that bobbin lace is believed to have developed — an elaborate form of ornamental braiding capable of integrating woven elements and delicate braids. Early designs were imitations of *Punto in Aria* but over time bobbin lace evolved to have its own style.

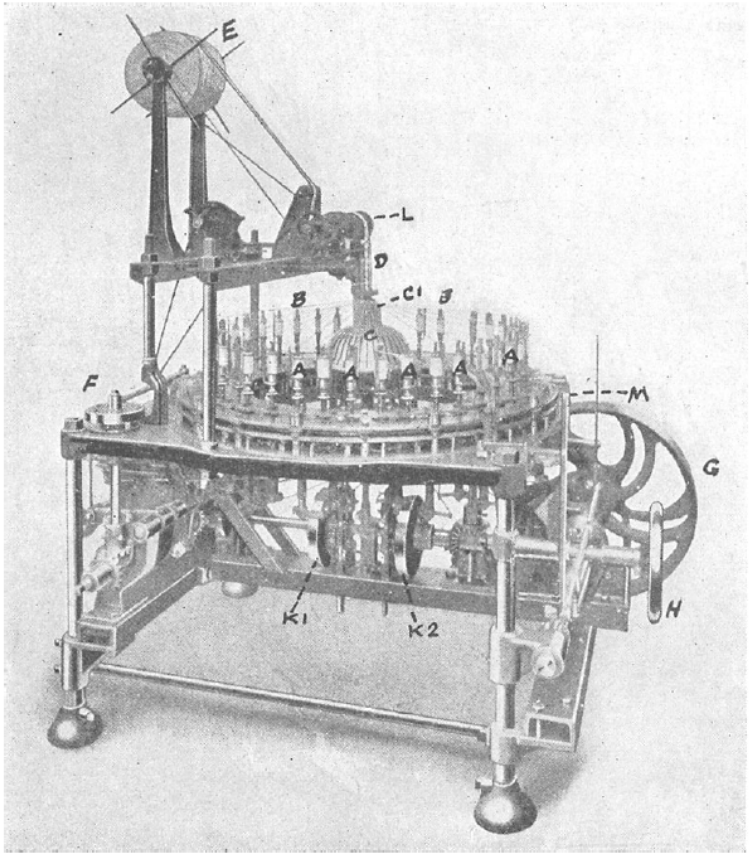


Figure 2
Barmen Torchon Machine

A—Spindle set in top plates of Barmen machine.
B—Yarn threaded up to mandrill.
C—Beater dome.
CI—Beater knives (closed).
D—Mandrill (or width factor—changeable for various desired widths).
E—Finished lace on reel.
F—Jacquard machine.
G—Drive pulley.
H—Hand wheel.
K1—, K2— } Beater cams.
L—Take-up rolls.
M—Starting handle.

Figure 9.1: A Barmen bobbin lace machine. Source: [58]

As fashions changed, so did the styles of bobbin lace. Lacemakers needed to continuously invent new grounds and techniques to keep pace. This innovative attitude carried on into the age of mechanical devices. Shortly after the invention of the Jacquard loom for weaving, the same technology was used to create nets and imitations of almost every style of lace. In the 1890's, the Barmen machine, shown in Figure 9.1, was developed [24]. Using rotating spindles, it could imitate the cross and twist actions of the lacemaker to produce simple Torchon style laces. Capable of working with up to 120 threads, it was controlled by a punched card system adapted from the Jacquard loom.

The punched card system went on to inspire Babbage and Lovelace in their development of the *Analytic Engine*, a mechanical precursor to the modern computer. It seems only fitting, therefore that computer algorithms should be used to further adapt the art and technique of bobbin lace to our modern needs.

