Generalized Forcing in Aperiodic Tilings

by
Linden Minnick

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 23, 1998

Abstract

An aperiodic tiling is one which uses a finite set of prototiles to tile space such that there is no translational symmetry within any one tiling. These tilings are of interest to researchers in the natural sciences because they can be used to model and understand quasicrystals, a recently-discovered type of matter which bridges the gap between glass, which has no regular structure, and crystals, which demonstrate translational symmetry and certain rotational symmetries. The connection between these two areas of research helps to create a practical motivation for the study of aperiodic tilings.

We plan to study both theoretical and applied aspects of several types of aperiodic structures, in one, two, and three dimensions. Penrose tilings are tilings of the plane which use sets of two prototiles (either kites and darts, or thin rhombs and thick rhombs) whose edges are marked in order to indicate allowed arrangements of the tiles. There is a small number of legal vertex configurations, and most of these force the placement of other tiles which may or may not be contiguous with the original ones. It is somewhat intuitive that in most cases the placement of a group of tiles forces the placement of some set of adjacent tiles. It is much more difficult to believe that many groups of tiles also force the placement of infinitely many non-adjacent tiles. This phenomenon has implications which may be important in designing and implementing an efficient data structure to explore and store information about aperiodic tilings.

For the most part, we plan to study forced tiles in two-dimensions, although we believe that this theory may be generalized to higher dimensions. We will use musical sequences and Ammann bars, concepts established by Conway and Ammann, to develop an algorithm for predicting the placement of fixed bars given an initial sequence of parallel bars. This, in turn, may be used to fix the positions of tiles and clusters of tiles within any legal tiling. Our goal is to determine a method for predicting the position of infinitely many forced tiles given an initial cluster of tiles. These descriptions will be partly grammar-based inflations of musical sequences, and partly algebraic solutions, and will themselves be aperiodic.

Acknowledgements

I would like to thank my advisor, Duane Bailey. He knew that I wanted to do a thesis before I did, and I couldn't, and probably wouldn't, have done this without him. The majority of the figures in this thesis were prepared using the **xfig** program, and I am grateful its writers for simplifying my task in creating such a figure-intensive paper. I appreciate input from the friends and faculty members who reviewed this work in its various forms, in particular Bill Lenhart, Andrew Stortz, and Noah Peeters. And last, but certainly not least, I'd like to thank my fellow inhabitants of the thesis lab, Ben, Brendan, and Jon.

Contents

List of Figures

List of Tables

Chapter 1

Background

1.1 Tilings

A *tiling* fills space using some finite number of shapes known as *prototiles*. This paper considers tilings in two dimensions, although we will devote some discussion towards generalizing these results to higher dimensions. A tiling in two dimensions covers the plane, allowing neither holes in the tiling nor overlapping tiles. A *periodic tiling* suggests some *unit cell* which is repeated through translation in some well-defined fashion throughout the tiling. For example, Figure 1.1 is a periodic hexagonal tiling of the plane. We notice that the entire plane may be covered using the hexagon as a unit cell. Furthermore, assuming that the tiling is infinite, if we translate the tiling from the vertex labeled **a** to the vertex labeled **b** along the vector shown in the figure, we will have another copy of the tiling which is exactly the same. This property is known as *translational symmetry*, and is an important property of periodic tilings. Furthermore, we notice that if we fix the vertex labeled **b**, we may rotate the tiling in multiples of 120° and still have the same tiling. This property is known as three-fold *rotational symmetry*. The hexagonal tiling also has six-fold rotational symmetry, as we may see from the fact that the tiling may be rotated around the point labeled **c** in multiples of 60° while leaving the tiling unchanged. A periodic tiling may have two-, three-, four-, six-, or eight-fold rotational symmetry, but not five-fold rotational symmetry. Finally, we notice that in a hexagonal tiling, all of the hexagons are oriented in the same direction.

