

# The Empire Problem in Penrose Tilings

by  
Laura Effinger-Dean

Duane Bailey, Advisor

A thesis submitted in partial fulfillment  
of the requirements for the  
Degree of Bachelor of Arts with Honors  
in Computer Science

WILLIAMS COLLEGE  
Williamstown, Massachusetts  
May 8, 2006

## Abstract

Nonperiodic tilings of the plane exhibit no translational symmetry. Penrose tilings are a remarkable class of nonperiodic tilings for which the set of prototiles consists of just two shapes. The *pentagrid* method, introduced by N.G. de Bruijn, allows us to generate Penrose tilings by taking a slice of the integer lattice in five-dimensional space. The empire problem asks: Given a subset of a Penrose tiling, what tiles appear in all tilings that include that subset? We present a new approach to the empire problem that uses the pentagrid method to identify elements of the empire.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Background</b>	<b>6</b>
2.1	Tilings and Aperiodicity . . . . .	6
2.2	Algebraic Methods . . . . .	12
2.2.1	Canonical Projection . . . . .	14
2.2.2	De Bruijn’s Pentagrids . . . . .	15
2.3	The Empire Problem . . . . .	16
<b>3</b>	<b>Vertex Configurations and Local Empires</b>	<b>20</b>
3.1	Vertex Configurations . . . . .	20
3.2	Constructing Local Empires . . . . .	21
3.3	Remotely Forced Tiles . . . . .	26
<b>4</b>	<b>Pentagrids</b>	<b>33</b>
4.1	Basic Definitions . . . . .	33
4.2	Constructing the Dual . . . . .	36
4.3	Vertex and Tile Placement . . . . .	41
4.4	Generalizations of Pentagrids . . . . .	43
4.4.1	Generalized Penrose Tilings . . . . .	43
4.4.2	Multigrids . . . . .	45
<b>5</b>	<b>An Algorithm for the Empire Problem</b>	<b>46</b>
5.1	Valid Sets . . . . .	46
5.2	Constraints for Valid Sets . . . . .	47
5.3	Testing Valid Set Inclusion . . . . .	49
5.4	Algorithm for a Single Tile . . . . .	51
5.5	Algorithm for the Empire . . . . .	51

<b>6</b>	<b>Conclusions and Future Work</b>	<b>53</b>
<b>A</b>	<b>Empires of the Eight Vertex Configurations</b>	<b>55</b>
<b>B</b>	<b>Proof of Theorem 5.2</b>	<b>64</b>

# Chapter 1

## Introduction

A tiling of the two-dimensional plane covers the plane with tiles, leaving neither gaps nor overlaps. Each tile is a rotation, translation, or reflection of one of several template shapes or *prototiles*. Most familiar tilings are *periodic*: if we translate the tiling by a certain distance, it will exactly match the original tiling. However, *aperiodic* sets of prototiles admit an infinite number of possible tilings, none of which are periodic. The well-known aperiodic Penrose tilings use just two prototiles to create an infinite number of distinct non-periodic tilings. [GS87]

Specific arrangements of tiles may *force* the presence of other tiles at different positions. Suppose we have some patch of tiles that we know to be part of a complete tiling. Without knowing the details of the entire tiling, we may be able to predict the locations of other tiles — even tiles that are very far away. This set of forced tiles for a specific patch is called the patch's *empire*. The *empire problem* asks:

*Given an initial patch of tiles  $P$ , what is the empire of  $P$ ?*

We consider this problem with respect to Penrose tilings, although it applies to any class of tilings. This thesis answers a question closely related to the empire problem:

*Given an initial patch of tiles  $P$  and another tile  $T$ , is  $T$  forced by  $P$ ?*

The tile  $T$  may be located anywhere in the tiling. Answering this question immediately leads to an algorithm for identifying finite subsets of the patch's empire.

Algorithms for identifying subsets or supersets of an initial patch’s empire in a Penrose tiling exist. In particular, Minnick [Min98], extending the work of Conway and Ammann, identified subsets of the empires by analyzing Ammann bars, a decoration for Penrose tiles realized as parallel lines that crisscross the tiling in five directions. Her algorithm predicted the locations of Ammann bars using a technique called canonical projection, in which points on the two-dimensional integer lattice are projected onto a line of irrational slope. However, Minnick herself identified a case in which canonical projection fails to identify forced tiles; other such exceptions may exist, casting doubt on the completeness of her algorithm.

Our approach will locate forced tiles using de Bruijn’s pentagrid method, an algebraic technique capable of generating all Penrose tilings. A pentagrid consists of five sets of parallel lines crisscrossing the plane at angles of  $\frac{2\pi}{5}$  to one another. Any pentagrid may be mapped to a Penrose tiling by thick and thin rhombs and vice versa. This method is equivalent to a projection method similar to the one used to generate musical sequences.

Just as Minnick was able to predict which Ammann bars would be forced using a projection from the integer lattice in two dimensions, it is possible to predict exactly which tiles will be forced using deBruijn’s higher-dimensional techniques. We present an algorithm for identifying forced tiles and compute the resulting empires for several important patches called *vertex configurations*. We expect that our results for Penrose tilings will extend to other tilings generated by projection methods, including tilings in three or more dimensions.

Chapter 2 overviews the essential concepts of Penrose tilings, and introduces the two main algebraic techniques for generating tilings. Chapter 3 discusses how to identify the *local empire* of a contiguous patch  $P$  — that is, the subset of the empire that is “next to”  $P$ . Chapter 4 explains de Bruijn’s pentagrid method in detail. Finally, Chapter 5 presents our algorithm for detecting forcing tiles, and Appendix A presents the results for the algorithm for various initial patches.

# Chapter 2

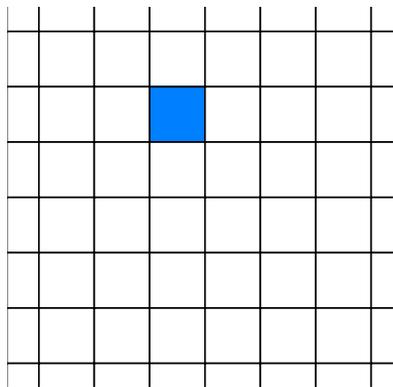
## Background

### 2.1 Tilings and Aperiodicity

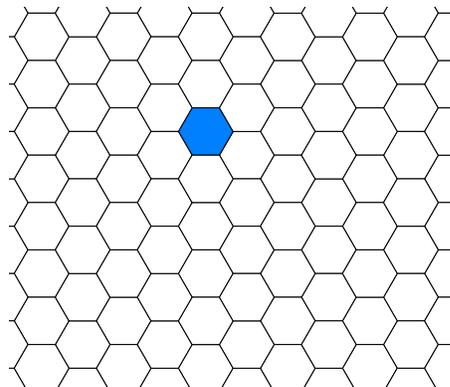
A tiling of the two-dimensional plane covers the plane with *tiles*, leaving neither gaps nor overlaps. Familiar tilings include those by regular polygons such as squares (Figure 2.1(a)) or hexagons (Figure 2.1(b)). Each tile is a translation, rotation, or reflection of one of a finite number of *prototiles*. For example, the set of prototiles for the tilings in Figures 2.1(a) and 2.1(b) are a single square and a single hexagon, respectively. We say that a set of prototiles *admits* a particular tiling if that tiling consists entirely of copies of those prototiles. These definitions naturally extend to three or more dimensions.

Notice the rigid structure of the tilings in Figure 2.1. Suppose we were to translate either tiling such that the shaded tile's new location exactly coincided with the original location of another tile. The resulting tiling would be identical to the original. We say that a tiling exhibits *translational symmetry* if there exists some nontrivial translation of the tiling that exactly matches up with the original. Other symmetries include *reflection* about a line or *rotation* about a point. The tiling by squares exhibits four-fold rotational symmetry — that is, a rotation through an angle of  $\frac{2\pi}{4} = 90^\circ$  yields the same tiling. The tiling by regular hexagons exhibits three-fold and six-fold rotational symmetry.

Tilings with translational symmetry are *periodic*: they have some pattern that repeats at regular intervals. In contrast, a tiling that exhibits no translational symmetry is *nonperiodic*. It is possible for a particular set of prototiles to admit both periodic and nonperiodic tilings — for instance, the tilings in Figure 2.2 are created from the same prototile, yet only Figure 2.2(a) is non-

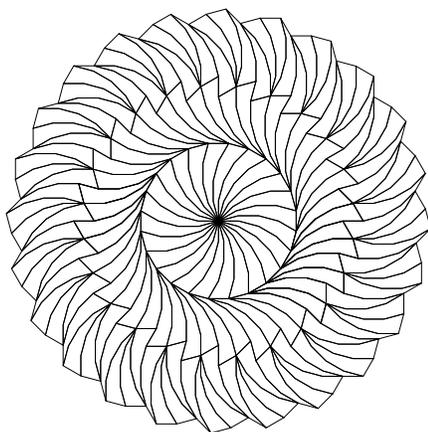


(a) A tiling by squares.

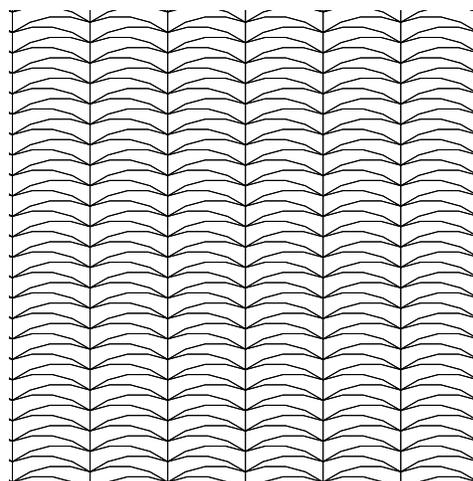


(b) A tiling by hexagons.

Figure 2.1: Tilings by regular polygons. In each, a single copy of the prototile is shaded.



(a) An example of a nonperiodic tiling.



(b) A periodic tiling with the same prototile.

Figure 2.2: The same set of prototiles may admit tilings both periodic and nonperiodic tilings. This particular prototile (a nine-sided polygon) and the spiral tiling in Figure 2.2(a) are from [GS87].

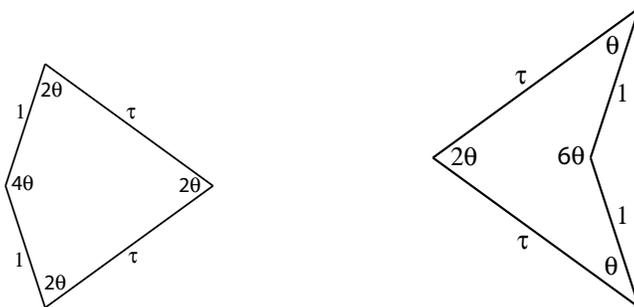


Figure 2.3: The Penrose kite and dart ( $\theta = \frac{\pi}{5}$ ).

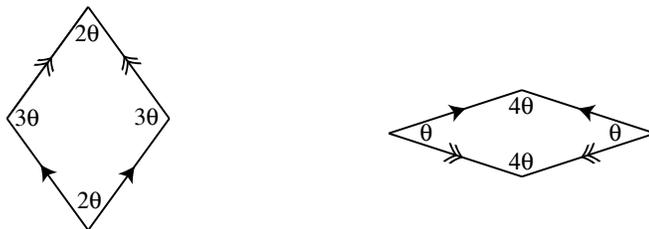


Figure 2.4: The Penrose rhombs ( $\theta = \frac{\pi}{5}$ ).

periodic. *Aperiodic* sets of prototiles admit an infinite number of different tilings, all of which are nonperiodic.

Tilings play an important role in crystallography, where each tile represents a “unit cell” in a crystal. The rules of crystallography for two dimensions allow one-, two-, three-, four-, or six-fold rotational symmetry, but forbid five-fold rotational symmetry and any symmetries higher than six-fold. (These “rules” may be derived using simple geometric arguments.) However, in 1984, researchers discovered that certain crystals exhibit icosahedral (twenty-fold) rotational symmetry, which is forbidden by the analogous rules of three-dimensional crystallography. This revelation shattered the existing assumption that crystals were periodic. Nonperiodic (or “quasiperiodic”) tilings emerged as a potential model for crystal structure.

In 1974, Penrose discovered a particularly elegant aperiodic set with just two prototiles, the kite and the dart. Both prototiles are quadrilaterals, with two sides of length  $\tau$  (the golden ratio,  $\frac{1+\sqrt{5}}{2}$ ) and two sides of length 1. Figure 2.3 gives the exact specifications. Tilings by kites and darts are not periodic,

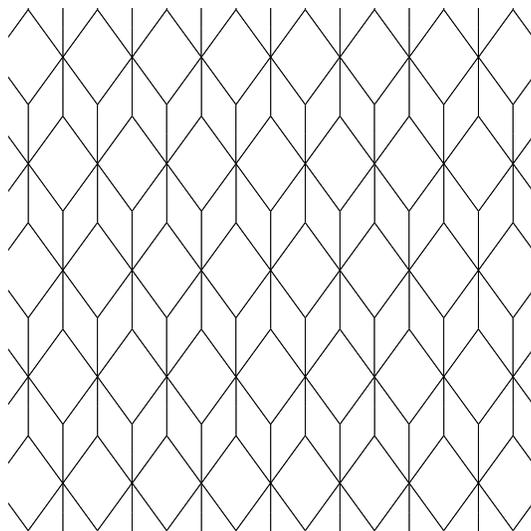


Figure 2.5: Without matching rules, the thick and thin rhombs admit periodic tilings of the plane.

and most have no rotational symmetry. However, all Penrose tilings exhibit five-fold rotational symmetry in finite patches, suggesting that they are relevant to quasicrystallography. Remarkably, Penrose discovered his class of tilings a decade before the appearance of quasicrystals, although the revelation that Penrose tilings had physical applications has certainly increased the level of general interest in their properties.

Tilings by kites and darts are equivalent to tilings by a second set of prototiles, the thick and thin rhombs. Each rhomb has four sides of unit length and angles that are multiples of  $\frac{\pi}{5}$ . Figure 2.4 gives the exact specifications. We will use rhombs instead of kites and darts for the remainder of this thesis, as their algebraic properties are extremely useful.

The thick and thin rhombs come with special *matching rules* that constrain how the tiles may fit together. We express these matching rules by decorating the tiles appropriately. In Figure 2.4 each edge has either one or two arrows; in a Penrose tiling the rhombs must be placed edge-to-edge such that each edge has the same number and direction of arrows. Without matching rules, the rhombs admit periodic tilings — for example, see the tiling in Figure 2.5.

An important distinction to make is that matching rules alone do not guarantee that the prototiles admit a complete tiling of the plane. It is easy

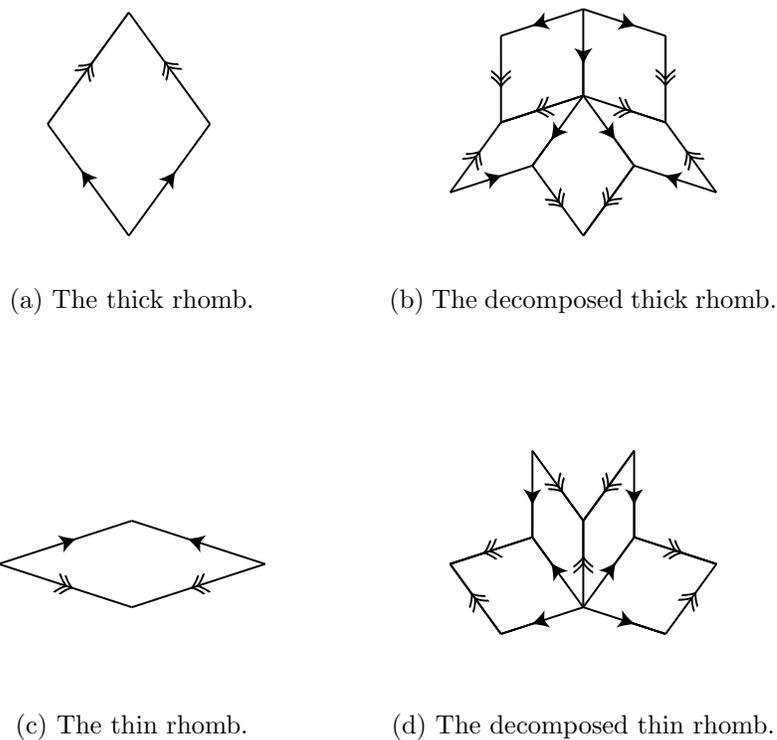


Figure 2.6: Substitution rules for the thick and thin rhombs.

to construct finite patches of tiles that obey matching rules but cannot be extended to a complete tiling. The matching rules guarantee that *if* we can find a complete tiling, then that tiling is nonperiodic.

Any tiling by thick and thin rhombs may be uniquely *decomposed* into a tiling by prototiles that are also thick and thin rhombs, but smaller by a factor of  $\tau$ . *Substitution rules* for decomposition are shown in Figure 2.6, and Figure 2.7 gives an example of decomposing a tiling. This mapping from a tiling to its decomposition is actually a bijection: we may uniquely *compose* any perfect tiling to produce a tiling by rhombs that are larger than the originals by a factor of  $\tau$ .