Our contribution to the evolution of lace is a mathematical model for bobbin lace ground patterns. Leveraging the work of mathematicians in the areas of topological graph theory, group theory and braid theory, we have been able to apply ideas from discrete math to model how threads combine to create workable designs. The result is a two part representation $(\Delta_1(G), \zeta)$ where $\Delta_1(G)$ is a topological embedding of a toroidal 2-regular digraph G and ζ is a mapping from the vertices of G to a set of braid words. Our focus in this thesis has been on the properties that characterize $\Delta_1(G)$, the tesselace embedding which represents the way pairs of thread come together to form braids. We have identified five key properties of $\Delta_1(G)$ as required by bobbin lace and expressed them in terms of graph theory.

Having developed a solid, logical foundation for bobbin lace grounds, we went on to enumerate and exhaustively generate patterns that conform to that model. We started by specifying an equivalence relation and defined what makes a pattern prime so that we could identify unique representatives. We then proved that there is an infinite number of prime workable patterns. One of the key properties identified in the model is that it must be possible to partition $\Delta_1(G)$ into a set of osculating circuits such that each circuit has a wrapping index of $(1, 0)$. We used this property to exhaustively generate workable patterns for increasing numbers of vertices in G by gluing together lattice paths in an osculating manner. Using a backtracking algorithm to process the lattice paths, we identified over 5 million distinct prime tesselace embeddings all of which can be paired with twist and cross actions to form workable lace patterns. This is well in excess of the roughly 1,000 found in lace ground catalogues.

The lattice paths used in our approach are members of a family of partially directed lattice paths that have not been previously reported. We explored these paths in detail, developed a recurrence relation and generating function for their enumeration and presented a bijection between these paths and a subset of Motzkin paths.

Finally, to draw out of the extremely large number of patterns some of the more aesthetically interesting cases for lacemakers to work on, we looked for examples that have a high degree of symmetry and in the process demonstrated that there are lace ground representatives from each of the 17 planar periodic symmetry groups. In the cases of six-fold and three-fold rotational symmetry we believe that these patterns could be new as we have not found their equivalent in any catalogues.

While we have taken some significant first steps in this inaugural look at computational bobbin lace, our results are but the beginning and can hopefully serve as a launching point for future explorations. The next step is to identify what current needs bobbin lace can serve. It can of course continue to inspire artists and fashion designers, but it can also do much more. The nose cone of the NASA space shuttle uses a triaxial weave. NASA chose this method of weaving because the fibers, which run in three different directions, can distribute stretch forces much more evenly than plain woven cloth, which has a diagonal bias resulting from its right angled thread structure. Much of the literature says that triaxial weave does not appear very often outside of basket weaving. Bobbin lacemakers, however, know triaxial weave by the name *half-stitch* ground (see Figure 7.22(e)) and can attest that it appears very frequently in bobbin lace along with many interesting variations such as *Point de Paris* where the threads of the triaxial weave twist around one another (see Figure 7.22(b)). Half-stitch ground is just the very simplest example of the diverse and complex range of fabric structures bobbin lace can produce.

The challenge is to see past previous applications and to think of bobbin lace in terms of its characteristics. Unlike fixed frame weaving, threads are not restricted to the two classes of warp and weft. Threads can move in all possible directions, even backwards for short stretches as shown in Chapter 7. Bobbin lace should therefore be considered for applications in which the fibers travel in many directions such as the structure of muscle fibers, possibly even the smooth muscles used in peristaltic contractions of the digestive system or a network of sensors integrated into the shell of a car, the wing of an airplane or the sleeve of a shirt, for monitoring environmental conditions or vital signs. Lace also has the characteristic that it is very open fabric with holes of many shapes and sizes and could be considered for building materials capable of working in high wind or water flow situations without impeding the currents. If researchers are made aware of the potential in bobbin lace to create a diverse range of fabrics, they will most likely think of many more examples.

It is the author's hope that bobbin lace will continue to evolve and adapt; to serve current needs rather than become a footnote in the history of fashion. Like the following rhyme, once used by young lacemakers to keep track of their place in a pattern and still taught today to young children learning to count, let's preserve bobbin lace for future generations.