By contrast, many tilings suggest no unit cell and there is no regular re-

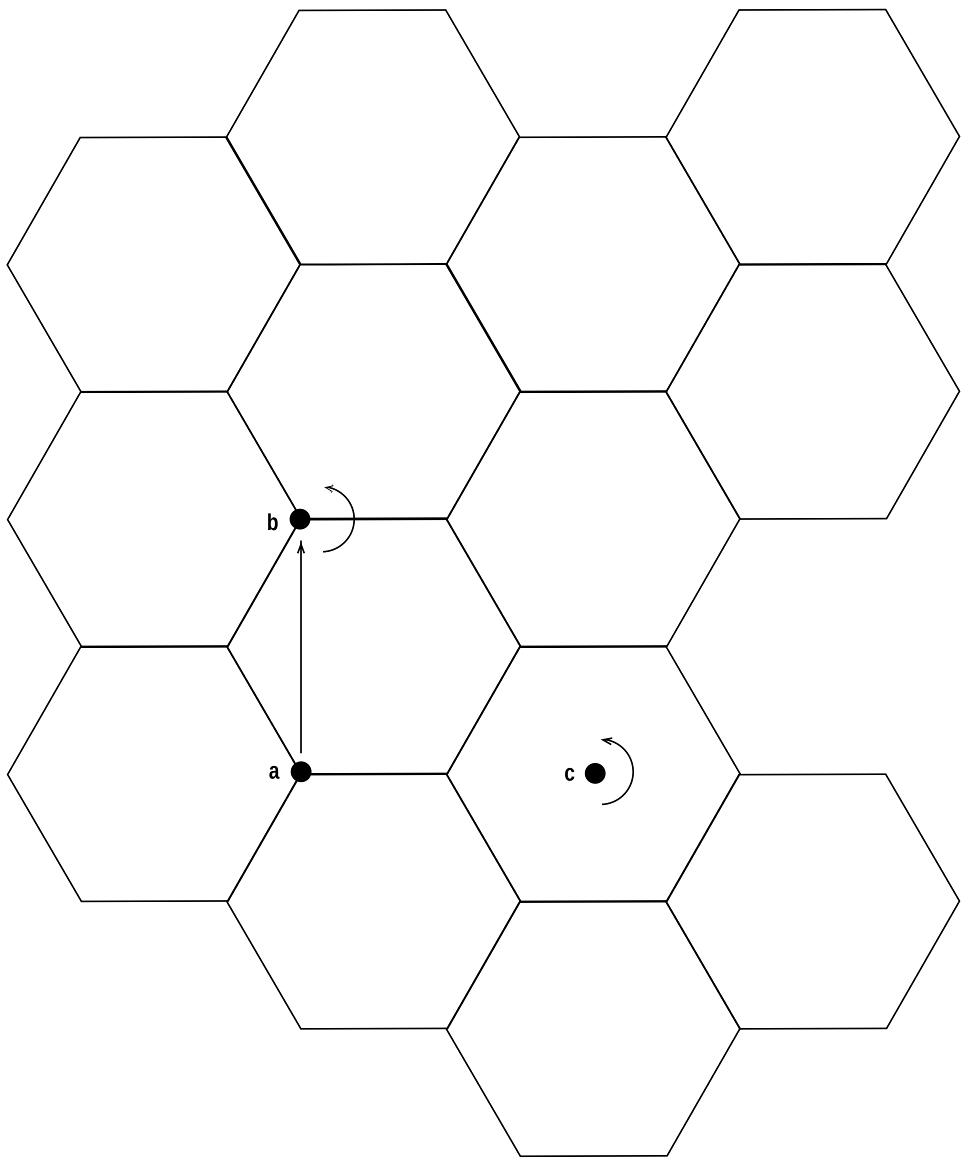


Figure 1.1: A partial hexagonal tiling of the plane.

peating pattern. A tiling is *nonperiodic* if there is no translational symmetry in the tiling. Sets of nonperiodic prototiles may tile the plane periodically or nonperiodically. A set of prototiles, along with matching rules, is said to be *aperiodic* if it is forced to be nonperiodic. In other words, it is not possible to tile the plane periodically using an aperiodic set of tiles. For example, Figure 1.2 is an aperiodic tiling using Penrose kites and darts. An aperiodic tiling may have local five-fold rotational symmetry, which is not possible in a periodic tiling. Aperiodic tilings may also have long-range symmetry of orientation. If we shade all of the decagons in Figure 1.2, we notice that they are all oriented in the same direction. This will be true to some extent throughout an aperiodic tiling. Often, as we will see later, forced tiles or patches of tiles are oriented in the same direction.

1.2 Related Material

The first example of an aperiodic tiling was discovered by Robert Berger in 1964, consisting of over 20,000 prototiles. By 1974, Penrose was able to reduce the number of tile shapes to two. He discovered several sets, including the kite/dart, and thin rhomb/thick rhomb. The two sets are related, and their tilings produce mosaics with the same positional and orientational symmetries. To prove that his matching rules forced nonperiodicity, Penrose showed that the matching rules were in a one-to-one correspondence with a set of deflation rules. In 1984, Shechtman et al discovered an alloy with icosahedral symmetry, an example of a fundamentally new phase of solid matter. It is not possible for icosahedra to fill three-dimensional space periodically. This type of matter was dubbed “quasicrystals” [Ste86], and is now considered a distinct form of matter.