Decomposition provides a simple way of generating finite tilings: start with a single prototile, decompose it into smaller tiles, expand the prototiles to their original size, and repeat. In this way we may generate arbitrarily large (but

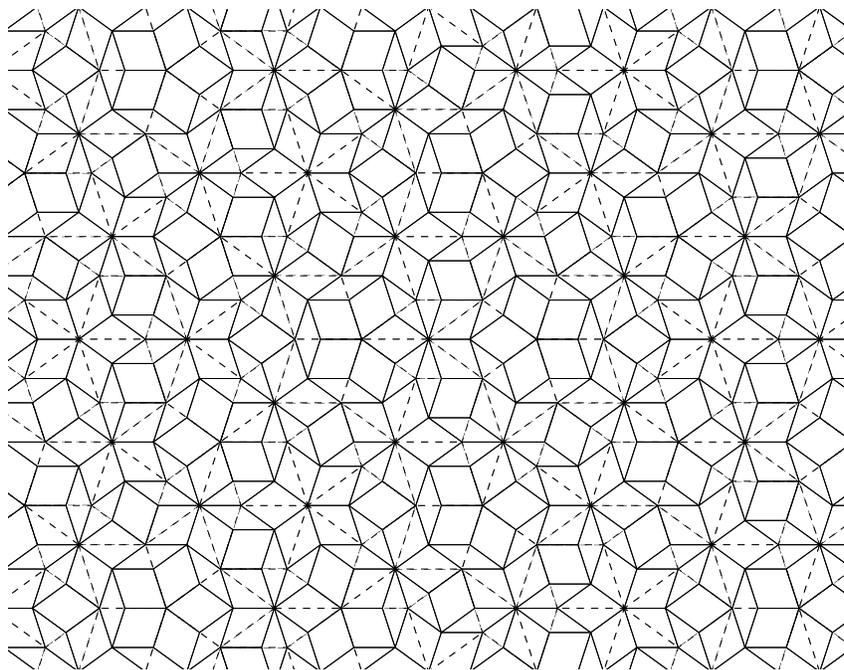


Figure 2.7: A tiling by rhombs and its decomposition. The original tiles have dashed edges.

finite) patches. As proved in [GS87], the ability to generate arbitrarily large finite patches guarantees that a complete tiling exists; therefore the Penrose rhombs admit a complete tiling of the plane.

One elegant property of Penrose tilings is *local isomorphism*. The local isomorphism property states that if a finite patch appears in a Penrose tiling, then it appears an infinite number of times in *every* Penrose tiling. Moreover, these identical patches are distributed evenly throughout the tiling. The finite patch may include an arbitrarily large number of tiles.

Ammann bars are an alternate method for specifying matching rules. Figure 2.8(a) shows the thick and thin rhombs decorated with Ammann bars. In a complete Penrose tiling, Ammann bars extend across the plane in five directions, with no kinks or holes (Figure 2.8(b) gives an example). The bars crisscross the tiling in five directions, separated by angles of  $\frac{2\pi}{5}$ .

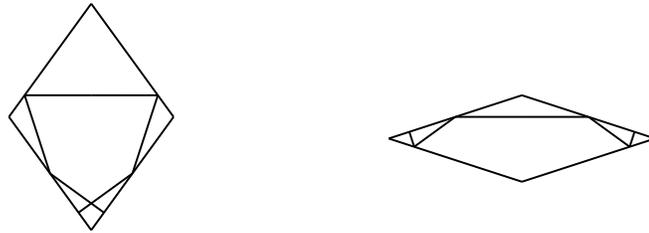
A fascinating property of Ammann bars (and one examined in detail by Minnick) are the patterns formed by the intervals between consecutive parallel bars. In Figure 2.8(b) we can see that each interval has one of two possible lengths, which we shall call L and S. The ratio of length L to length S is  $\tau$ . We will never find two consecutive S intervals or three consecutive L intervals. In fact, if we take just one set of parallel Ammann bars and translate it in the direction perpendicular to the bars, the sequence will never match the original exactly, regardless of how far we translate it. The intervals form a *musical sequence*, a one-dimensional aperiodic tiling. Many of the properties of Penrose tilings carry over to musical sequences, including composition/decomposition, local isomorphism. We can define substitution rules for decomposing musical sequences as follows:

$$\begin{aligned} L &\rightarrow LS \\ S &\rightarrow L \end{aligned}$$

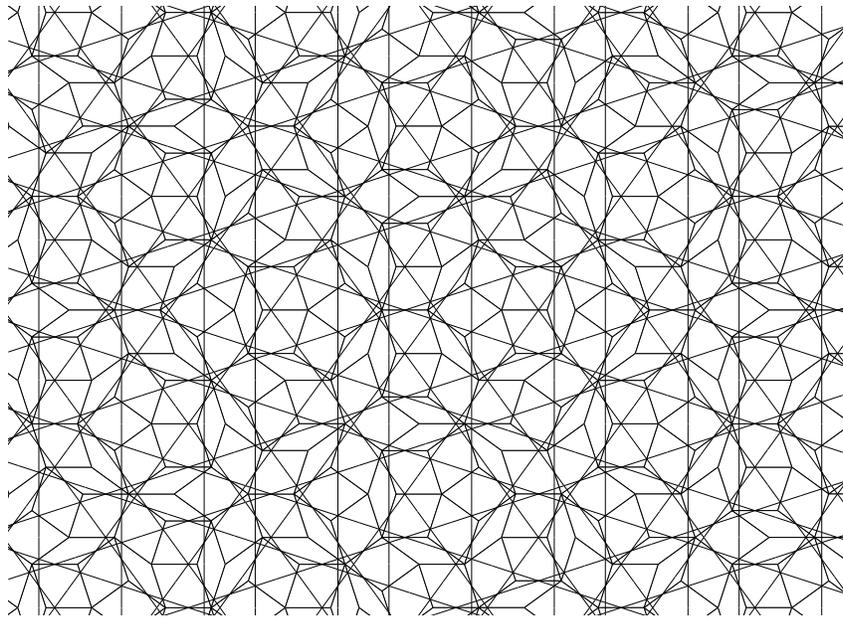
The algebraic techniques for generating tilings that we will examine in the next section are also useful for generating musical sequences.

## 2.2 Algebraic Methods

We now introduce two important methods for algebraically generating Penrose tilings. Both methods make use of the somewhat startling fact that aperiodic tilings are closely linked to *periodic* tilings in higher dimensions. The details presented below are based on the presentation Senechal [Sen95].



(a) The thick and thin rhombs decorated with Ammann bars.



(b) A tiling using Ammann bars.

Figure 2.8: Using Ammann bars for matching rules on the kite and dart.

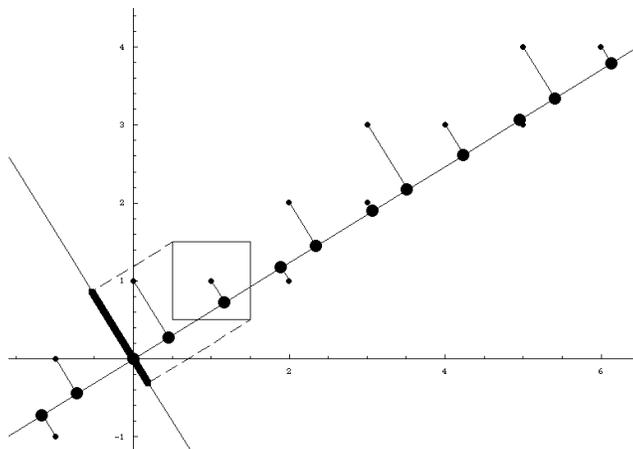


Figure 2.9: Canonical projection for the one-dimensional case. The intervals between the projected points form a musical sequence.

### 2.2.1 Canonical Projection

We introduce the technique of canonical projection by describing how to generate a one-dimensional nonperiodic tiling — namely, the musical sequences described in the last section. We project a subset of the two-dimensional integer lattice orthogonally onto a line of irrational slope. The resulting set of points (or rather, the sequence of intervals between each consecutive pair of points) forms a musical sequence. See Figure 2.9 for an example of using canonical projection to generate a musical sequence.

Let  $L$  be the line of slope  $\tau^{-1}$  that passes through the origin and let  $L^\perp$  be the line perpendicular to  $L$  that intersects  $L$  at the origin:

$$L = \left\{ (x, y) \mid y = \frac{x}{\tau}, x \in \mathbb{R} \right\}$$

$$L^\perp = \left\{ (x, y) \mid y = -\tau x, x \in \mathbb{R} \right\}$$

Now take the unit square at the origin,  $H$ , and translate it by some shift vector  $\vec{\gamma}$ :

$$H = \{ (x, y) \mid x, y \in [0, 1] \}$$

$$H + \vec{\gamma} = \{ (x, y) \mid x \in [\gamma_x, \gamma_x + 1] \text{ and } y \in [\gamma_y, \gamma_y + 1] \}$$

where  $\gamma_x$  and  $\gamma_y$  are the  $x$  and  $y$  components of  $\vec{\gamma}$ , respectively. We define a *window*  $W$ , which is the orthogonal projection of the translated unit square

onto  $L^\perp$ :

$$W = \Pi^\perp(H + \vec{\gamma})$$

$\Pi^\perp$  is the transformation that projects points onto  $L^\perp$ . Finally, take all points on the two-dimensional integer lattice whose projection onto  $L^\perp$  lies within the window  $W$ . These points, when orthogonally projected onto  $L$ , form the set of vertices  $V$  of a musical sequence.

$$V = \{ \Pi(x, y) \mid x, y \in \mathbb{Z} \text{ and } \Pi^\perp(x, y) \in W \},$$

where  $\Pi$  is the transformation that projects points orthogonally onto  $L$ .

This technique takes a periodic two-dimensional tiling — specifically, the tiling of the plane by unit squares — and maps a subset of the vertices of that tiling to a line of irrational slope. The one-dimensional variation of canonical projection is advantageous because it uses only two dimensions, and is therefore simple to visualize. The shift vector  $\vec{\gamma}$  dictates precisely *which* musical sequence is generated; it is possible to generate *all* musical sequences simply by changing the value of  $\vec{\gamma}$ .

For the Penrose tilings, we must project points from five-dimensional space ( $\mathbb{E}^5$ ). The mathematics of the five-dimensional projection method are quite similar to those described above for musical sequences. Instead of projecting points onto a line  $L$ , we project onto a plane  $\mathcal{E}$ .  $\mathcal{E}$  is a *totally irrational* subspace of  $\mathbb{E}^5$ : it intersects the five-dimensional integer lattice only at the origin. (In two dimensions, a line of irrational slope that passes through the origin is a totally irrational subspace.)  $L^\perp$  is now  $\mathcal{E}^\perp$ , the three-dimensional subspace of  $\mathbb{E}^5$  that is orthogonal to  $\mathcal{E}$ .  $H$  is the unit hypercube, and  $W$  is the projection onto  $\mathcal{E}^\perp$  of a translation (again by a shift vector  $\vec{\gamma}$ ) of  $H$ . We project onto  $\mathcal{E}$  those points whose projection onto  $\mathcal{E}^\perp$  lies within  $W$ . The projected points form the vertices of a Penrose tiling by thick and thin rhombs.

### 2.2.2 De Bruijn’s Pentagrids

De Bruijn’s seminal paper [dB81] describes the pentagrid method, in which the Penrose tilings appear as the *orthogonal duals* of another tiling. Two tilings  $\mathcal{T}$  and  $\mathcal{T}'$  are *dual* if

1. There exist one-to-one mappings from the vertices, edges, and faces of  $\mathcal{T}$  to the faces, edges, and vertices of  $\mathcal{T}'$ , respectively.

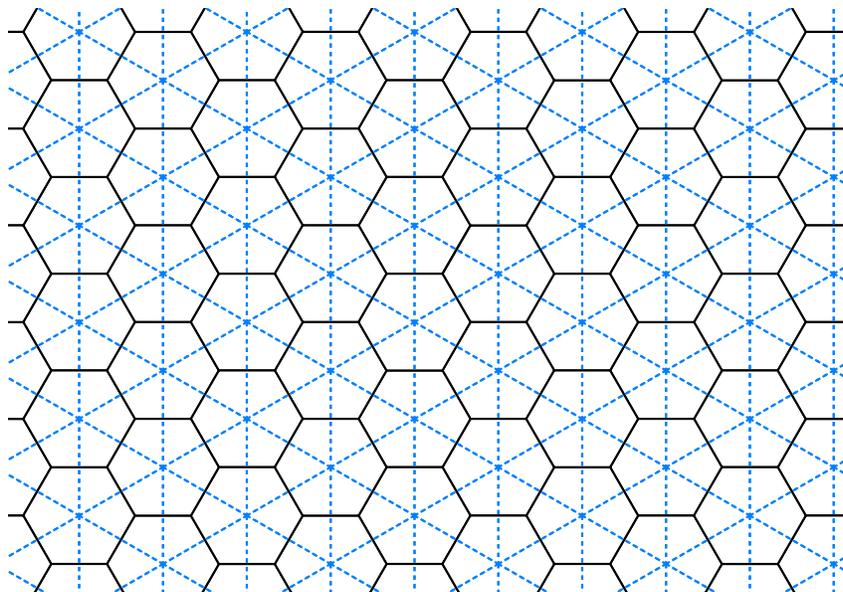


Figure 2.10: A tiling by hexagons and its orthogonal dual tiling by equilateral triangles (the dashed lines).

2. If a face  $\mathcal{F}$  in  $\mathcal{T}$  includes a vertex  $v$ , then the face corresponding to  $v$  in  $\mathcal{T}'$  includes the vertex corresponding to  $\mathcal{F}$ .

The tilings are *orthogonally dual* if each edge is perpendicular to its corresponding edge in the dual. Figure reffig:duals shows two dual tilings superimposed. We can construct a Penrose tiling by rhombs by finding an orthogonal dual of a pentagrid, a type of tiling created by superimposing five sets of parallel lines. The construction of the pentagrid and its dual is explained in detail in Chapter 4.

## 2.3 The Empire Problem

This thesis examines the problem of forcing in Penrose tiles. We say a patch of tiles  $P$  *forces* a tile  $T$  if  $T$  appears in the same position and orientation in all tilings in which  $P$  appears. The set of all tiles forced by a patch is called that patch's *empire*. The *empire problem* asks:

*Given an initial patch of tiles  $P$ , what is the empire of  $P$ ?*

Earlier we discussed Ammann bars, an alternative form of the matching rules for Penrose tiles. Conway and Ammann's work on the empire problem focused on patterns of Ammann bars. If the initial patch contains two or more parallel bars, then an infinite number of bars in that direction are forced across the tiling. If several forced bars intersect at just the right places, a tile is forced. [GS87, Gar95] For instance, Figure 2.11 is the empire of a particular kite and dart vertex configuration called the deuce, as presented in [GS87]. The darkened lines are the forced Ammann bars; when three forced bars form a particular isosceles triangle, a kite is forced.

Minnick's 1998 thesis extended Conway and Ammann's work on forced Ammann bars. She presented two algorithms for identifying forced bars. One algorithm exhaustively tested all possible musical sequences and identified their common elements; the other used a variation on the canonical projection method described in Section 2.2.1. However, she also noticed that this technique does not locate *all* forced bars. In particular, she noticed one case (which Conway overlooked) in which an Ammann bar is forced by an arrangement of two adjacent kites. The deuce's empire, for example, is not complete as presented in Figure 2.11. The red bars in Figure 2.12 are also forced, as are the red tiles.

The existence of Minnick's exceptional case throws into doubt the use of Ammann bars for attacking the empire problem. We cannot assume that Minnick's algorithms identify all forced bars; other cases like that of the two adjacent kites may exist. The limitation of this approach is that it addresses a one-dimensional version of a two-dimensional problem. Our approach will locate forced tiles by considering the tiling as a whole, instead of focusing on a particular musical sequence.

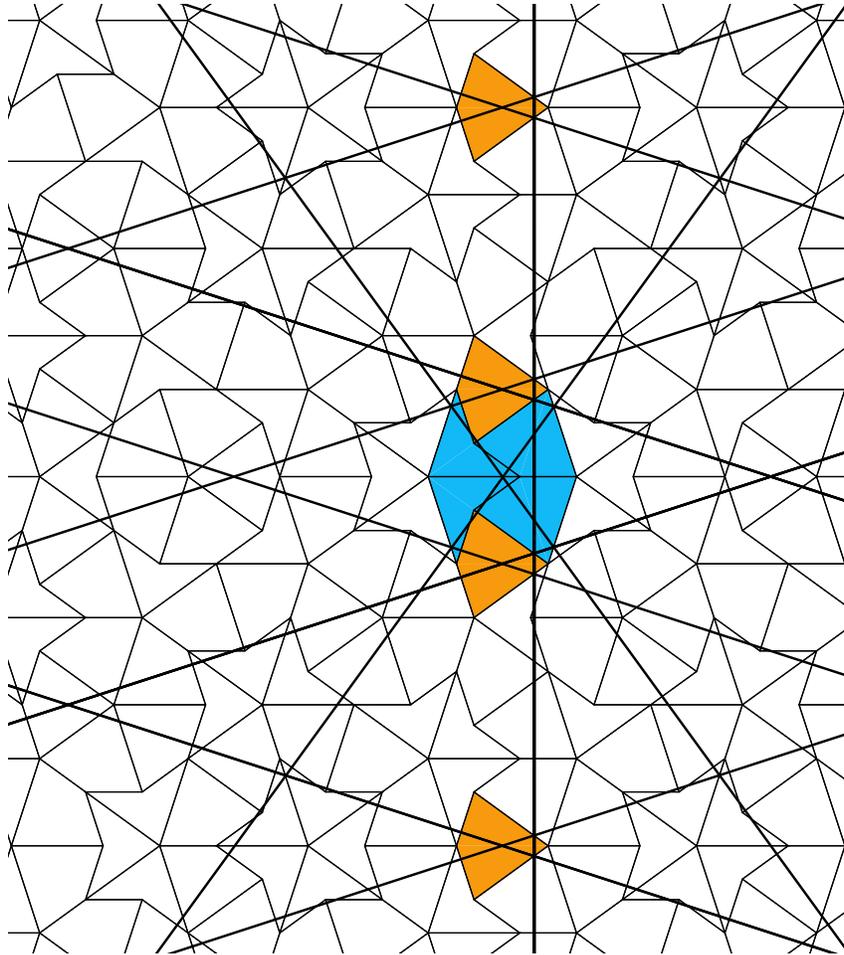


Figure 2.11: The empire of the deuce, as presented in [GS87].