One, two,
Buckle my shoe;
Three, four,
Shut the door;
Five, six,
Pick up sticks;
Seven, eight,
Lay them straight.

-Anonymous

Appendix A

Additional Information

A.1 Enumeration of Toroidal k-ary Arrays

Interest in the one dimensional variation of toroidal arrays dates back at least as far 1872 and work by Moreau on necklaces [63]. A *k-ary necklace* of length n is an equivalence class of a string with n beads in which a bead can be any one of k colours. All rotations of the string are considered equivalent consistent with the analogy of rotating a necklace around your neck (Note: The necklace does not have a clasp so all bead positions are considered equivalent). The necklace can not be removed from your neck; that is the necklace can not be turned over or reflected. The equivalence class is typically represented by the lexicographically largest rotation. A counting function for necklaces of length n, attributed to MacMahon [56] in 1891, can be found using Burnside's Lemma and is shown in Lemma A.1.1.

Lemma A.1.1. *Number of k-ary necklaces of length n:*

$$T_k(n) = \frac{1}{n} \sum_{d \backslash n} \phi(d) k^{n/d} \tag{A.1}$$

where ϕ is the Euler totient function (A000010)

An enumeration of k-ary toroidal arrays for $2 \leq k \leq 8$ was reported in the OEIS by Hardin in 2011 (see Table A.1) but a counting function was not provided. A counting function for binary toroidal arrays, derived using the Pólya enumeration theorem, was provided by Ethier in 2013 [26].

2-ary	A184271[†]
3-ary	A184284
4-ary	A184277
5-ary	A184288
6-ary	A184291
7-ary	A184331
8-ary	A184294

[†] Entry includes a counting function for binary toroidal arrays from Ethier (2012)

Table A.1: OEIS entries for k-ary toroidal arrays with rotation allowed in columns and rows

OEIS entries for k-ary toroidal arrays with rotation allowed

Generalizing the result of Ethier, a counting function for k-ary arrays with cyclic symmetry is given in Lemma A.1.2.

Lemma A.1.2. *Number of k-ary $m \times n$ toroidal arrays:*

$$T_k(m,n) = \frac{1}{mn} \sum_{c \backslash m} \sum_{d \backslash n} \phi(c)\phi(d) k^{\frac{mn}{cd} gcd(c,d)} \tag{A.2}$$

where ϕ is the Euler totient function (A000010) and lcm stands for least common multiple.

Proof. A proof for $k = 2$ is given by Ethier [26] using the Pólya enumeration theorem. Because the number of bead colours does not affect the number of cycles of bead position, the proof can be extended to a necklace with k colours. □

Open Problem A.1.1. *Can the formula Ethier provided for binary toroidal arrays in which rotation AND reflection of the rows and columns are allowed (dihedral symmetry) be generalized for k-ary toroidal arrays?*

More relevant to the study of tesselace embeddings, the enumeration of prime k-ary toroidal arrays is relatively new territory. The one dimensional version of this problem, enumerating prime necklaces (also known as aperiodic necklaces or Lyndon words), has received considerable attention [44, 10, 74]. A counting function for prime necklaces, commonly referred to as *Moreau's necklace-counting function*, was proposed in 1872 [63] and is shown in Lemma A.1.3. A table of $P_2(n)$ results appears in OEIS entry A000048 and the more general $P_k(n)$ results appears in OEIS entry A074650.

Lemma A.1.3. *[Moreau's necklace-counting function] Number of k-ary prime necklaces of length n:*

$$P_k(n) = \frac{1}{n} \sum_{d \backslash n} \mu(n/d) k^d \tag{A.3}$$

where μ is the Möbius function

In the following description we make use of Möbius inversion.

Lemma A.1.4 (Möbius inversion formula). *Given two arithmetic functions f and g that satisfy the relation:*

$$g(n) = \sum_{d \backslash n} f(d) \text{ for all } n \geq 1$$

then

$$f(n) = \sum_{d \backslash n} \mu(d) g(n/d) \text{ for all } n \geq 1$$

Extending the counting function for prime necklaces to two dimensions, we derive the following theorem for toroidal arrays.