Quasicrystals have a novel kind of order, intermediate between purely crystalline and amorphous. Quasicrystals are analogous to some types of aperiodic tilings in several ways. They are locally five-fold symmetrical, the "unit cells" arranged without strict periodicity, and they exhibit both a kind of long-range translational symmetry and a skewed variation of the lattice planes found in regular crystals. The study of quasicrystals has brought together two existing branches of theory — the theory of metallic glasses and the mathematical theory of aperiodic tilings. A tiling can be a good analogue of a crystal. Just as a crystal fills three-dimensional space with unit cells, so a tiling fills two-dimensional space with tiles. Many properties of three-dimensional crystals

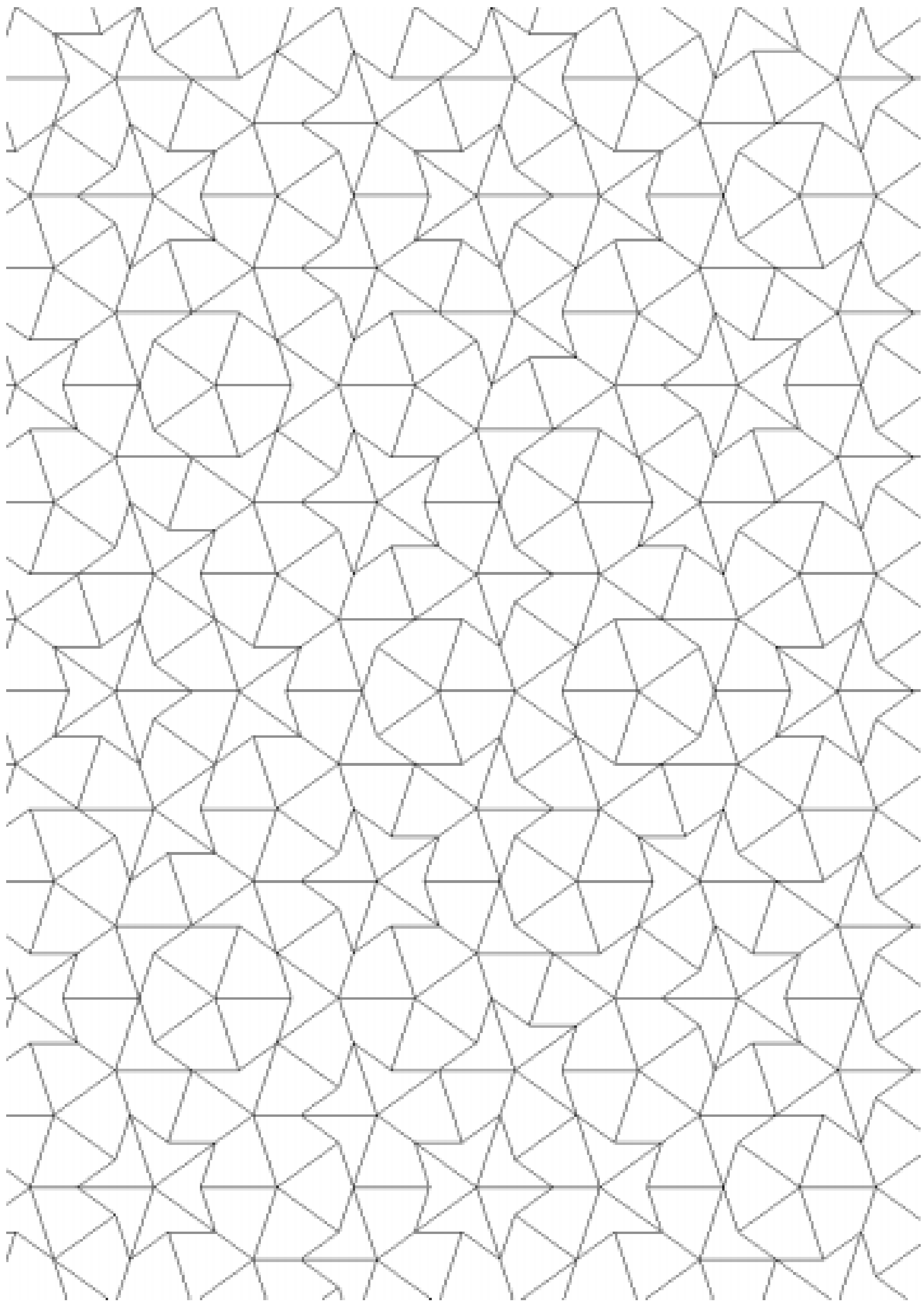


Figure 1.2: An aperiodic tiling of the plane, using Penrose kites and darts.