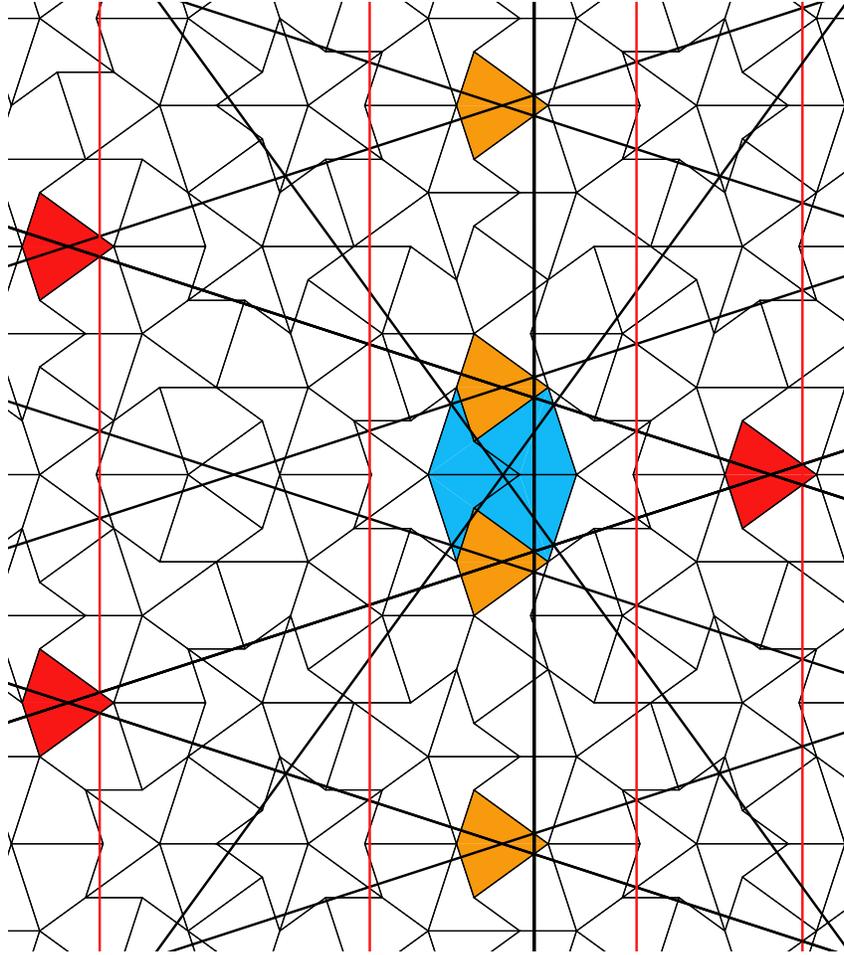


Figure 2.12: The empire of the deuce, as presented in [Min98].

## Chapter 3

# Vertex Configurations and Local Empires

Each vertex in a Penrose tiling by rhombs may be classified as one of eight different vertex configurations. These configurations allow us to identify forced tiles that are adjacent to an initial patch. In this section, we explain how to determine these local forcings and present results for each of the eight vertex configurations. We also suggest an inefficient algorithm for determining whether non-adjacent tiles are forced — that is, whether arbitrary tiles are part of the patch’s *empire*.

### 3.1 Vertex Configurations

The *neighborhood* of a vertex is the set of tiles incident to that vertex. De Bruijn [dB81] showed that the neighborhood of each vertex in a Penrose tiling will be congruent (up to rotation) to one of eight possible *vertex configurations*, all of which are shown in Figure 3.1. The names of these configurations are due to de Bruijn and derive from Conway’s names for the seven vertex configurations that appear in kite and dart tilings: king, queen, jack, deuce, ace, sun and star [Gar77].

Vertex configurations are an important example of the limitations of matching rules. We define a patch to be *legal* if and only if the patch appears in complete Penrose tilings. Recall from Chapter 2 that finite patches that obey matching rules are not guaranteed to be legal; rather, a complete tiling that obeys matching rules is guaranteed to be nonperiodic. Hence we must exercise

caution when considering finite patches; we cannot assume that a patch is legal unless we can find an example of the patch in a complete tiling.

Consider the vertex neighborhood in Figure 3.2. This finite patch obeys the matching rules, yet is not one of the eight legal vertex configurations. Indeed, if we try to extend the neighborhood with more tiles, it quickly becomes obvious that the neighborhood is not legal. Interestingly, we shall see in Section 4.4.1 that certain generalizations of the Penrose tilings *do* include examples of this neighborhood; for the class of tilings that this thesis considers, however, the neighborhood is illegal.

The eight vertex configurations prove to be important in two different ways:

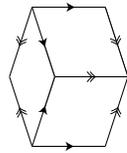
1. We defined empires in Chapter 2 in relation to some arbitrary “patch” of tiles. In most cases, this patch will be one of the eight vertex configurations. In more complex cases, patches may include several, perhaps non-contiguous, tiles or vertex configurations.
2. In Section 3.2, we will see that vertex configurations are useful when looking for locally forced tiles.

Sections 3.2 and 3.3 explain the technique of using vertex configurations to identify forced tiles.

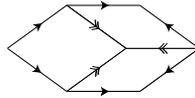
## 3.2 Constructing Local Empires

Let us explore the idea of using these vertex configurations to locate forced tiles. We see the step-by-step construction the local empire (defined below) of the Q in Figure 3.3. The construction process is as follows:

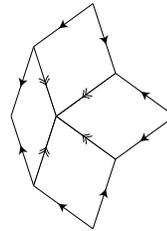
- (a) Start with the Q vertex configuration (Figure 3.3(a)).
- (b) The darkened vertex in Figure 3.3(b) is incomplete in the initial Q. However, only one vertex configuration, the J, includes both a  $108^\circ$  angle and a  $36^\circ$  angle. Therefore we can “fill in” the tiles of the J around that vertex.
- (c) Filling in the J forces two D configurations at the two darkened points in Figure 3.3(c).
- (d) The Q is symmetric about its horizontal axis, so the logic we used in steps (b) and (c) applies equally to the lower half of the configuration. In Figure 3.3(d) we have filled in forced tiles accordingly.



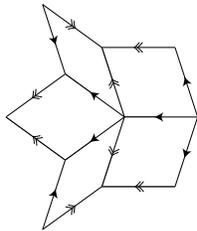
(a) The D.



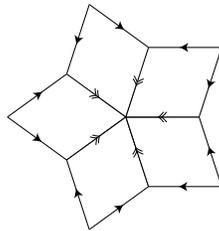
(b) The Q.



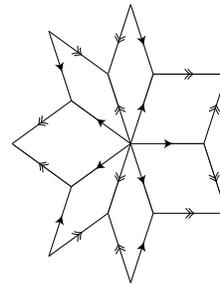
(c) The K.



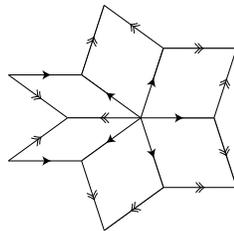
(d) The J.



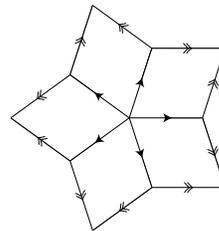
(e) The S.



(f) The S3.



(g) The S4.



(h) The S5.

Figure 3.1: The eight vertex configurations.

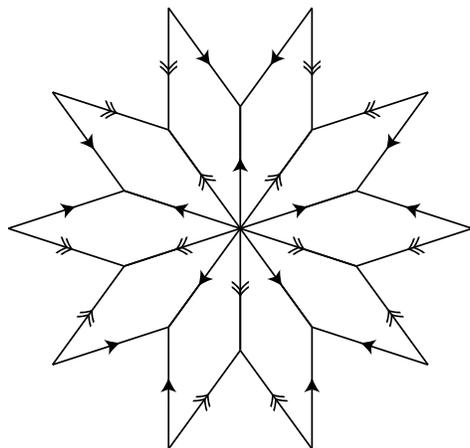
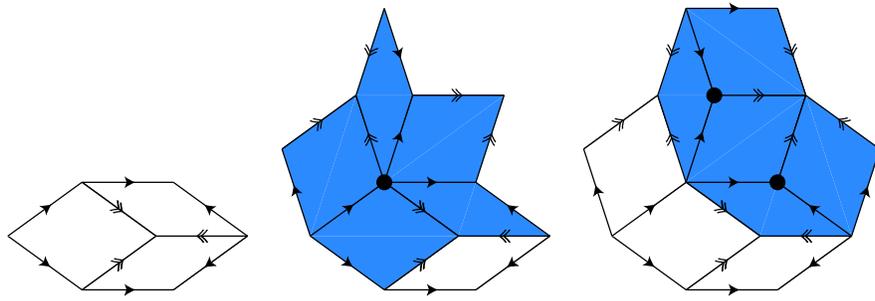


Figure 3.2: This vertex neighborhood is illegal, despite the fact that it obeys the matching rules.

- (e) None of the remaining incomplete vertices force a particular configuration, so we are done.

A *locally forced* tile is adjacent either to the initial patch or to another locally forced tile. The patch and its set of locally forced tiles form the *local empire*. If the initial patch is contiguous, then the local empire will also be a contiguous patch. The technique whereby we found the Q's local empire may be generalized to an algorithm for identifying local empires of arbitrary patches.

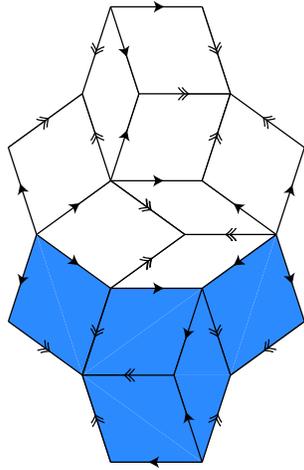
1. Start with some initial patch of tiles,  $P$ .
2. Consider an incomplete vertex in the patch. If the vertex matches exactly one of the eight possible vertex configurations, add the tiles in the configuration to  $P$ .
3. Repeat from step 2 until no incomplete vertices force a particular vertex configuration.



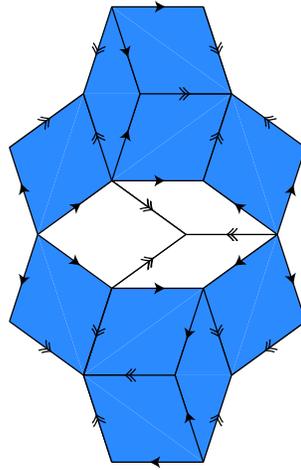
(a) The initial Q.

(b) Adding a forced J.

(c) Adding two forced D configurations.



(d) The lower half matches the upper half.



(e) The complete local empire.

Figure 3.3: Finding the local empire of the Q.

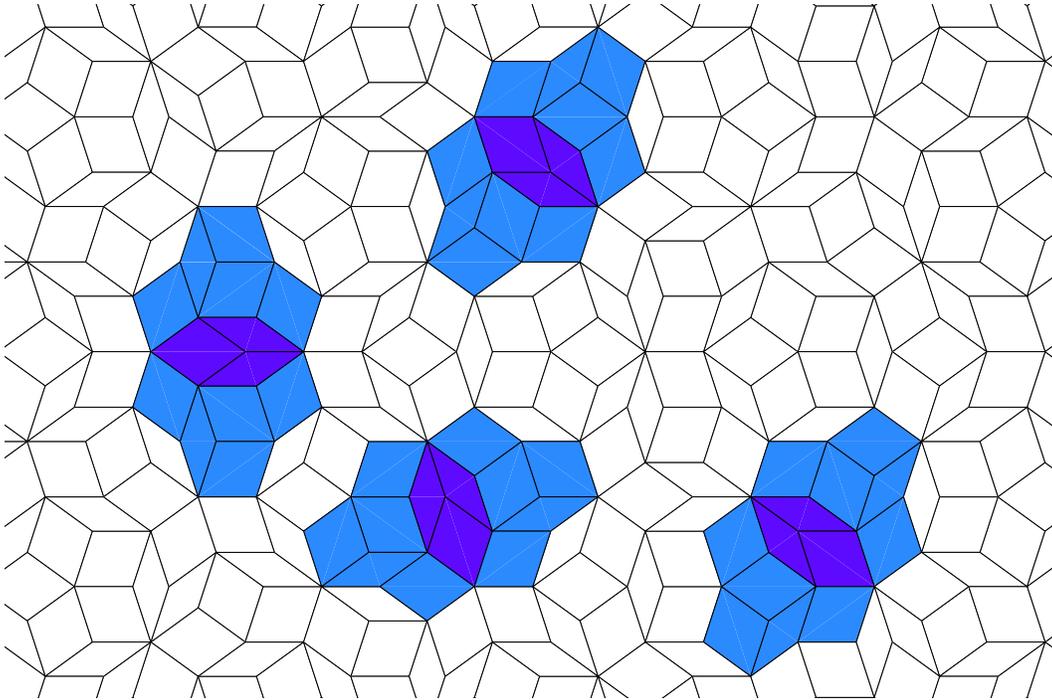


Figure 3.4: Several occurrences of the  $Q$  in a complete tiling. Notice that none of the tiles adjacent to the  $Q$ 's local empire universally agree.

This algorithm attempts to identify each incomplete vertex with a unique vertex configuration, incrementally filling in forced tiles.

However, there is no guarantee that the local empires found by this algorithm are complete. We have assumed that all finite patches are legal, and therefore may be considering more possibilities for vertex configurations than necessary. It is possible that the  $Q$ , for example, locally forces more tiles than shown in Figure 3.3(d). Usually we can confirm that the local empire is complete by comparing a few examples of the patch in complete tilings. For instance, several occurrences of the  $Q$  and its local empire are highlighted in Figure 3.4; it is easy to see that all tiles adjacent to the local empire are unforced. At the least, this algorithm provides a lower bound on the patch's local empire.

Each of the eight vertex configurations has a local empire that may be identified using the algorithm above<sup>1</sup>. We present the algorithmically constructed local empires for the D, the K, the J, the S, the S3, the S4, and the S5 in Figures 3.5 through 3.11.

### 3.3 Remotely Forced Tiles

We now have a method for identifying (potentially incomplete) local empires. However, our goal is to identify *all* forced tiles, not just those that are locally forced. In this Section, we present an extension of the local algorithm that determines whether non-local tiles are forced. We take as a precondition that the initial patch  $P$  falls within a circle of radius  $R$  centered at a given vertex.

1. Construct all possible finite completions of the patch  $P$  that fall within the circle of radius  $R$ . (There will be a finite number of such completions.) Represent each completion as a set  $S_i$  of tiles.
2. Suppose there are  $n$  possible completions,  $S_1 \dots S_n$ . Then the set of forced tiles is the intersection of these  $n$  sets:

$$S_1 \cap \dots \cap S_n$$

The intersection yields a subset of the patch's empire within a distance  $R$  of the center vertex. This approach is useful because it allows us to identify forced tiles that are not part of the local empire. We say that such tiles are *remotely forced*. However, the approach suffers from two critical weaknesses. As with the local algorithm in Section 3.2, there is no guarantee that the patches we construct will be legal arrangements (that is, arrangements that appear in complete tilings). As we check vertices out to only a finite distance, it is possible that at some distance further out we will discover that one or more of the possible completions has an inconsistency. Suppose all finite completions in which a tile  $T$  does *not* appear are illegal; in this case,  $T$  is forced, but the algorithm determines it to be unforced. Another weakness of the algorithm is its inefficiency. We require a large amount of space to store all of the possible finite completions. Suppose that, on average, each completion has  $m$  vertices

---

<sup>1</sup>We are not being entirely truthful here. In several cases, multiple vertex configurations fit around an incomplete vertex, but we can still fill in forced tiles because the possible configurations have one or more tiles in common.

within the circle of radius  $R$  ( $m$  is proportional to  $R^2$ ) and that each vertex may be completed in at most  $k$  ways. Then we must store  $O(m^k)$  arrangements.

In order to determine with certainty which tiles are forced, we must be able to distinguish legal arrangements from illegal arrangements. The solution is de Bruijn's pentagrids, a powerful method for producing legal Penrose tilings.

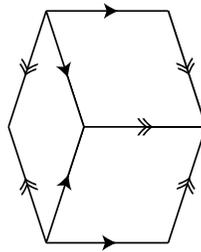


Figure 3.5: The D does not locally force any tiles, but we include it here for the sake of completeness.

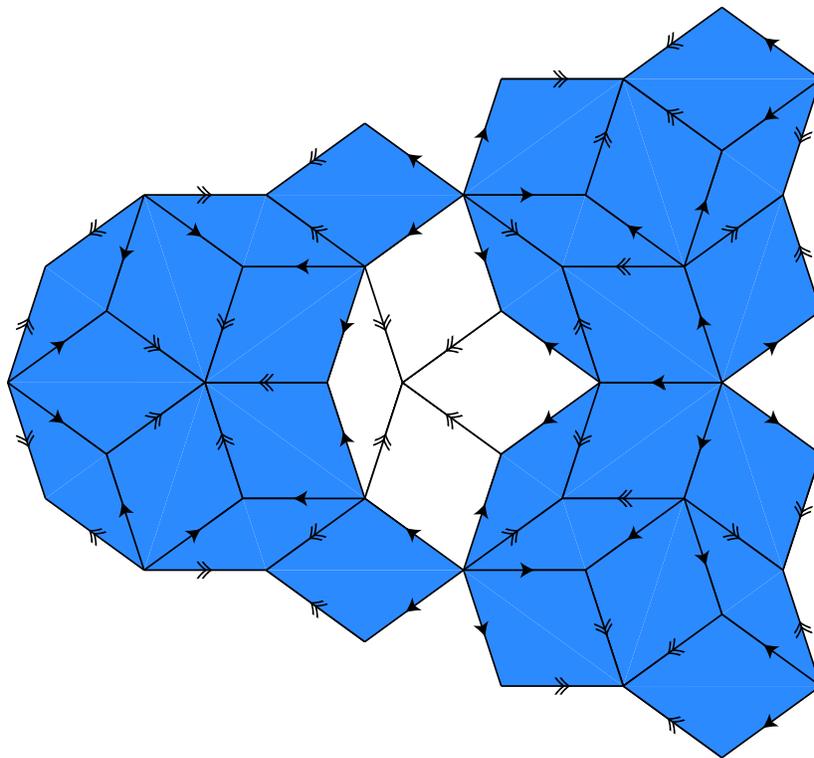


Figure 3.6: The K's local empire.