Theorem A.1.5. *Number of prime k-ary toroidal arrays of size $m \times n$:*

$$P_k(m, n) = \sum_{c \backslash m} \sum_{d \backslash n} \mu(c) \mu(d) T_k(m/c, n/d) \tag{A.4}$$

where μ is the Möbius function and $T_k(m, n)$ is given in Lemma A.1.2.

Open Problem A.1.2. *Theorem A.1.5 involves four summations. Is there a way to simplify this down to two?*

Proof. The total number of toroidal k-ary arrays of size $n \times m$ is equal to the sum over all combinations of prime arrays which can cover $n \times m$.

$$T_k(m,n) = \sum_{c\backslash m} \sum_{d\backslash n} P_k(c,d)$$

Define $f_1(c,n) = \sum_{d\backslash n} P_k(c,d)$. Then, by Möbius inversion,

$$\sum_{d\backslash n} P_k(m,d) = \sum_{c\backslash m} \mu(c) T_k(m/c,n)$$

Applying Möbius inversion a second time in the horizontal direction with $f_2(d) = P_k(m,d)$ gives our final result. □

The results of this enumeration, apart from the $m = 1$ case, do not appear in the OEIS. A sample for $k = 3$ and small values of n and m is shown in Table A.2.

row/column	1	2	3	4	5	6
1	3	3	8	18	48	116
2	3	18	116	810	5880	44220
3	8	116	2192	44220	956576	21522344
4	18	810	44220	2690028	174336264	11767874940
5	48	5880	956576	174336264	33891544512	6863037256208

Table A.2: Number of prime 3-ary toroidal arrays for small numbers of rows and columns

Bibliography

[1] A. Amir and G. Benson. "Two-dimensional periodicity in rectangular arrays". In: *SIAM Journal on Computing* 27.1 (1998), pp. 90–106.

[2] D. André. "Solution directe du probleme résolu par M. Bertrand". In: *Comptes Rendus Acad. Sci. Paris* 105 (1887), pp. 436–437.

[3] G. E. Andrews. "Euler's "exemplum memorabile inductionis fallacis" and q-trinomial coefficients". In: *J. Amer. Math. Soc.* 3.3 (1990), pp. 653–669.

[4] Arachne Lace Group. *Arachne mailing list*. http://www.arachne.com/lace/list_instructions.html. [Online; accessed 11-June-2016]. 2016.

[5] E. Artin. "The theory of braids". In: *American Scientist* 38.1 (1950), pp. 112–119.

[6] E. Barcucci, R. Pinzani, and R. Sprugnoli. "The Motzkin family". In: *Pure Math. Appl. Ser. A* 2.3-4 (1992), pp. 249–279.

[7] E. Barcucci et al. "ECO method and hill-free generalized Motzkin paths". In: *Sém. Lothar. Combin.* 46 (2001/02), Art. B 46b, 14 pp. (electronic).

[8] Y. Beeckman et al. *Moderne Gründe*. Übach-Palenberg, Germany: Deutscher Klöppelverband e.V., 2013.

[9] S.-M. Belcastro and C. Yackel. *Making mathematics with needlework*. Wellesley MA, USA: A.K. Peters Ltd, 2008.

[10] F. Bergeron, G. Labelle, and P. Leroux. *Combinatorial species and tree-like structures*. Cambridge University Press, 1998.

[11] M. Bousquet-Mélou. "Counting walks in the quarter plane". In: *Mathematics and Computer Science II*. Springer, 2002, pp. 49–67.

[12] P. L. Boyland, H. Aref, and M. Stremler. "Topological fluid mechanics of stirring". In: *Journal of Fluid Mechanics* 403 (2000), 277––304.

[13] G. Brinkmann. "Generating regular directed graphs". In: *Journal of Discrete Mathematics* 313.1 (2013), pp. 1 –7.