are also observed in two-dimensional tilings. For example, the two-dimensional hexagonal packing has the lattice planes and long-range orientational symmetry that are found in three-dimensional crystals [Nel86].

In 1991 Socolar described an algorithm for producing defect-free Penrose tilings using thin rhombs and thick rhombs. He classifies surfaces at the edge of a patch of tiles as forced, unforced, or *marginal*. The algorithm adds tiles to forced locations until only unforced and marginal edges remain. The result is a patch of tiles shaped like either a regular trapezoid (a coffin) or a rhombus (a diamond). In either case, the corners of these shapes have angles of $108°$ and $72°$. Thick rhombs are added to one of the $108°$ corners at a marginal site, which results in more forced sites, forcing tiles until another patch with only unforced and marginal edges is created. Socolar proved that this method will always result in a defect-free tiling and, indeed, that all tilings that may be created given a particular initial patch may be created using this method [Soc91].

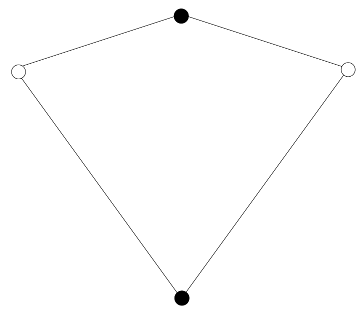
Whether the plane may be tiled aperiodically with only one tile remains an open problem. In 1996, Gummelt discovered a single tile which *covers* the plane aperiodically. Copies of the tile are allowed to overlap, and they are based upon the Penrose tiles. The tile is constructed using the fact that the "cartwheel" patch can be used to cover every Penrose tiling. The intersection of all possible ways in which two cartwheel patches may intersect each other leads to a series of dark and light patches on a decagon. Appropriate overlapping of patches determines the matching rules [Gum96].

Chapter 10 of Grunbaum and Shephard's book Tilings and Patterns explores many aspects of aperiodic tilings [GS87]. It gives a comprehensive introduction to musical sequences — aperiodic sequences which are essential in order to understand forcing patterns in aperiodic tilings. Ammann lines are also introduced as decorations of tilings which can be used as an alternative to matching rules. This thesis considers extensions to many of these basic concepts.

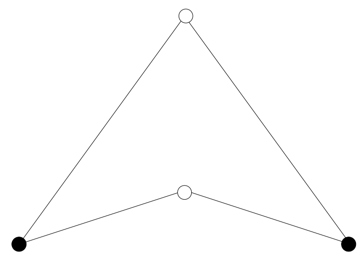
1.3 Penrose Kites and Darts[1]

We will study Penrose kites and darts in great depth. Figure 1.3 illustrates a *kite* and a *dart* marked according to the matching rules which ensure that they will tile the plane aperiodically. The edges of kites and darts have two

[1]We follow, for the most part, the terminologies of Penrose and Conway.



(a) kite



(b) dart

Figure 1.3: A Penrose kite and dart - matching rules are indicated via vertex decorations.

lengths, long and short. The ratio of the lengths of the edges is 1 to τ, where τ is the golden ratio, $\frac{1+\sqrt{5}}{2} \approx 1.618...$[2]. The matching rules for the Penrose kites and darts are quite simple: filled circles must be aligned with other filled circles, open circles must be aligned with other open circles, and edges of the same length must be fitted next to each other.

In any legal tiling of the Penrose kites and dart, there are seven possible vertex configurations, shown in Figure 1.4, that are consistent with the matching rules. Each of these arrangements of tiles will be found infinitely often in any tiling of kites and darts. Given some initial cluster of tiles, it is easy to verify that in many cases there is some group of adjacent tiles which is forced by the initial patch. In any tiling containing the patch, these adjacent tiles are always present. The simplest example involves the dart: a dart requires that two kites fit along its two short edges, forming the vertex configuration known as the *ace*. We can convince ourselves of this necessity either by the matching rules or by the fact that the only legal vertex configuration involving the concave vertex of a dart is the ace. Therefore, whenever we see a dart in a tiling, we will see an ace. A slightly more complicated example involving the queen vertex configuration is given in Figure 1.5. The dark grey tiles are the initial patch corresponding to the queen — the light grey tiles are adjacent tiles forced by the queen. Notice that the rough outline of the tiles forced by a queen is a diamond, or rhombus. In the same figure, we see that the outline of

[2]There are many uses of the golden ratio in art, mathematics, and nature.