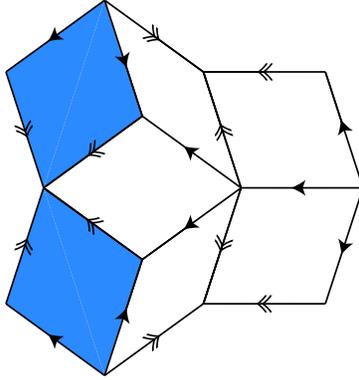


Figure 3.7: The J's local empire.

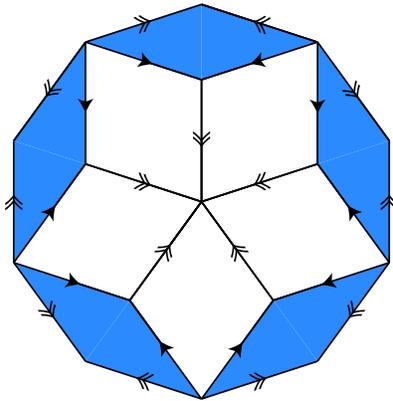


Figure 3.8: The S's local empire.

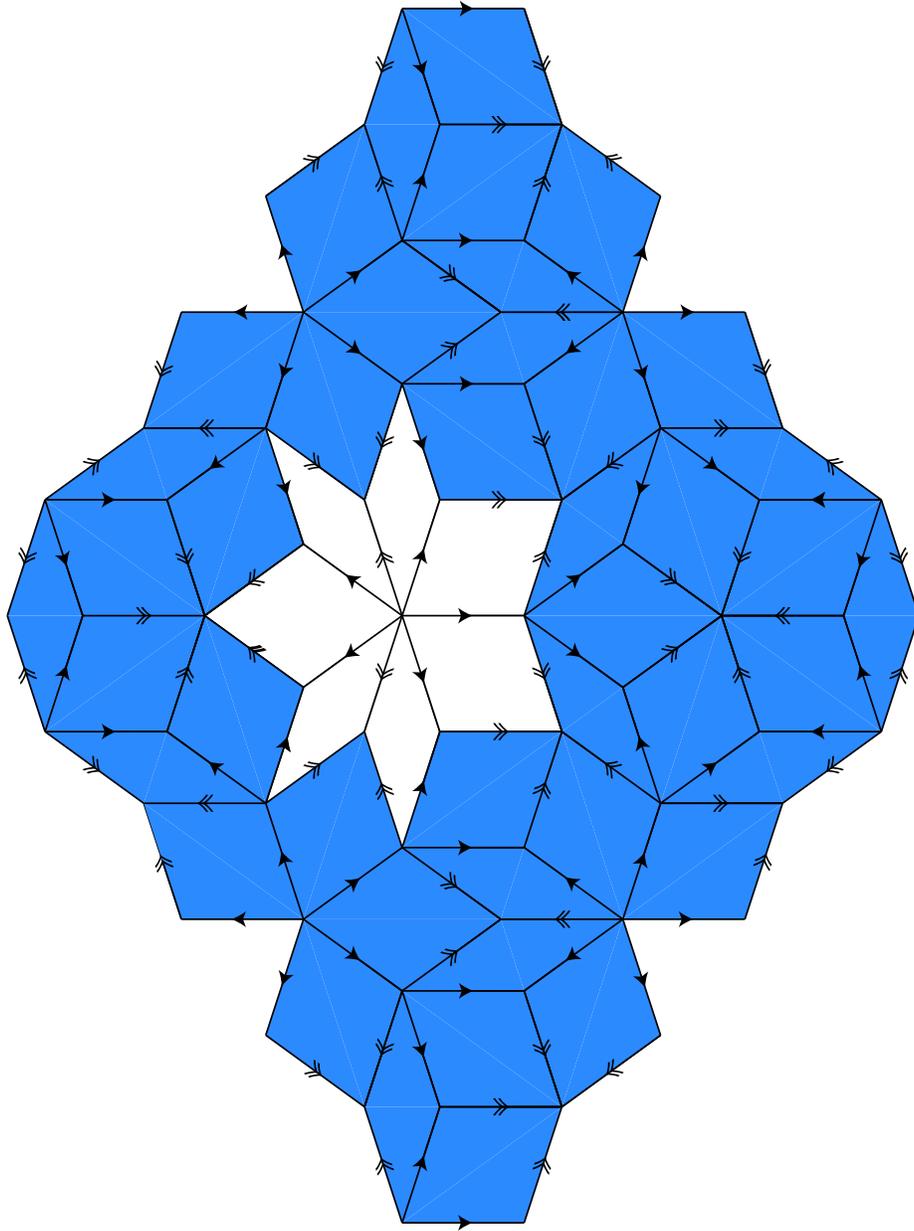


Figure 3.9: The  $S_3$ 's local empire.

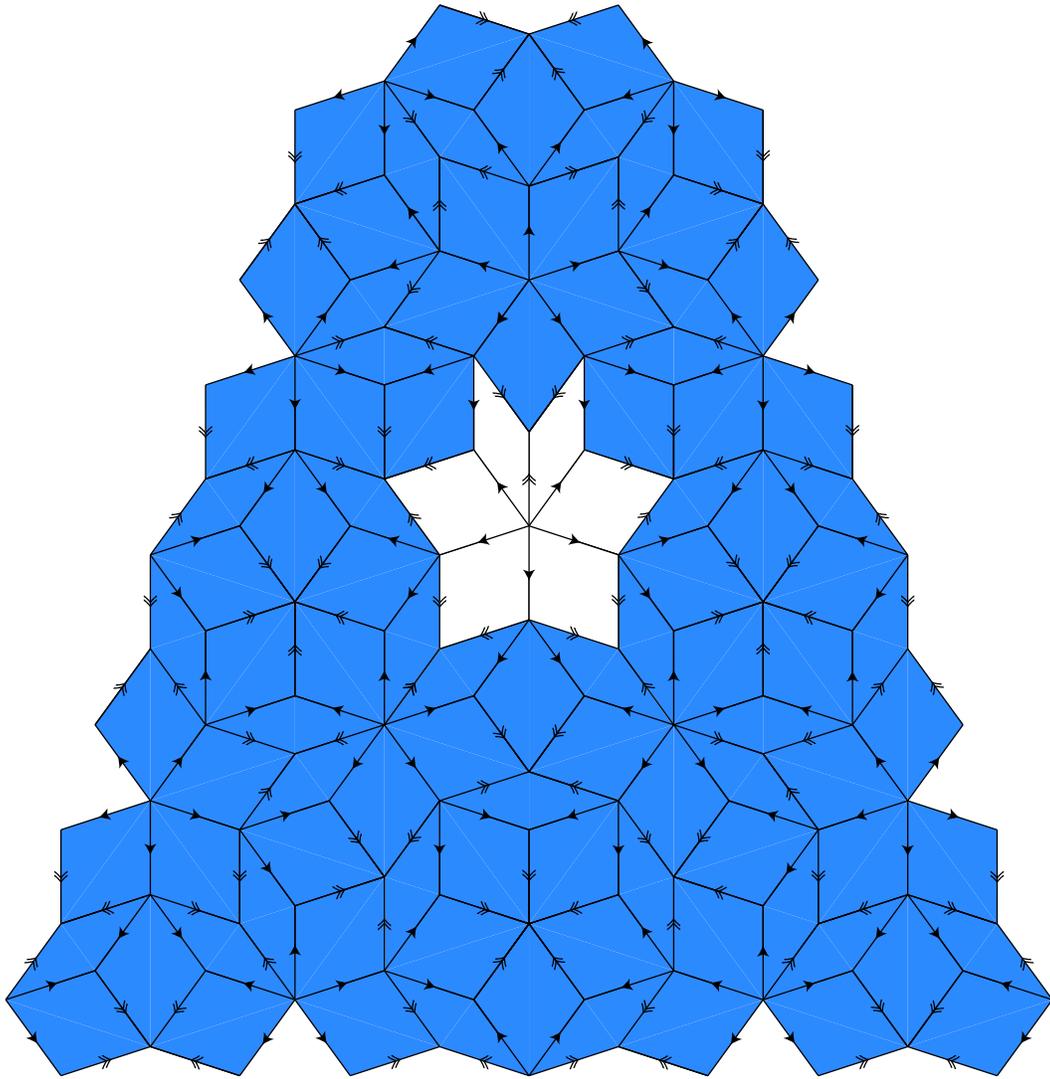


Figure 3.10: The  $S_4$ 's local empire.

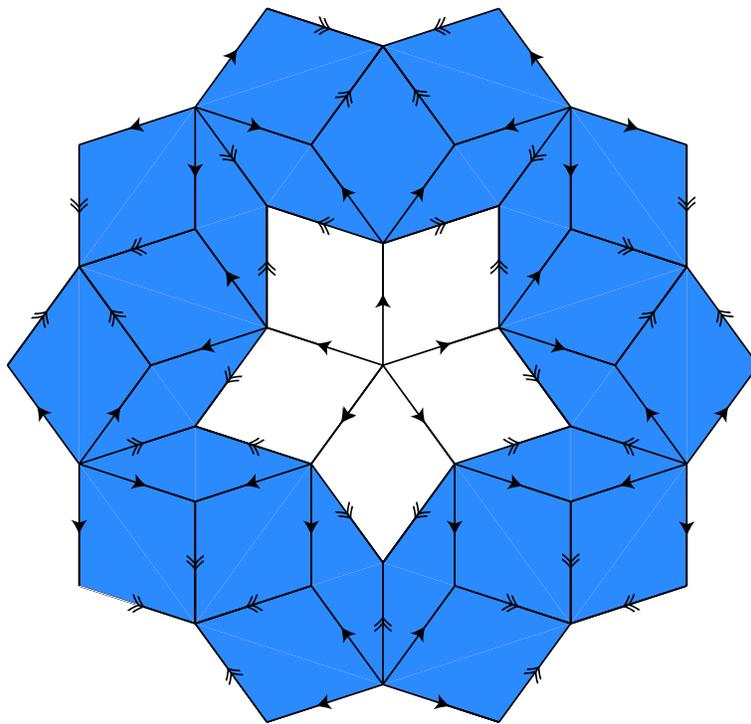


Figure 3.11: The S5's local empire.

# Chapter 4

## Pentagrids

In Chapter 2, we introduced the idea that Penrose tilings could be generated algebraically. This chapter focuses on pentagrids, one of the two algebraic methods discussed previously. Pentagrids were introduced by N. G. de Bruijn in a seminal 1981 paper, just a few years after Penrose tilings were discovered. The material in this chapter borrows heavily from de Bruijn's paper [dB81] as well as Marjorie Senechal's excellent text [Sen95].

### 4.1 Basic Definitions

A *grid* is an infinite collection of regularly spaced parallel lines. The points of the grid satisfy the *grid equation*:

$$\vec{x} \cdot \vec{e} + \gamma = k$$

for some integer  $k$ . Every line in the grid is perpendicular to a *grid vector*  $\vec{e}$ . The distance between consecutive lines is  $\frac{1}{|\vec{e}|}$ . The grid is shifted from the origin by a distance  $-\gamma$  in the direction of  $\vec{e}$ .

A *pentagrid* consists of five superimposed grids. We associate each grid with an integer  $j$  from 0 to 4, a grid vector  $\vec{e}_j$ , and a shift amount  $\gamma_j$ . We define  $\vec{e}_j$  such that the five grids are parallel to the sides of a regular pentagon, and consecutive grid lines are separated by a distance of 1:

$$\vec{e}_j = \left( \cos\left(\frac{2\pi j}{5}\right), \sin\left(\frac{2\pi j}{5}\right) \right) \quad (4.1)$$

Figure 4.1 is an example of a pentagrid.

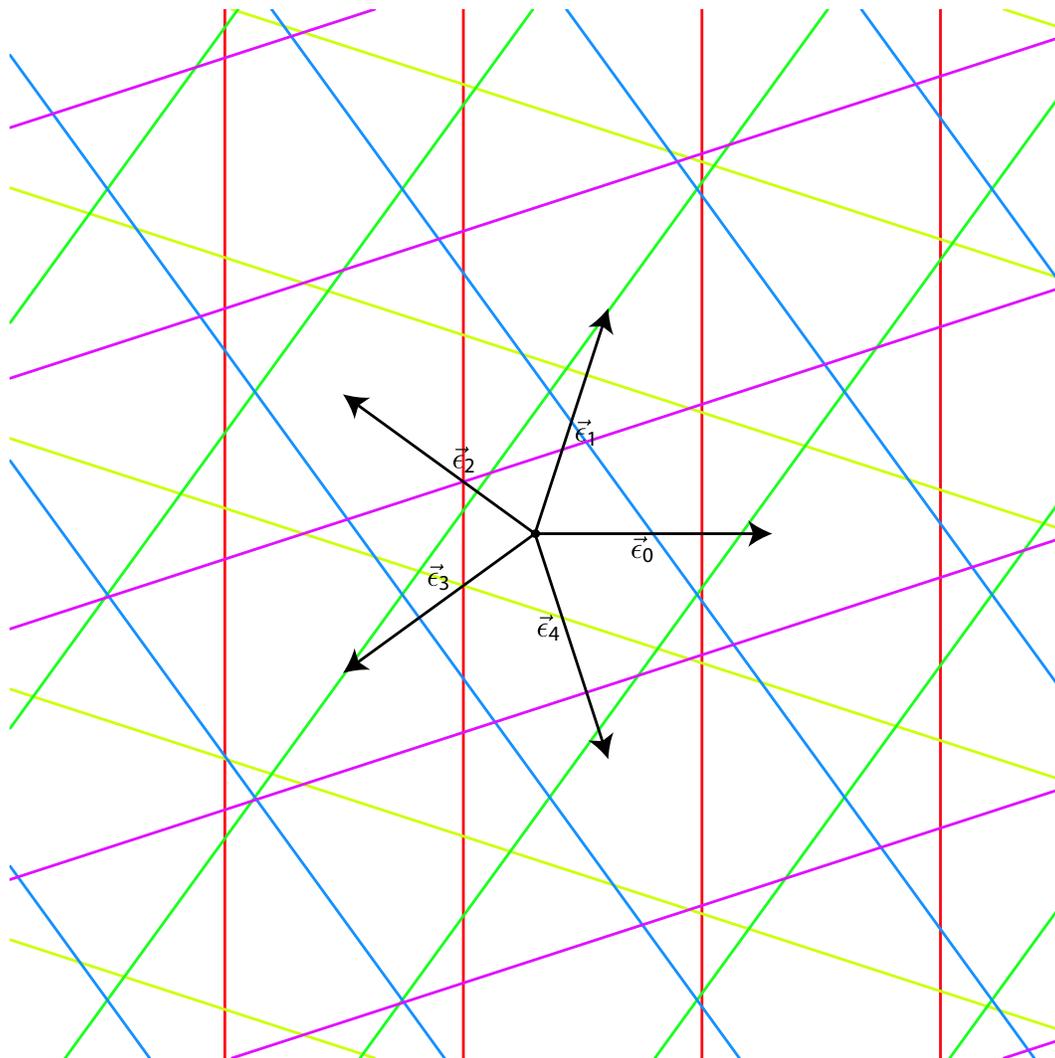


Figure 4.1: An example of a pentagrid.

Suppose that a line on grid  $j$  satisfies the grid equation

$$\vec{x} \cdot \vec{\epsilon}_j + \gamma_j = k_j.$$

We say that the line has *index*  $k_j$ .

We define the *shift vector* of a pentagrid to be a five-dimensional vector whose components are the shift amounts  $(\gamma_j)$  for each of the five grids:

$$\vec{\gamma} = (\gamma_0, \gamma_1, \gamma_2, \gamma_3, \gamma_4)$$

We construct a Penrose tiling from a pentagrid by finding the orthogonal dual of the pentagrid. Recall from Chapter one that two tilings  $\mathcal{T}$  and  $\mathcal{T}'$  are *dual* if

1. There exist one-to-one mappings from the vertices, edges, and faces of  $\mathcal{T}$  to the faces, edges, and vertices of  $\mathcal{T}'$ , respectively.
2. If a face  $\mathcal{F}$  in  $\mathcal{T}$  includes a vertex  $v$ , then the face corresponding to  $v$  in  $\mathcal{T}'$  includes the vertex corresponding to  $\mathcal{F}$ .

The tilings are *orthogonally dual* if each edge is perpendicular to its corresponding edge in the dual [GS87]. In Section 4.2, we give an example of how to construct the dual tiling.

De Bruijn proved that the orthogonal dual of a pentagrid is a Penrose tiling if the shift vector satisfies the *sum condition*:

$$\sum_{j=0}^4 \gamma_j = 0 \tag{4.2}$$

De Bruijn also showed that every Penrose tiling can be generated by a pentagrid that satisfies the sum condition.