[14] J. Cantarella, M. Piatek, and E. Rawdon. "Visualizing the tightening of knots". In: *Visualization, 2005. VIS 05. IEEE*. IEEE. 2005, pp. 575–582.

[15] G. Chartrand, L. Lesniak, and P. Zhang. *Graphs & digraphs*. CRC Press, 2010.

[16] J. H. Conway, H. Burgiel, and C. Goodman-Strauss. *The Symmetries of Things*. Wellesley MA, USA: A.K. Peters, 2008, pp. 15–41.

[17] B. M. Cook and G. Stott. *The Book of Bobbin Lace Stitches*. London, UK: Charles T. Branford Co., 1980.

[18] H. S. Coxeter and W. Moser. *Generators and relations for discrete groups*. Springer-Verlag, 1964.

[19] É. C. De Verdière and A. de Mesmay. "Testing graph isotopy on surfaces". In: *Discrete & Computational Geometry* 51.1 (2014), pp. 171–206.

[20] É. C. De Verdière and A. de Mesmay. "Testing graph isotopy on surfaces". In: *Discrete & Computational Geometry* 51.1 (2014), pp. 171–206.

[21] M. Dehn. "Die gruppe der abbildungsklassen". In: *Acta Mathematica* 69.1 (1938), pp. 135–206.

[22] R. Donaghey and L. W. Shapiro. "Motzkin numbers". In: *J. Combinatorial Theory Ser. A* 23.3 (1977), pp. 291–301.

[23] G. Dye and A. Thunder. *Beginner's guide to bobbin lace*. Tunbridge Wells, UK: Search Press, 2008.

[24] P. Earnshaw. *Lace machines and machine laces, v. 2*. Gorse, Guildford, 1995.

[25] J. Edkins. *Jo Edkins' Bobbin Lace School*. http://gwydir.demon.co.uk/jo/lace/. [Online; accessed 11-June-2016]. 2014.

[26] S. Ethier. "Counting toroidal binary arrays". In: *Journal of Integer Sequences* 16.2 (2013), p. 3.

[27] R. Fathauer and N. Selikoff. *The 2015 JMM Exhibition of Mathematical Art*. Tessellations Publishing, 2015, p. 70.

[28] R. Fathauer and N. Selikoff. *The 2016 JMM Exhibition of Mathematical Art*. Tessellations Publishing, 2016, pp. 56–57.

[29] R. D. Fray and D. P. Roselle. "Weighted lattice paths". In: *Pacific J. Math.* 37 (1971), pp. 85–96.

[30] A. Gibbons. *Algorithmic graph theory*. Cambridge, UK: Cambridge University Press, 1985.

[31] S. Grishanov, V. Meshkov, and A. Omelchenko. "A topological study of textile structures. Part I: An introduction to Topological Methods". In: *Textile Research Journal* 79.8 (2009), pp. 702–713.

[32] B. Grunbaum and G.-C. Shephard. *Tilings and patterns*. WH Freeman and Company, Co-published by Akademie-Verlag, 1987.

[33] B. Grünbaum and G. C. Shephard. "Satins and twills: an introduction to the geometry of fabrics". In: *Mathematics Magazine* 53.3 (1980), pp. 139–161.

[34] L. Halley. *lynxlace.com*. http://www.lynxlace.com. [Online; accessed 11-June-2016]. 2015.

[35] A. Heath et al. "Transcatheter closure of large patent ductus arteriosus at high altitude with a novel nitinol device". In: *Catheterization and Cardiovascular Interventions* 79.3 (2012).

[36] A. F. von Henneberg. *The Art & Craft of Old Lace*. Lacis Publications, 1999.

[37] International Organization of Lace. *IOLI discussion forum*. http://laceioli.ning.com. [Online; accessed 11-June-2016]. 2016.

[38] IOLI. *International Organization of Lace, Inc.* http://www.internationalorganizationoflace.org/. [Online; accessed 11-June-2016]. 2016.