One caveat is that the dual construction in Section 4.2 assumes that no more than two lines intersect at any point. A pentagrid that satisfies this condition is *regular*; a pentagrid in which three or more lines intersect at a single point is *singular*. (This restriction is unrelated to the sum condition; singular pentagrids may obey the sum condition, and pentagrids that obey the sum condition may be singular.) The orthogonal dual of a singular pentagrid includes tiles that are not rhombs. For instance, the tiling in Figure 4.2 is the dual of the pentagrid with  $\gamma_j = 0$  for all  $j$ . Note that the dual includes three shapes of tiles that do not appear in Penrose tilings. The two types of

hexagons correspond to threefold intersections, and the decagon corresponds to a fivefold intersection. Pentagrids that satisfy the sum condition will never have exactly four lines coincident at one point.

De Bruijn proposed that we interpret singular pentagrids as corresponding to multiple Penrose tilings. We may replace the decagon in Figure 4.2 with one of the ten possible rotations of the patch seen in Figure 4.3. The fat hexagon is replaced with one of two possible rotations of the Q (Figure 4.4(b)); the thin hexagon is replaced with one of two possible rotations of the D (Figure 4.4(a)). A pentagrid with a fivefold intersection corresponds to ten different Penrose tilings, while a singular pentagrid with no more than three lines coincident at any one point corresponds to two different Penrose tilings. Each of the two or ten possible tilings corresponds to an infinitesimal perturbation of the shift vector  $\vec{\gamma}$  such that the pentagrid is regular.

## 4.2 Constructing the Dual

Figure 4.5 illustrates the steps in constructing part of a Penrose tiling from a regular pentagrid.

- (a) Identify the intersections of the grid lines. In the figure, we have labeled five intersections.
- (b) At each intersection point, draw a rhomb whose edges are perpendicular to the intersecting lines.
- (c) Redraw the rhombs such that they lie edge-to-edge.
- (d) It is always possible to draw arrows such that the new tiles obey the matching rules. In this case, the five rhombs form a J, so we draw arrows that match Figure 3.1(d).

We also could have associated each face of the pentagrid with a vertex of the Penrose tiling and drawn edges between them; however, we believe that this approach is a little easier to visualize.

Every intersection in a pentagrid corresponds to a tile in the dual Penrose tiling. If the angles between the two intersecting lines are  $72^\circ$  and  $108^\circ$ , the corresponding tile will be a thick rhomb. If the angles between the two intersecting lines are  $36^\circ$  and  $144^\circ$ , the corresponding tile will be a thin rhomb.

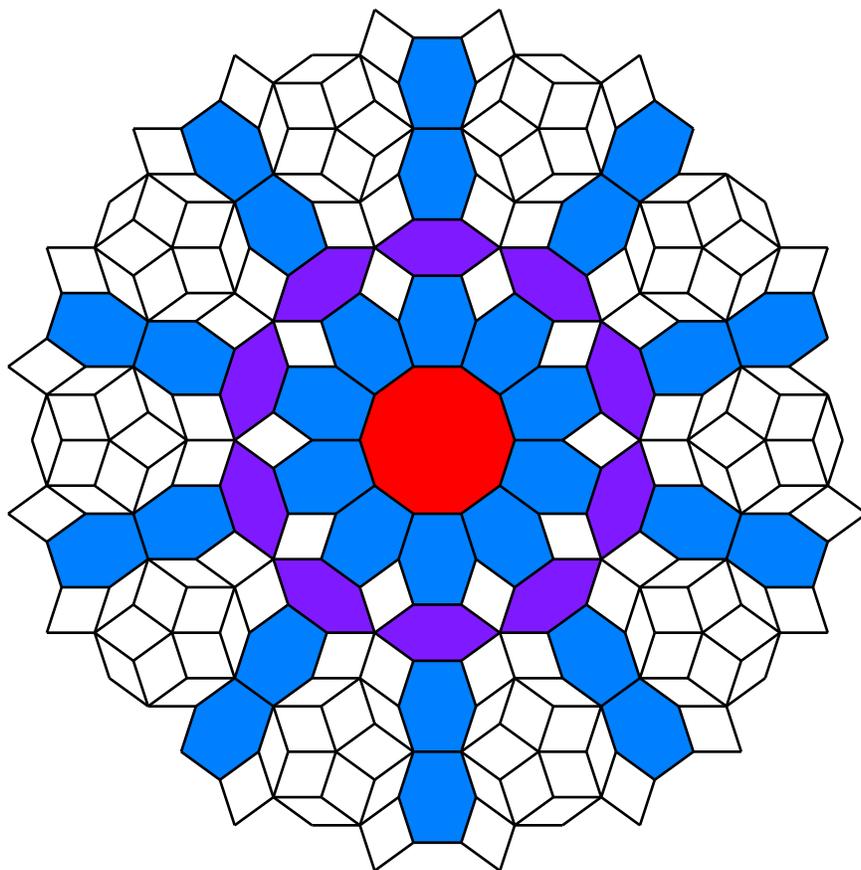


Figure 4.2: The orthogonal dual of the singular pentagrid with  $\gamma_j = 0$  for all  $j$ .

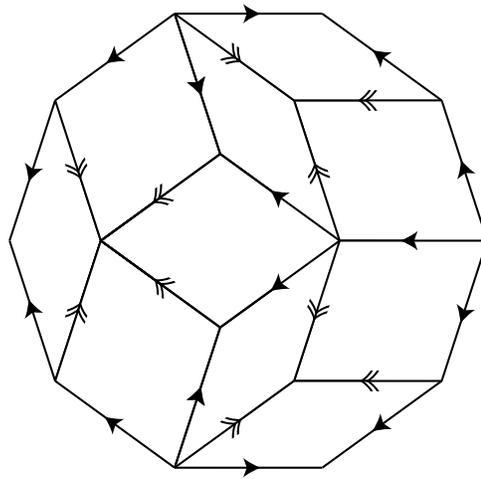
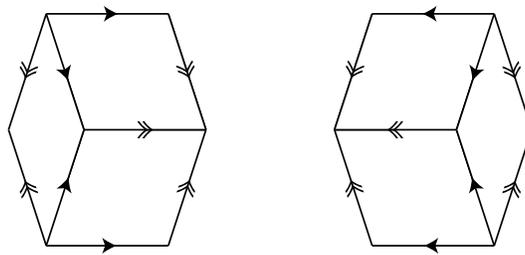
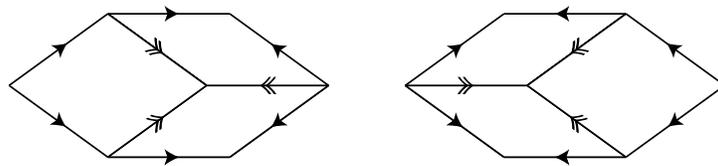


Figure 4.3: The decagon in Figure 4.2 corresponds to the ten possible rotations of this patch.

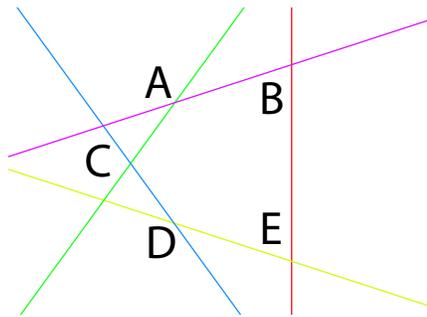


(a) The two possible replacements for the thin hexagon in Figure 4.2.

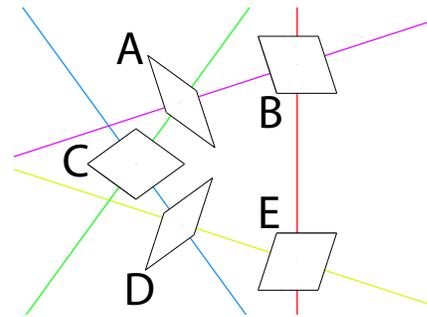


(b) The two possible replacements for the fat hexagon in Figure 4.2.

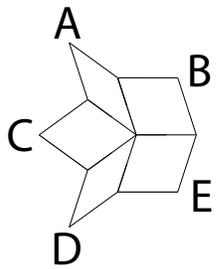
Figure 4.4: Patches corresponding to threefold intersections in singular pentagrids.



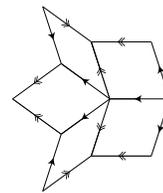
(a) Identify intersections.



(b) Attach rhombs.



(c) Place rhombs edge-to-edge.



(d) Draw arrows.

Figure 4.5: Constructing the dual of a pentagrid.

Adjacent tiles are correspond to consecutive intersections along the same grid line, which is always perpendicular to the shared edge.

Every face in the pentagrid corresponds to a vertex in the Penrose tiling. If a face is surrounded by  $n$  intersections, then the neighborhood of the corresponding vertex is a vertex configuration with  $n$  tiles. If two vertices in a Penrose tiling are connected by an edge, then the corresponding faces in the pentagrid must share an edge.

Although knowing how to construct the dual is useful for practical work with pentagrids, we need a stronger mathematical approach in order to reason about forcing. The next section puts the dual construction into mathematical terms.

### 4.3 Vertex and Tile Placement

Recall from Section 4.1 that a point  $(x, y)$  falls on line  $k$  in grid  $j$  if and only if

$$(x, y) \cdot \vec{e}_j + \gamma_j = k_j$$

It follows that a point  $(x, y)$  falls *between* lines  $k - 1$  and  $k$  if and only if

$$k - 1 < (x, y) \cdot \vec{e}_j + \gamma_j < k$$

Now consider a single face in the pentagrid. The face falls between two consecutive lines for each grid  $j$ . Hence  $[(x, y) \cdot \vec{e}_j + \gamma_j]$  is constant for every point internal to the face. Let that constant be  $k_j$  for each grid  $j$ ; then we may uniquely identify each face in the pentagrid with a five-tuple of integers  $(k_0, k_1, k_2, k_3, k_4)$ .

Each face corresponds to a vertex in the pentagrid's dual rhomb tiling. If the five-tuple associated with a face is  $(k_0, k_1, k_2, k_3, k_4)$ , then the coordinates of the corresponding vertex are

$$\sum_{j=0}^4 k_j \left( \cos \left( \frac{2\pi j}{5} \right), \sin \left( \frac{2\pi j}{5} \right) \right)$$

where  $(x, y)$  are the coordinates of any point in the face. Note that we are imposing Euclidean coordinates on the Penrose tiling. If we want to take an algebraic approach to forcing, we need some sort of coordinates in order to specify the locations of tiles and vertices.

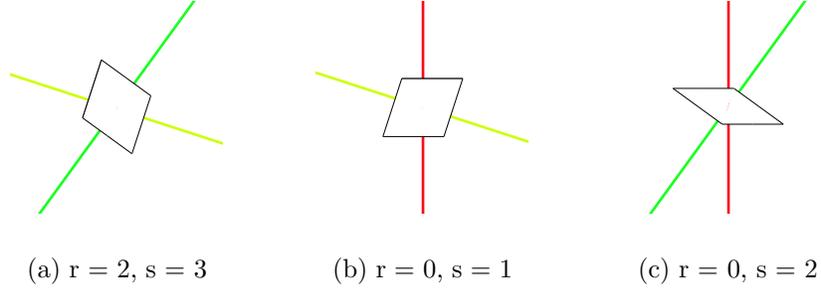


Figure 4.6: Examples of intersection types and the shapes and orientations of the corresponding rhombs in the dual Penrose tiling.

Recall that each tile in the Penrose tiling corresponds with an intersection in the pentagrid, and vice versa. Suppose that a tile  $T$  corresponds to the intersection of lines  $k_r$  and  $k_s$  in grids  $r$  and  $s$ , respectively, where  $0 \leq r < s \leq 4$ . ( $r$  and  $s$  determine the shape and orientation of the tile, as in Figure 4.6.) Let  $\vec{x}$  be the coordinates of the intersection. By [dB81], the four vertices of the tile are associated with the five-tuples

$$(K_0(\vec{x}), K_1(\vec{x}), K_2(\vec{x}), K_3(\vec{x}), K_4(\vec{x})) + \varepsilon_1(\delta_{0r}, \dots, \delta_{4r}) + \varepsilon_2(\delta_{0s}, \dots, \delta_{4s})$$

where

$$\begin{aligned} K_j(\vec{x}) &= \lceil \vec{x} \cdot \epsilon_j + \gamma_j \rceil \\ (\varepsilon_1, \varepsilon_2) &\in \{(0, 0), (0, 1), (1, 0), (0, 1)\} \end{aligned}$$

and  $\delta_{ij}$  is the Kronecker delta:  $\delta_{ij} = 1$  if  $i = j$  and 0 otherwise. Note that  $K_r(\vec{x}) = k_r$  and  $K_s(\vec{x}) = k_s$ . For each integer  $t \neq r, s$  such that  $0 \leq t \leq 4$ , define

$$k_t = K_t(\vec{x}) = \lceil \vec{x} \cdot \epsilon_t + \gamma_t \rceil \quad (4.3)$$

Then each tile is associated with two grid numbers,  $r$  and  $s$ , and a five-tuple  $\vec{k} = (k_0, k_1, k_2, k_3, k_4)$ . We say that  $T = (r, s, \vec{k})$ .

Note that we do not need to specify the number and direction of arrows on the tile's edges. Recall from Chapter 2 that the arrows are one of several methods of ensuring that a tiling is in fact a Penrose tiling. The sum condition (equation 4.2) is another such method. We may draw arrows on the tilings for aesthetic purposes, but they are not strictly necessary as long as the pentagrid meets the sum condition.

## 4.4 Generalizations of Pentagrids

In this section we consider two generalizations of the pentagrid method, both of which lead to interesting variations on the standard Penrose tilings. We expect that our analysis of forcing in Chapter 5 in typical Penrose tilings will also apply to these generalizations.

### 4.4.1 Generalized Penrose Tilings

Suppose we were to disobey the sum condition by using a shift vector  $\vec{\gamma}$  such that

$$\sum_j \gamma_j = \frac{1}{2}$$

The result (Figure 4.7) is interesting: we obtain a tiling by thick and thin rhombs that superficially resembles normal Penrose tilings, but is clearly a different sort of tiling. We cannot draw arrows on these rhombs in a way that obeys the matching rules set out in Chapter 2. We can also find vertices (several of which are darkened) whose neighborhoods do not match the eight legal configurations (Figure 3.1). These new vertex configurations include the patch that we used as an example of illegal patches in Chapter 3 (Figure 3.2).

We define *generalized Penrose tilings* to be those tilings which may be generated by constructing the dual of pentagrids. The Penrose tilings that we have discussed prior to this section are the simplest subclass of this larger class of tilings. In particular, they are the only type of Penrose tiling with “simple” matching rules; for any other value of  $\sum \gamma_j$ , there will be multiple possible decorations for each type of tile.

Socolar and Steinhardt [SS86] classify generalized Penrose tilings as *local isomorphism classes*. Recall the property of local isomorphism defined in Chapter 2: any finite patch that appears in any Penrose tiling appears an infinite number of times in every Penrose tiling. We say that two tilings are locally isomorphic if every finite patch of tiles in each occurs in the other. It turns out that two generalized Penrose tilings with shift vectors  $\vec{\gamma}$  and  $\vec{\gamma}'$  are locally isomorphic if and only if

$$\sum_j \gamma_j \equiv \sum_j \gamma'_j \pmod{1}$$

Hence we may decompose the class of generalized Penrose tilings into an uncountable number of *local isomorphism classes*.

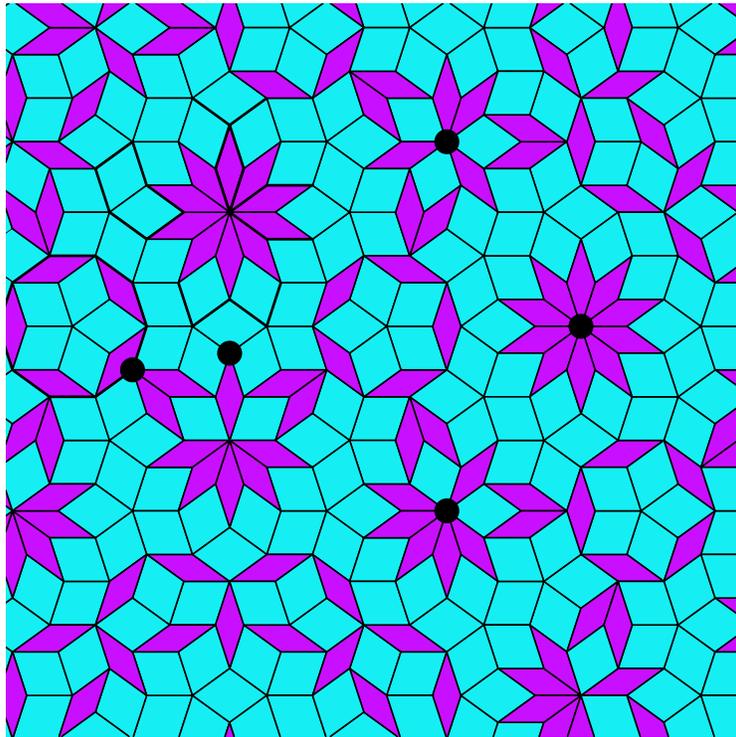


Figure 4.7: A tiling generated as the dual of a pentagrid with  $\sum \gamma_j = \frac{1}{2}$ .

## 4.4.2 Multigrids

De Bruijn's 1986 paper [dB86] generalizes the pentagrid method to multiple dimensions. The *multigrid method* constructs an  $n$ -dimensional tiling as the orthogonal dual of  $m$  grids in  $n$ -dimensional space. For pentagrids,  $n = 2$  and  $m = 5$ . We assume that  $m < n$ .

Each grid is an infinite collection of regularly spaced  $(n - 1)$ -dimensional hyperplanes. (If  $n = 2$ , an  $(n - 1)$ -dimensional hyperplane is a line; if  $n = 3$ , an  $(n - 1)$ -dimensional hyperplane is a plane, and so on.) A point  $\vec{x}$  on the grid satisfies  $\vec{e}_j \cdot \vec{x} + \gamma \in \mathbb{Z}$  for grid vector  $\vec{e}$  and shift amount  $\gamma$ . Note that this definition is identical to that for pentagrids; the only change is that  $\vec{x}$  and  $\vec{e}$  are now elements of  $\mathbb{R}^n$  instead of  $\mathbb{R}^2$ . A multigrid consists of  $m$  such grids, each of which is associated with a grid vector  $\vec{e}_j$  and a shift amount  $\gamma_j$ .

As with pentagrids, open  $n$ -dimensional faces in the multigrid correspond to vertices in the dual tiling. The face  $E(\vec{k})$  corresponding to an  $m$ -tuple of integers  $\vec{k}$  is the set of points  $\vec{x}$  that satisfy

$$k_j - 1 < \vec{e}_j \cdot \vec{x} + \gamma_j < k_j$$

for all  $0 \leq j < m$ . If  $E(\vec{k}) \neq \emptyset$ , then the vertex corresponding to  $\vec{k}$  (which we construct in a way similar to that used for pentagrids) appears in the dual tiling.

The dual tiling of a multigrid consists of *zonotopes*. A zonotope is a polytope (a generalization of polygons to multiple dimensions) with point symmetry whose faces also have point symmetry [Sen95]. A three-dimensional zonotope is a *zonohedron*. For multigrids with  $n = 3$  and  $m = 6$ , Socolar and Steinhardt have developed a class of three-dimensional tilings using four types of zonohedra: the rhombohedron, the rhombic dodecahedron, the rhombic icosahedron and the rhombic triacontahedron [SS86].

We shall henceforth ignore these generalizations. We proceed considering only simple Penrose tilings, that is, tilings dual to pentagrids whose shift vectors satisfy the sum condition.

# Chapter 5

## An Algorithm for the Empire Problem

Pentagrids give us an algebraic approach to Penrose tilings. In this chapter, we harness that algebraic power to develop an algorithm for determining the empire of an initial patch.

### 5.1 Valid Sets

Recall from Chapter 4 that for each Penrose tiling by rhombs, there exists a pentagrid that satisfies the sum condition (equation 4.2) such that the tiling is dual to the pentagrid. Let the dual tiling for a pentagrid with shift vector  $\vec{\gamma}$  be  $dual(\vec{\gamma})$ . For convenience, we will call the set of shift vectors satisfying the sum condition  $\mathbb{P}$ :

$$\mathbb{P} = \left\{ \vec{\gamma} \mid \sum_{j=0}^4 \gamma_j = 0 \right\}$$

If  $\vec{\gamma} \in \mathbb{P}$ , then  $dual(\vec{\gamma})$  is a Penrose tiling by rhombs.

We identify a tile with two grid numbers,  $r$  and  $s$ , and 5 line indices,  $k_0 \dots k_4$ . We will say that a tile  $T = (r, s, \vec{k})$ , where  $\vec{k}$  is a five-dimensional vector whose components are the five line indices.

A shift vector  $\vec{\gamma}$  is defined to be *valid* for a tile  $T$  if  $T$  appears in  $dual(\vec{\gamma})$ . We define the *valid set* for a tile  $T$  to be the set

$$V(T) = \{ \vec{\gamma} \in \mathbb{P} \mid T \text{ appears in } dual(\vec{\gamma}) \}$$

We defer discussion of how to identify the valid set for each tile until Section 5.2. For now, we shall assume that we can calculate  $V(T)$  for any tile  $T$ . Now consider a patch of tiles  $P = \{T_1 \dots T_n\}$ . The valid set for that patch consists of the shift vectors that are valid for each tile in the patch:

$$V(P) = V(T_1) \cap \dots \cap V(T_n)$$

Note that this definition does not assume that the tiles of the patch are contiguous.

In Chapter 2, we said that a tile  $T$  is forced by a patch  $P$  if  $T$  appears in all tilings in which  $P$  appears. In our new terminology, a tile  $T$  is forced by a patch  $P$  if, when a shift vector  $\vec{\gamma}$  is valid for  $P$ , it is also valid for  $T$ . The valid sets of  $T$  and  $P$  must then satisfy

$$V(P) \subseteq V(T)$$

Note that if  $P$  is an illegal patch, then  $P$  trivially forces all possible tiles because  $V(P)$  is empty. We will assume that  $P$  is legal and hence that  $V(P)$  is nonempty.

In the following sections, we develop an algorithm for identifying forced tiles that works by testing inclusion of valid sets.

## 5.2 Constraints for Valid Sets

A tile  $T = (r, s, \vec{k})$  corresponds to the intersection of lines  $k_r$  and  $k_s$  in grids  $r$  and  $s$ , respectively. Theorem 5.1 gives necessary and sufficient conditions on the shift vector  $\vec{\gamma}$  such that  $\vec{\gamma}$  is valid for  $T$ . In the theorem,  $\vec{\epsilon}_j$  is defined as in equation 4.1:

$$\vec{\epsilon}_j = \left( \cos\left(\frac{2\pi j}{5}\right), \sin\left(\frac{2\pi j}{5}\right) \right)$$

**Theorem 5.1.** *Let  $\vec{\gamma} \in \mathbb{R}^5$ .  $\vec{\gamma}$  is valid for a tile  $T = (r, s, \vec{k})$  if and only if*

$$k_t - 1 < \vec{x} \cdot \vec{\epsilon}_t + \gamma_t < k_t \tag{5.1}$$

for  $t \neq r, s$ , where  $\vec{x}$  is the simultaneous solution of

$$\vec{x} \cdot \vec{\epsilon}_r + \gamma_r = k_r \tag{5.2}$$

$$\vec{x} \cdot \vec{\epsilon}_s + \gamma_s = k_s \tag{5.3}$$

*Proof.* Suppose that  $T$  appears in the tiling. Recall that we defined  $k_t$  for  $t \neq r, s$  to be

$$k_t = \lceil \vec{x} \cdot \epsilon_t + \gamma_t \rceil$$

We assume that either  $\vec{\gamma}$  is regular or that we have perturbed  $\vec{\gamma}$  such that  $\vec{x} \cdot \epsilon_t + \gamma_t \notin \mathbb{Z}$ . Then we have that

$$k_t - 1 < \vec{x} \cdot \vec{\epsilon}_t + \gamma_t < k_t$$

Now suppose that  $\vec{x}$  satisfies 5.1. Then  $\lceil \vec{x} \cdot \vec{\epsilon}_t + \gamma_t \rceil = k_t$  for  $t \neq r, s$ , so  $T = (r, s, \vec{k})$  appears in the tiling.  $\square$

Note that each tile requires three constraints of the form 5.1, one for each of the three possible values of  $t$ . For instance, if  $r = 0$  and  $s = 1$ , we must apply constraints for  $t = 2, t = 3$ , and  $t = 4$ .

By solving equations 5.2 and 5.3 for the components of  $\vec{x}$ , we may rewrite 5.1 in terms of  $\vec{\gamma}$  and  $\vec{k}$ . The exact form of the solution depends on the values of  $r, s$ , and  $t$ . If  $r - t \equiv t - s \pmod{5}$ , the constraints take the form

$$f_{rst}(\vec{k}) - 1 < f_{rst}(\vec{\gamma}) < f_{rst}(\vec{k}) \quad (5.4)$$

where, for a vector  $\vec{v} = (v_0, v_1, v_2, v_3, v_4)$ ,

$$f_{rst}(\vec{v}) = \begin{cases} \frac{1}{\tau}(v_r + v_s) + v_t & r - s \equiv \pm 1 \pmod{5} \\ -\tau(v_r + v_s) + v_t & r - s \equiv \pm 2 \pmod{5} \end{cases} \quad (5.5)$$

In this case, we say that  $t$  is the *symmetry grid* for  $r$  and  $s$ . In a pentagrid, the angle between grids  $r$  and  $t$  will be equal to that between  $s$  and  $t$ .

If  $r - t \not\equiv t - s \pmod{5}$ , then either  $r$  will be the symmetry grid for  $s$  and  $t$  or  $s$  will be the symmetry grid for  $r$  and  $t$ . (The skeptical reader may confirm this by hand.) We assume that  $r$  is the symmetry grid; if  $s$  is the symmetry grid, simply swap  $r$  and  $s$ . The constraints take the same form as in 5.4, with

$$f_{rst}(\vec{v}) = \begin{cases} \tau v_r + (v_s + v_t) & s - t \equiv \pm 1 \pmod{5} \\ -\frac{1}{\tau} v_r + (v_s + v_t) & s - t \equiv \pm 2 \pmod{5} \end{cases} \quad (5.6)$$

The functions  $f_{rst}$  are a notational convenience; it is useful to characterize the constraints for a particular tile in terms of just  $\vec{\gamma}$  and  $\vec{k}$ . In the next section, we will use each  $f_{rst}$  as the *objective function* in a linear program.

Equations 5.4, 5.5 and 5.6 give necessary and sufficient constraints on  $\vec{\gamma}$  such that  $\vec{\gamma}$  is valid for a tile  $T = (r, s, \vec{k})$ . We may now give a more formal definition of the valid set for  $T$ :

$$V(T) = \{\vec{\gamma} \in \mathbb{P} \mid f_{rst}(\vec{k}) - 1 < f_{rst}(\vec{\gamma}) < f_{rst}(\vec{k}), t \neq r, s\}$$

The valid set for a patch is the intersection of the valid sets for each tile in that patch.

The constraints on  $\vec{\gamma}$  take the form of simple linear inequalities. The inequalities in equation 5.4 define a “slice” of  $\mathbb{R}^5$ ; the valid set for  $n$  tiles is the intersection of  $3n$  such slices. This intersection defines a convex polyhedron in  $\mathbb{R}^5$ .

### 5.3 Testing Valid Set Inclusion

Recall from Section 5.1 that in order to determine whether a tile is forced by an initial patch, we must know whether the valid set for the patch is a subset of the valid set for the tile. Theorem 5.3 is the key for deciding valid set inclusion.

Define the *valid set closure* for a tile  $T = (r, s, \vec{k})$  to be the valid set for a tile with the strict inequalities in 5.4 relaxed:

$$V'(T) = \{\vec{\gamma} \mid f_{rst}(\vec{k}) - 1 \leq f_{rst}(\vec{\gamma}) \leq f_{rst}(\vec{k}) \text{ for } t \neq r, s\}$$

Relaxing the inequalities explicitly includes the singular cases in which  $T$  does not appear in all perturbations of  $\vec{\gamma}$ . We continue to define  $V'(P)$  for a patch  $P = \{T_1 \dots T_n\}$  to be the intersection of the valid sets of  $T_1 \dots T_n$ . Relaxing the inequalities allows us to identify the boundaries of the valid sets, which in turn will allow us to determine if one valid set includes another.

Consider the linear functions given in equations 5.5 and 5.6. Suppose we could identify the maximum and minimum values of these functions within the valid set closure for a patch  $P$ . Define  $max_{rst}(P)$  and  $min_{rst}(P)$  to be the maximum and minimum values of  $f_{rst}(\vec{\gamma})$  for all  $\vec{\gamma} \in V'(P)$ , respectively. Theorem 5.3 explains how these values may be used to determine if  $V(P) \subseteq V(T)$  for any tile  $T$ .

First, we must prove that valid sets and valid set closures are equivalent for our purposes. The proof of the necessary result, Theorem 5.2, is straightforward but makes use of mathematical machinery beyond the scope of this thesis. Hence we defer the proof until Appendix B.

**Theorem 5.2.** Let  $T$  be a tile and  $P$  be a patch such that  $V(P)$  and  $V(T)$  are nonempty.  $V(P) \subseteq V(T)$  if and only if  $V'(P) \subseteq V'(T)$ .

**Theorem 5.3.** A tile  $T = (r, s, \vec{k})$  is forced by a patch  $P$  if and only if

$$\max_{rst}(P) \leq f_{rst}(\vec{k}) \quad \text{and} \quad \min_{rst}(P) \geq f_{rst}(\vec{k}) - 1 \quad (5.7)$$

for  $t \neq r, s$ .

*Proof.* Suppose that  $T$  is forced. Then  $V(P) \subseteq V(T)$  and, by Theorem 5.2,  $V'(P) \subseteq V'(T)$ . Let  $\vec{\gamma} \in V'(P)$  such that  $f_{rst}(\vec{\gamma}) = \max_{rst}(P)$  ( $f_{rst}(\vec{\gamma}) = \min_{rst}(P)$ ).  $\vec{\gamma} \in V'(T)$ , so  $f_{rst}(\vec{\gamma}) \leq f_{rst}(\vec{k})$  ( $f_{rst}(\vec{\gamma}) \geq f_{rst}(\vec{k}) - 1$ ). Therefore  $\max_{rst}(P) \leq f_{rst}(\vec{k})$  ( $\min_{rst}(P) \geq f_{rst}(\vec{k}) - 1$ ).

Now suppose that  $\max_{rst}(P) \leq f_{rst}(\vec{k})$  and  $\min_{rst}(P) \geq f_{rst}(\vec{k}) - 1$  for  $t \neq r, s$ . Let  $\vec{\gamma} \in V'(P)$ . Then  $\min_{rst}(P) \leq f_{rst}(\vec{\gamma}) \leq \max_{rst}(P)$ , so  $f_{rst}(\vec{k}) - 1 \leq f_{rst}(\vec{\gamma}) \leq f_{rst}(\vec{k})$ . Also,  $\vec{\gamma} \in V'(P)$ , so  $\vec{\gamma} \in \mathbb{P}$ . Hence  $\vec{\gamma} \in V'(T)$ , so  $V'(P) \subseteq V'(T)$ , and by Theorem 5.2  $V(P) \subseteq V(T)$ .  $\square$

By Theorem 5.3, it is simple to decide whether a tile is forced by a patch  $P$  if we can calculate  $\max_{rst}(P)$  and  $\min_{rst}(P)$  for each possible combination of  $r, s$  and  $t$ . We can accomplish this task by rewriting the constraints on  $\vec{\gamma}$  given in equations 5.1, 5.5 and 5.6 as a *linear program*. A linear program consists of a set of linear constraints and an *objective function*, a linear function that we wish to maximize or minimize subject to those constraints. In this case, the objective function will be one of the functions  $f_{rst}$ . Efficient algorithms exist to solve linear programs, the most well-known of which is the Simplex algorithm. [CLRS01]

Let us consider an example. Suppose our initial patch  $P$  contains just one tile,  $T = (1, 4, \vec{k})$ . The constraints on  $\vec{\gamma}$  are

$$\begin{aligned} -\tau(k_1 + k_4) + k_0 - 1 &\leq -\tau(\gamma_1 + \gamma_4) + \gamma_0 \leq -\tau(k_1 + k_4) + k_0 \\ \tau k_4 + (k_1 + k_2) - 1 &\leq \tau\gamma_4 + (\gamma_1 + \gamma_2) \leq \tau k_4 + (k_1 + k_2) \\ \tau k_1 + (k_3 + k_4) &\leq \tau\gamma_1 + (\gamma_3 + \gamma_4) \leq \tau k_1 + (k_3 + k_4) \end{aligned} \quad (5.8)$$

Take the case when  $\vec{k} = (1, 0, 0, 0, 0)$ . 5.8 becomes:

$$\begin{aligned} 0 &\leq -\tau(\gamma_1 + \gamma_4) + \gamma_0 \leq 1 \\ -1 &\leq \tau\gamma_4 + (\gamma_1 + \gamma_2) \leq 0 \\ -1 &\leq \tau\gamma_1 + (\gamma_3 + \gamma_4) \leq 0 \end{aligned}$$

Adding more tiles to the patch adds more constraints to the program. We must also include the sum condition as an *equality constraint*:

$$\gamma_0 + \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 = 0;$$

We may now use a linear solver to find  $max_{rst}(P)$  and  $min_{rst}(P)$  for any combination of  $r$ ,  $s$  and  $t$ . For example, for  $r = 1$ ,  $s = 4$  and  $t = 0$ , we find that  $max_{140}(P) = 1$  and  $min_{140}(P) = 0$ .

## 5.4 Algorithm for a Single Tile

In Section 5.5, we will present the algorithm for finding the empire of a patch. For the moment, let us concentrate on a simpler question: given an initial patch  $P$  and a tile  $T$ , is  $T$  forced by  $P$ ? Sections 5.1, 5.2 and 5.3 provide all the tools we need to answer this question. Suppose that  $P = \{T_1 \dots T_n\}$  and  $T = (r, s, \vec{k})$ . We require that the valid sets  $V(P)$  and  $V(T)$  be nonempty; that is, there must exist tilings where  $P$  and  $T$  appear.

1. Determine the constraints on  $\vec{\gamma}$  due to all  $T_i$  in  $P$ . Each tile will require three constraints of the form given in 5.4, in addition to the sum condition as an equality constraint.
2. Use a linear solver to find  $min_{rst}(P)$  and  $max_{rst}(P)$  for  $t \neq r, s$ .
3. Use Theorem 5.3 to determine if  $V(P) \subseteq V(T)$ . If so,  $T$  is forced by  $P$ ; else it is unforced.

Theorem 5.3 guarantees that the result of this algorithm will be correct.