[39] V. Irvine. *InkScape Extension for Bobbin Lace Grounds*. https://tesselace.com/tools/inkscape-extension/. [Online; accessed 11-June-2016]. 2016.

[40] V. Irvine. "Matemáticas y bolillos". In: *Vuelta y Cruz* (13 2015), pp. 28–29.

[41] V. Irvine. *TesseLace*. http://tesselace.com/. [Online; accessed 11-June-2016]. 2016.

[42] K. Iwano and K. Steiglitz. "Planarity testing of doubly periodic infinite graphs". In: *Networks* 18.3 (1988), pp. 205–222.

[43] X. Y. Jiang and H. Bunke. "Including geometry in graph representations: A quadratic-time graph isomorphism algorithm and its applications". In: *Advances in Structural and Syntactical Pattern Recognition*. Springer, 1996, pp. 110–119.

[44] D. E. Knuth. "Generating all Tuples and Permutations; [Vol. 4A]". In: *The Art of Computer Programming*. Addison-Wesley, Reading, MA, 2011, Section 7.2.

[45] D. E. Knuth. "Introduction to Backtracking; [Vol. 4, Pre-fasicle 4B]". In: *The Art of Computer Programming*. Addison-Wesley, Reading, MA, 2016.

[46] D. E. Knuth. "Seminumerical Algorithms; [Vol. 2]". In: *The Art of Computer Programming*. Addison-Wesley, Reading, MA, 1998.

[47] E.-L. Kortelahti. *Nyplaättyä Pitsiaä*. Riste, Finland: Self published, 1981.

[48] J. Labelle and Y. N. Yeh. "Generalized Dyck paths". In: *Discrete Math.* 82.1 (1990), pp. 1–6.

[49] Lace for study. *Digital Lace Collection*. http://www.laceforstudy.org.uk/. [Online; accessed 11-June-2016]. 2016.

[50] E. Lamb. "The Stunning Symbiosis between Math and Knitting". In: *Scientific American* (2014).

[51] J. Leader. *An account of the history of lace making*. https://www.laceguild.org/craft/history.html. [Online; accessed 11-June-2016]. 2013.

[52] S. M. Levey. *Lace: A History*. London, UK: Victoria and Albert Museum, 1983.

[53] U. Löhr. *Viele Gute Gründe*. Bremen, Germany: Gabriele Zeller, 2002.

[54] R. C. Lyndon. *Groups and geometry*. Vol. 101. Cambridge University Press, 1985.

[55] M. Macauley. "Braids and juggling patterns". In: *Senior thesis, Department of Mathematics, Harvey Mudd College, Claremont, California* (2003).

[56] P. A. MacMahon. "Applications of a Theory of Permutations in Circular Procession to the Theory of Numbers". In: *Proceedings of the London Mathematical Society* 1.1 (1891), pp. 305–318.

[57] S. Marcus and D. Sokol. "2D Lyndon Words and Applications". In: *Algorithmica* (2013), pp. 1–18.

[58] R. P. Marenzana. "Yarn Limitations in the Manufacture of Barmen Torchon Laces". In: *The Melliand* 2.2 (1930).

[59] B. D. McKay. *Practical graph isomorphism*. Congr. Numer. 30, 1980, pp. 45–87.

[60] M. Meringer. "Fast generation of regular graphs and construction of cages". In: *Journal of Graph Theory* 30.2 (1999).

[61] S. G. Mohanty and B. R. Handa. "On lattice paths with several diagonal steps". In: *Canad. Math. Bull.* 11 (1968), pp. 537–545.

[62] B. Mohar and C. Thomassen. *Graphs on surfaces*. Vol. 10. JHU Press, 2001.

[63] C. Moreau. "Sur les permutations circulaires distinctes". In: *Nouvelles annales de mathématiques, journal des candidats aux écoles polytechnique et normale* 11 (1872), pp. 309–314.