## 5.5 Algorithm for the Empire

Our algorithm to determine if a single tile is forced may be generalized to an algorithm for finding the empire of an initial patch. Suppose we are given an example tiling that includes that a patch  $P = \{T_1 \dots T_n\}$ . In practice, it is very easy to find such an example by adjusting the pentagrid parameters. Every tile in the empire must also appear in that example tiling — that is, after all, the definition of the empire! Therefore we can find the empire of the

patch by repeatedly running the algorithm for a single tile on each of a finite set of tiles in the example tiling.

Suppose we are given an initial patch  $P = \{T_1 \dots T_n\}$  and a set of tiles  $S = \{T'_1 \dots T'_m\}$  to test for inclusion in the empire. We assume that there exists some  $\vec{\gamma} \in \mathbb{P}$  such that  $\vec{\gamma}$  is valid for  $P$  and for each tile in  $S$ . We can save some time by calculating  $\min_{rst}(P)$  and  $\max_{rst}(P)$  ahead of time.

1. Determine the constraints on  $\vec{\gamma}$  due to all  $T_i$  in  $P$ . Each tile will require three constraints of the form given in 5.4, in addition to the sum condition as an equality constraint.
2. Use a linear solver to find  $\min_{rst}(P)$  and  $\max_{rst}(P)$  for each possible combination of  $r$ ,  $s$  and  $t$ .
3. For each tile  $T'_i$  in  $S$ , use Theorem 5.3 to determine if  $V(P) \subseteq V(T'_i)$ . If so, add  $T'_i$  to the empire.

The results will necessarily be a finite subset of the empire. Nonetheless, this algorithm is the first to determine even a finite empire with complete certainty. Minnick's algorithm, for instance, found subsets of the finite empire but could not guarantee that *all* tiles in the finite empire were identified [Min98]. The brute force algorithm in Section 3.3 suffered from the same weakness. But Theorem 5.3 is strong enough to detect both when a tile is part of the empire and when it is not.

Note that the algorithm does not differentiate between local and remote empires. In fact, running this algorithm on just the initial patch will identify the local empire and the remote empire. Moreover, the patch  $P$  need not be one of the eight vertex configurations; it need not even be contiguous. The algorithm will identify the empire of any legal combination of tiles.

Appendix A presents the results of running this algorithm on the eight vertex configurations.

# Chapter 6

## Conclusions and Future Work

The main contribution of this thesis is the algorithm presented in Chapter 5. We have reduced the empire problem in Penrose tilings to linear optimization, a well-defined problem for which efficient algorithms exist. Our algorithm is the first to determine precisely which tiles are and are not included in the empire of an initial patch. The results in Appendix A are striking for their beauty and, in some cases, for the sheer number of tiles that are forced by a just a handful of initial tiles.

Perhaps the most exciting aspect of this algorithm is its potential for generalization. We expect that it will be straightforward to rewrite the language of Chapter 5 for generalized Penrose tilings (Section 4.4.1) and tilings generated by any multigrid (Section 4.4.2). We also hope to translate our results back to tilings by kites and darts, although those prototiles do not exhibit the useful algebraic properties of the thick and thin rhombs.

There are still many directions for future research. We have discovered finite portions of the empires, but we lack a means of characterizing an empire in its entirety. The patterns of the forced tiles are intriguing and well worth further study. In particular, the S exhibits fractal-like patterns in its finite empire (Figure A.5), but we would like to show that these patterns continue *ad infinitum*. Do similar patterns appear in other empires (assuming that they do appear in the empire of the S)? A second avenue of research might be called the “reverse empire problem”: given a set of a tiles  $S$  that appears in some complete tiling, which patches force all of the tiles in  $S \cup P$ ? For instance, suppose that  $S = \emptyset$ ; then the set of patches  $P$  is precisely the set of patches that have no empire besides the initial patch. If we add the further constraint that  $P$  be contiguous, we have the problem of the *princess patch*:

find the largest contiguous patch that does not force any tiles.

Despite the numerous questions that we raise in the preceding paragraphs, we have made a significant advance in the study of the empire problem. The elegant correspondence between pentagrids and Penrose tilings by rhombs lets us capture the empire problem in mathematical terms and solve it using well-known techniques in linear optimization. This powerful approach allows us to find the complete empires of Penrose tilings for the first time.

# Appendix A

## Empires of the Eight Vertex Configurations

We have implemented our algorithm using a linear solver, `lp_solve`<sup>1</sup>. The following pages give the results of running the algorithm on the eight vertex configurations in Figure 3.1. In each figure, the tiles in the initial patch are yellow. Unforced tiles are not shown.

---

<sup>1</sup>[http://groups.yahoo.com/group/lp\\_solve/](http://groups.yahoo.com/group/lp_solve/)



Figure A.1: The empire of the D configuration.

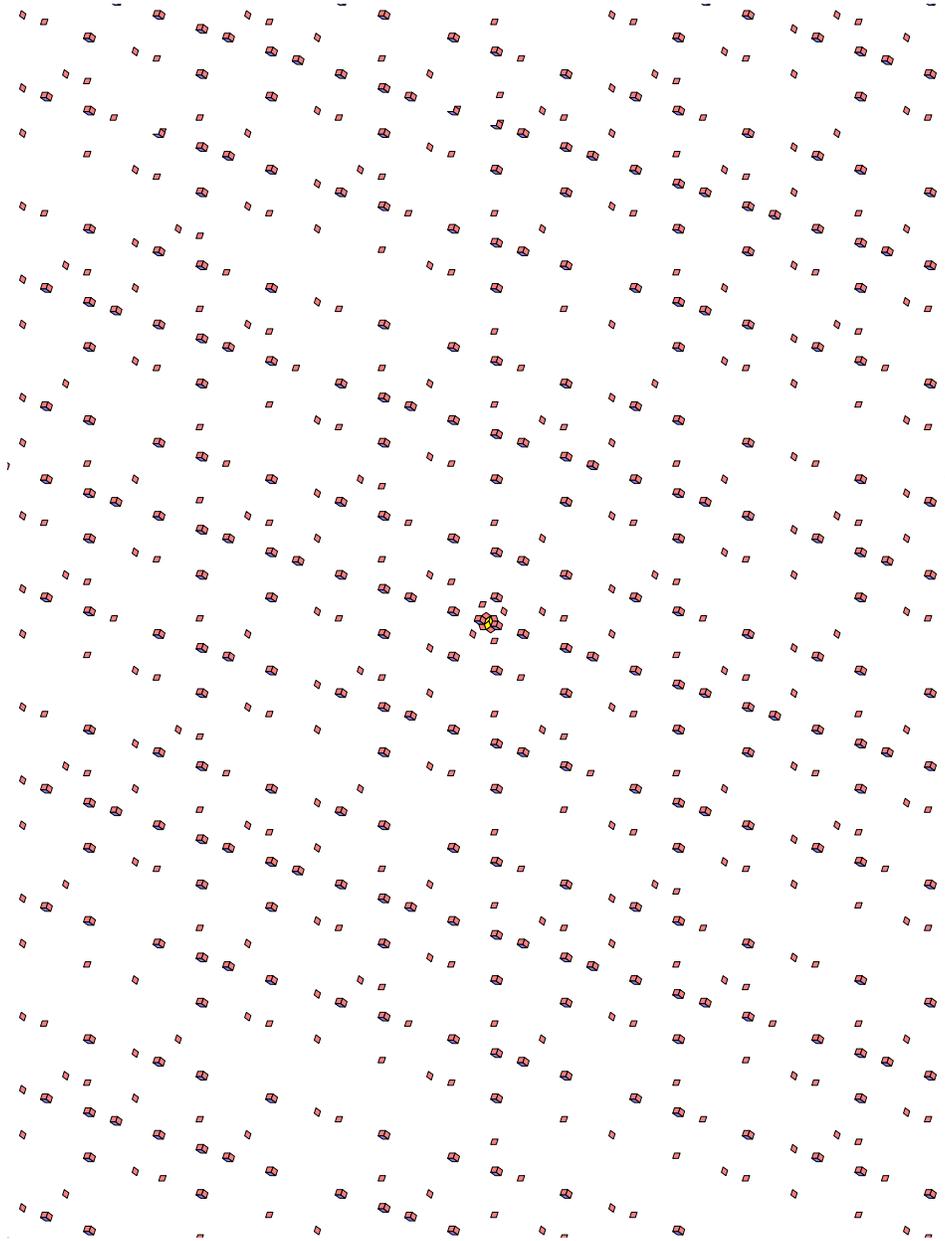


Figure A.2: The empire of the Q configuration.

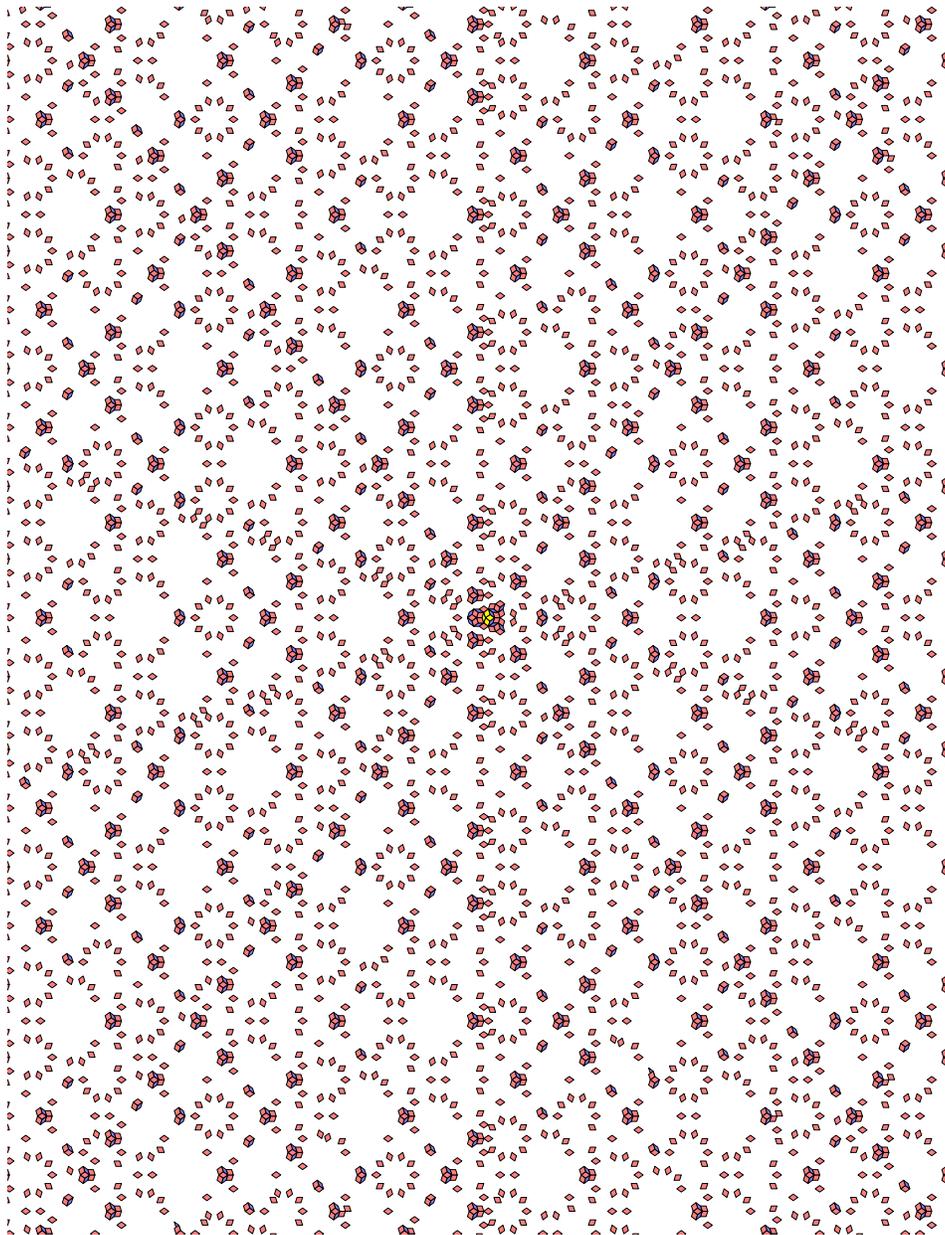


Figure A.3: The empire of the K configuration.

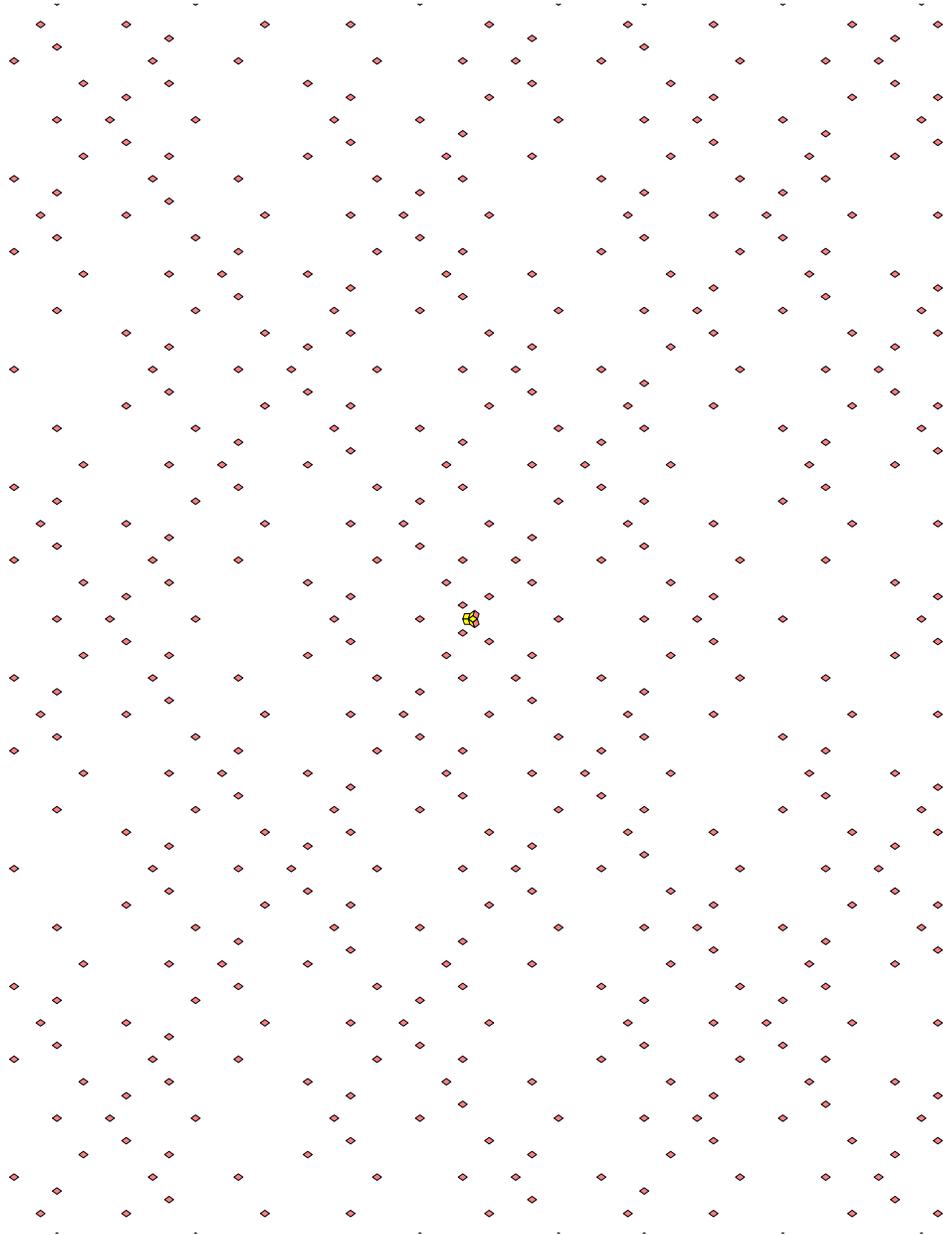


Figure A.4: The empire of the J configuration.

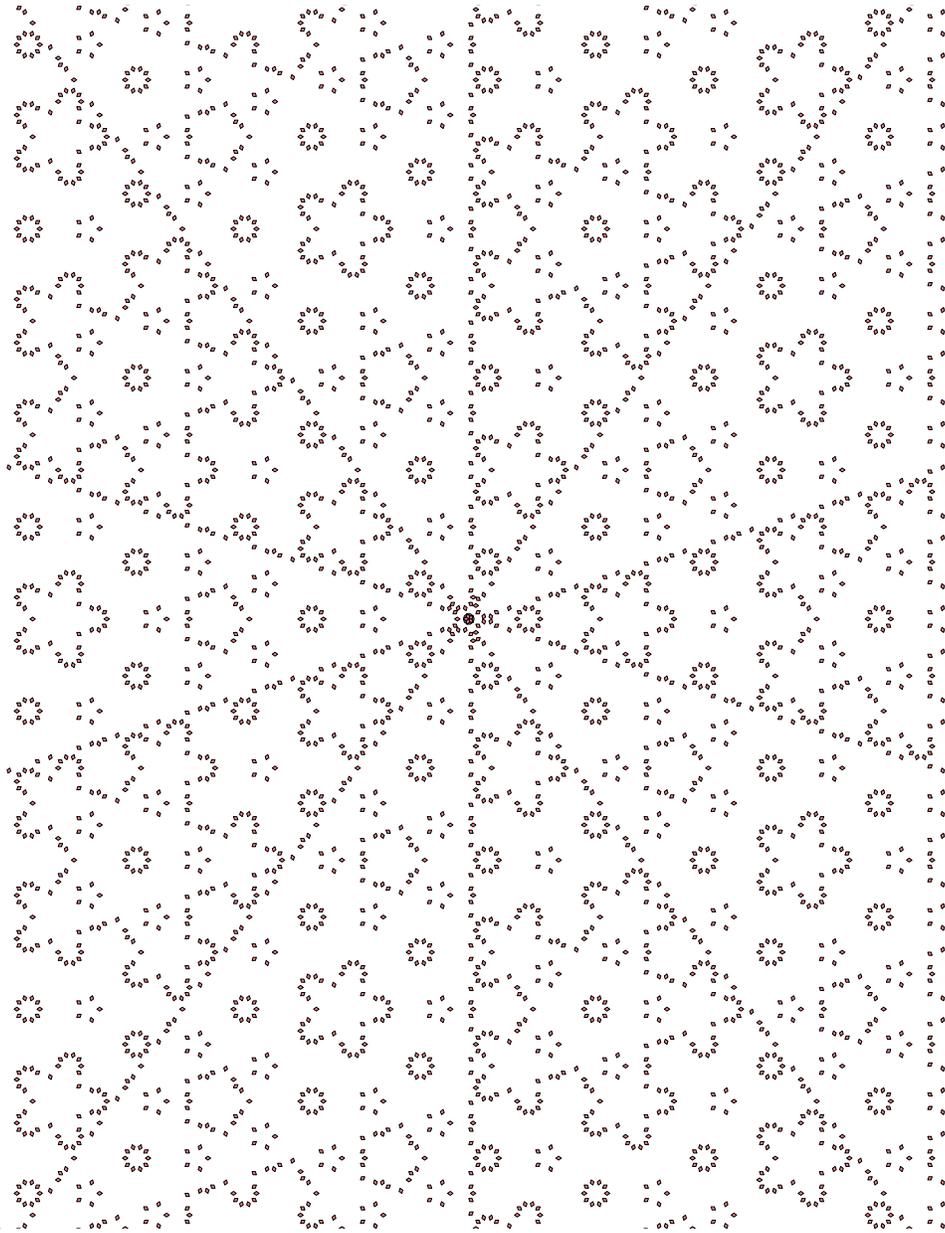


Figure A.5: The empire of the S configuration.

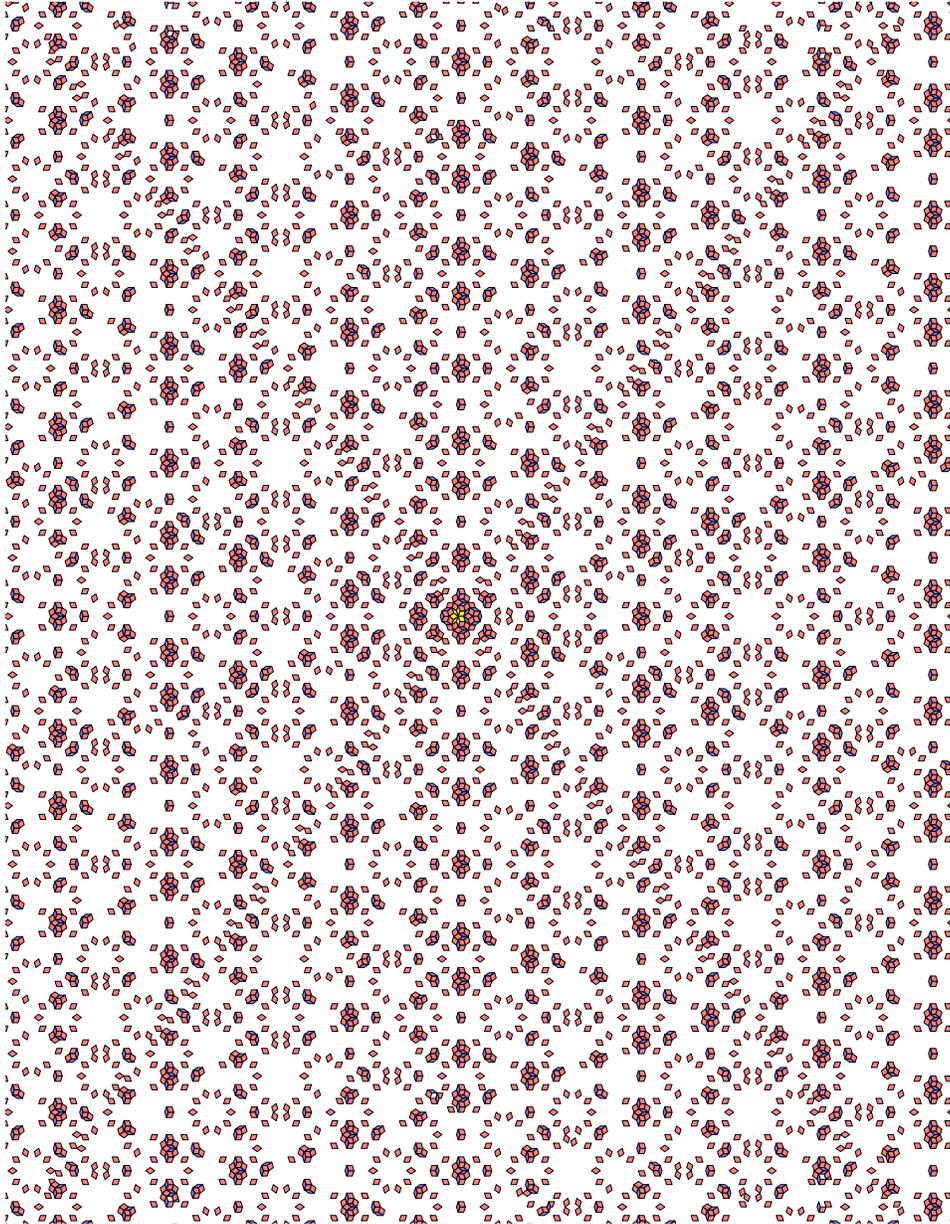


Figure A.6: The empire of the S3 configuration.

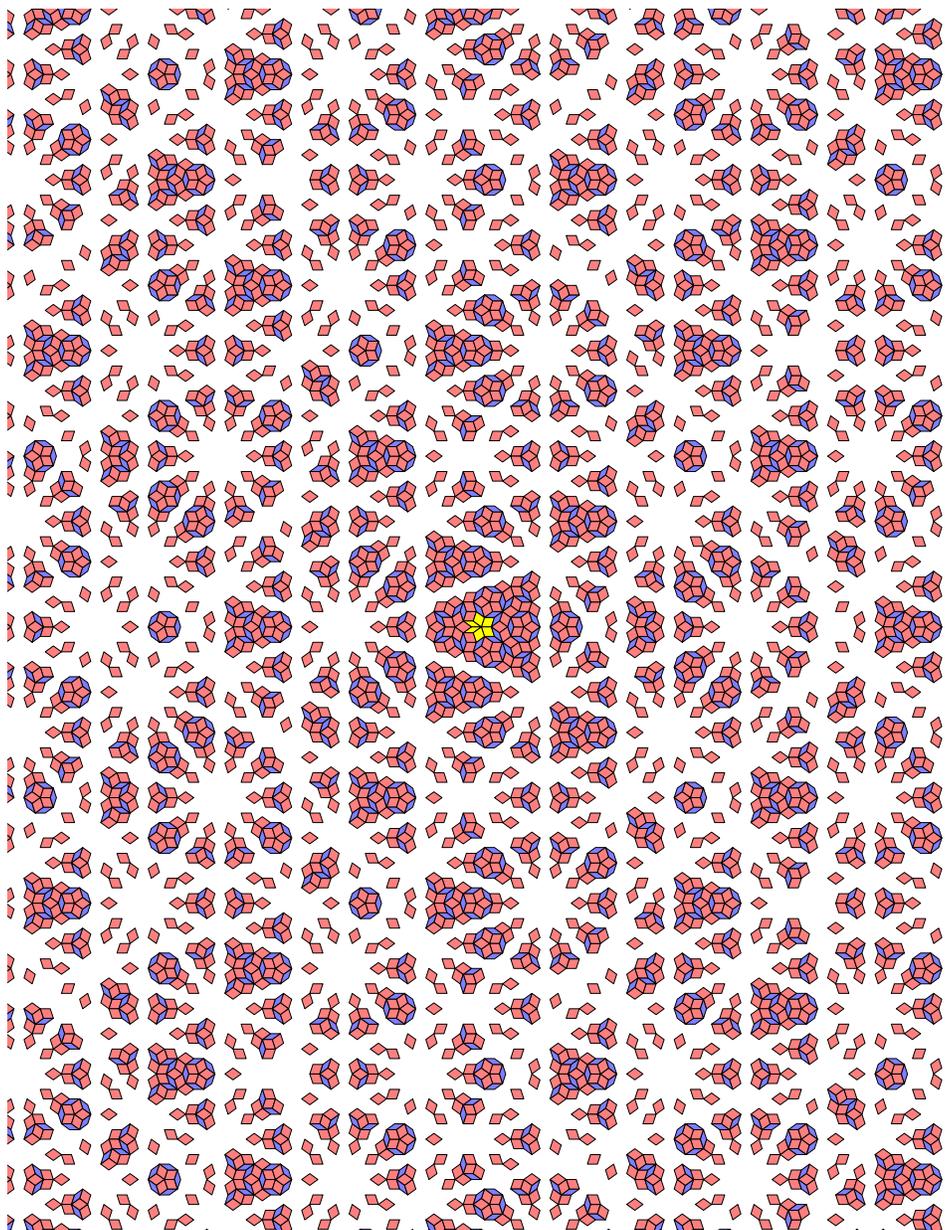


Figure A.7: The empire of the S4 configuration.

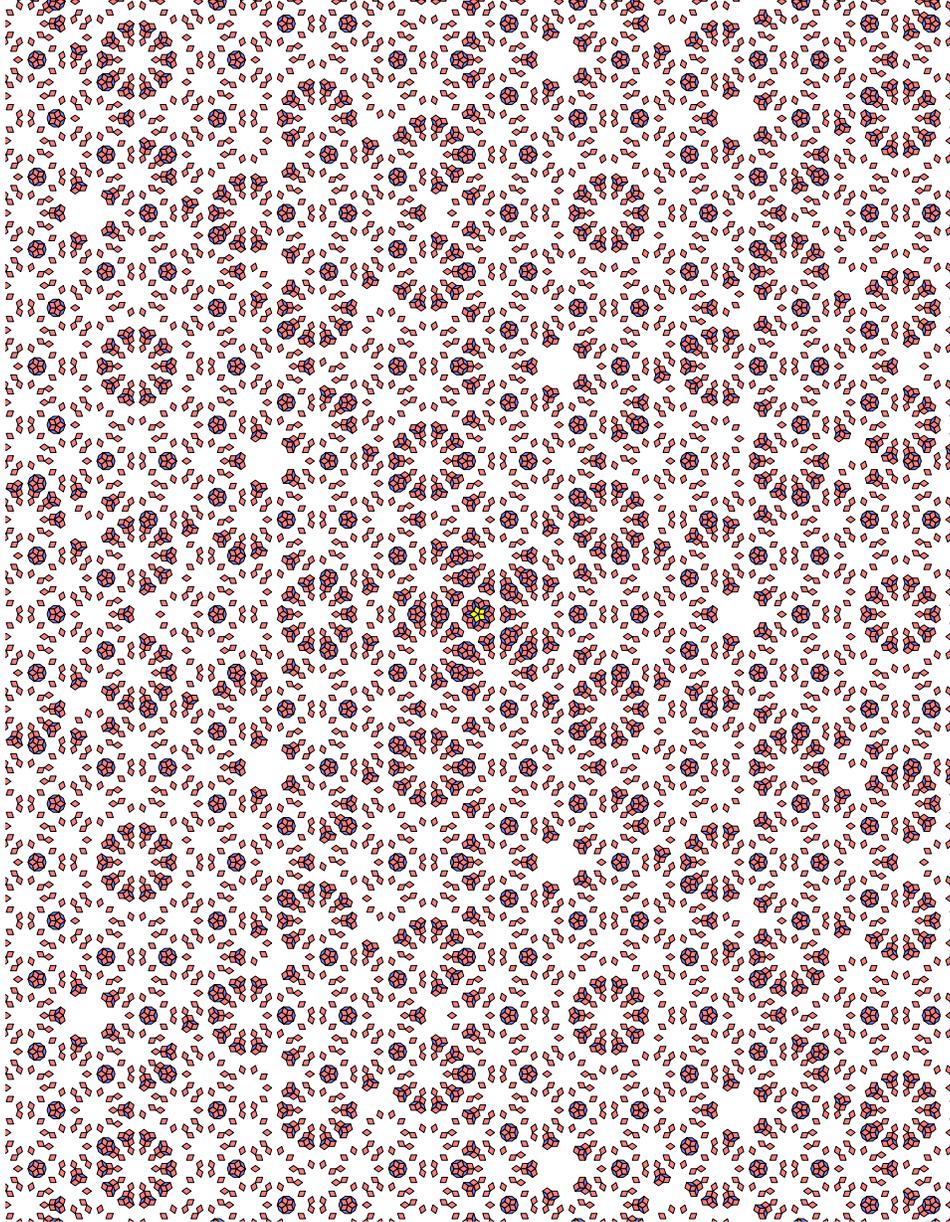


Figure A.8: The empire of the S5 configuration.

# Appendix B

## Proof of Theorem 5.2

The proof of Theorem 5.2 makes use of basic concepts in topology and convexity; more information can be found in any introductory textbook, such as [Lay82]. We will use the following notation:

- $cl A$  is the topological closure of a set  $A$ .
- $int A$  is the interior of a set  $A$ .

We define an *open half-space* to be the area on one side of a plane cutting through  $\mathbb{R}^n$ :

$$\vec{a} \cdot \vec{x} > b \tag{B.1}$$

where  $\vec{a} = \{a_1 \dots a_n\}$  is normal to the plane and  $b$  is constant. The closure of an open half-space is the corresponding *closed half-space*, defined by equation B.1 with the strict inequality relaxed (that is,  $>$  replaced by  $\geq$ ). An open half-space is an open convex set; a closed half-space is a closed convex set. A valid set (defined in Section 5.1) is the intersection of a finite number of open half-spaces, while a valid set closure (defined in Section 5.3) is the intersection of a finite number of closed half-spaces.

**Lemma B.1.** *Let  $A$  and  $B$  be open convex sets such that  $A \cap B$  is nonempty. Then  $cl(A \cap B) = cl A \cap cl B$ .*

*Proof.* Let  $S = cl A \cap cl B$ . As  $A$  and  $B$  are both open sets, it follows that  $int S = A \cap B$ :

$$\begin{aligned} int S &= int(cl A \cap cl B) \\ &= int(cl A) \cap int(cl B) \\ &= A \cap B \end{aligned}$$

$S$  is convex because the closure of a convex set is convex, as is the intersection of two convex sets. By [Lay82],  $cl(int S) = cl S$  if  $S$  is convex and  $int S \neq \emptyset$ . Therefore

$$\begin{aligned} cl(A \cap B) &= cl(cl(A) \cap cl(B)) \\ &= cl(A) \cap cl(B) \end{aligned}$$

where the last step follows because the intersection of two closed sets is also closed.  $\square$

**Lemma B.2.** *Let  $P = \{T_1 \dots T_n\}$  be a patch such that  $V(P)$  is nonempty.  $cl(V(P)) = V'(P)$ .*

*Proof.* Follows from induction on Lemma B.1.  $\square$

**Lemma B.3.** *Let  $P$  be a patch such that  $V(P)$  is nonempty.  $int(V'(P)) = V(P)$ .*

*Proof.*  $V'(P) = cl(V(P))$  by Lemma B.2. Hence  $int(V'(P)) = int(cl(V(P))) = V(P)$  as  $V(P)$  is an open set.  $\square$

**Theorem B.1.** *Let  $T$  be a tile and  $P$  be a patch such that  $V(P)$  and  $V(T)$  are nonempty.  $V(P) \subseteq V(T)$  if and only if  $V'(P) \subseteq V'(T)$ .*

*Proof.* Suppose  $V(P) \subseteq V(T)$ . By Lemma B.2,  $cl(V(P)) = V'(P)$  and  $cl(V(T)) = V'(T)$ . Hence  $V'(P) \subseteq V'(T)$ .

Now suppose  $V'(P) \subseteq V'(T)$ . By Lemma B.3,  $int(V'(P)) = V(P)$  and  $int(V'(T)) = V(T)$ . Hence  $V(P) \subseteq V(T)$ .  $\square$

# Bibliography

- [CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*, chapter 29. MIT Press, London, 2001.
- [dB81] N. G. de Bruijn. Algebraic theory of Penrose’s non-periodic tilings. *Nederl. Akad. Wetensch. Proc.*, 84:39–66, 1981.
- [dB86] N. G. de Bruijn. Quasicrystals and their Fourier transform. *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen, Mathematical Sciences*, A89:123–152, 1986.
- [Gar77] Martin Gardner. Extraordinary nonperiodic tiling that enriches the theory of tiles. *Scientific American*, pages 110–121, January 1977.
- [Gar95] Martin Gardner. *From Penrose Tiles to Trapdoor Ciphers: Essays on Recreational Mathematics*, chapter 1-2, pages 1–29. W. H. Freeman, New York, 1995.
- [GS87] Branko Grünbaum and G. C. Shephard. *Tilings and Patterns*. W. H. Freeman and Company, New York, 1987.
- [Lay82] Steven R. Lay. *Convex Sets and Their Applications*. John Wiley & Sons, 1982.
- [Min98] Linden Minnick. Generalized forcing in aperiodic tilings. Technical report, Williams College Department of Computer Science, 1998.
- [Sen95] Marjorie Senechal. *Quasicrystals and Geometry*. Cambridge University Press, Cambridge, 1995.
- [SS86] Joshua E. S. Socolar and Paul J. Steinhardt. Quasicrystals. II. unit-cell configurations. *Physical Review B*, 34(2):617–647, July 1986.