[64] K. Murasugi and B. Kurpita. *A study of braids*. Vol. 484. Dordrecht, Netherlands: Kluwer Academic Publishers, 1999.

[65] N. Neff. *Arachne Lace Archives*. https://www.mail-archive.com/lace%40arachne.com/msg30855.html. 2009.

[66] H. Niederhausen. "Rota's umbral calculus and recursions". In: *Algebra Universalis* 49.4 (2003), pp. 435–457.

[67] P. Nottingham. *The Technique of Bobbin Lace*. London, UK: Batsford, 1976.

[68] OIDFA. *The International Bobbin and Needle Lace Organization*. http://www.oidfa.com/. [Online; accessed 11-June-2016]. 2016.

[69] A. Owczarek, R. Brak, and A. Rechnitzer. "Self-avoiding walks in slits and slabs with interactive walls". In: *Journal of Mathematical Chemistry* 45.1 (2009), pp. 113–128.

[70] J. Pol. *Diagrams for Bobbin Lace*. http://d-bl.github.io/. [Online; accessed 11-June-2016]. 2016.

[71] J. Pol. *Generator of Flanders meshes*. http://jo-pol.github.io/DiBL/flanders/. [Online; accessed 11-June-2016]. 2014.

[72] M. V. S. Ramanath and T. R. Walsh. "Enumeration and generation of a class of regular digraphs". In: *Journal of Graph Theory* 11.4 (1987), pp. 471–479.

[73] F. Ruskey. "Combinatorial generation". In: *Preliminary working draft. University of Victoria, Victoria, BC, Canada* 11 (2003), p. 20.

[74] F. Ruskey, C. Savage, and T. M. Y. Wang. "Generating necklaces". In: *Journal of Algorithms* 13.3 (1992), pp. 414–430.

[75] N. Selikoff. *Mathematical Art Galleries of the 2015 Bridges Conference*. http://gallery.bridgesmathart.org/exhibitions/2015-Bridges-Conference. [Online; accessed 11-June-2016]. 2015.

[76] R. Shepherd. "Lace Classification System". In: *Powerhouse Museum, Sydney* (2003).

[77] N. J. A. Sloane. "The on-line encyclopedia of integer sequences". In: *Ann. Math. Inform.* 41 (2013), pp. 219–234.

[78] R. P. Stanley. *Enumerative combinatorics; [Vol. 2]*. Vol. 62. Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge, 1999, pp. xii+581.

[79] L. Suchanek. *Lenka's Way of Lace*. https://lenkas.com/. [Online; accessed 11-June-2016]. 2016.

[80] R. A. Sulanke. "Moments of generalized Motzkin paths". In: *J. Integer Seq.* 3.1 (2000), Article 00.1.1, 1 HTML document.

[81] T. Sunada. "Lecture on topological crystallography". In: *Japanese Journal of Mathematics* 7.1 (2012), pp. 1–39.

[82] The Lace Guild. *The Lace Guild*. http://www.laceguild.org. [Online; accessed 11-June-2016]. 2016.

[83] The National Museum of American History. *Lace Collection*. http://americanhistory.si.edu/collections/object-groups/lace-collection. [Online; accessed 11-June-2016]. 2016.

[84] I. Theuerkauf. *Tierisch Flandrisch*. Melle, Germany: Self published, 2001.

[85] U. Ulrich. *Gründe mit System, 2nd edition*. Gammelby, Germany: Barbara Fay Verlag, 2011.

[86] E. J. J. Van Rensburg, T. Prellberg, and A. Rechnitzer. "Partially directed paths in a wedge". In: *Journal of Combinatorial Theory, Series A* 115.4 (2008), pp. 623–650.

[87] A.-M. Verbeke-Billiet. *Syllabus Binche I*. Bruges, Belgium: Kantcentrum, 1985.

[88] L. Waters. *Science and Lace*. http://scienceandlace.ning.com. [Online; accessed 11-June-2016]. 2016.

[89] S. Willard. *General Topology*. Addison-Wesley, 1970.