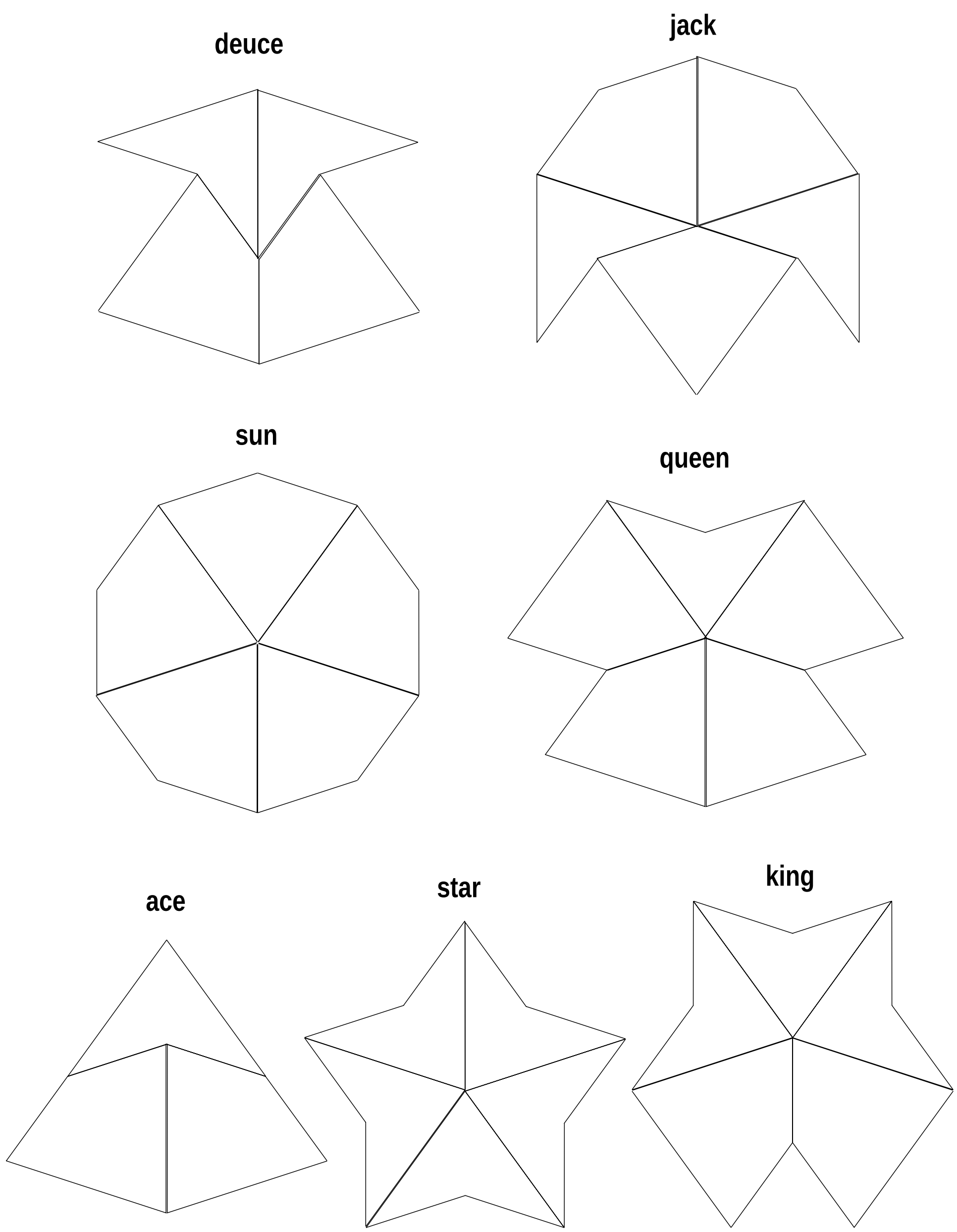


Figure 1.4: The seven legal vertex configurations in any global tiling of Penrose kites and darts. The common names are due to Conway.

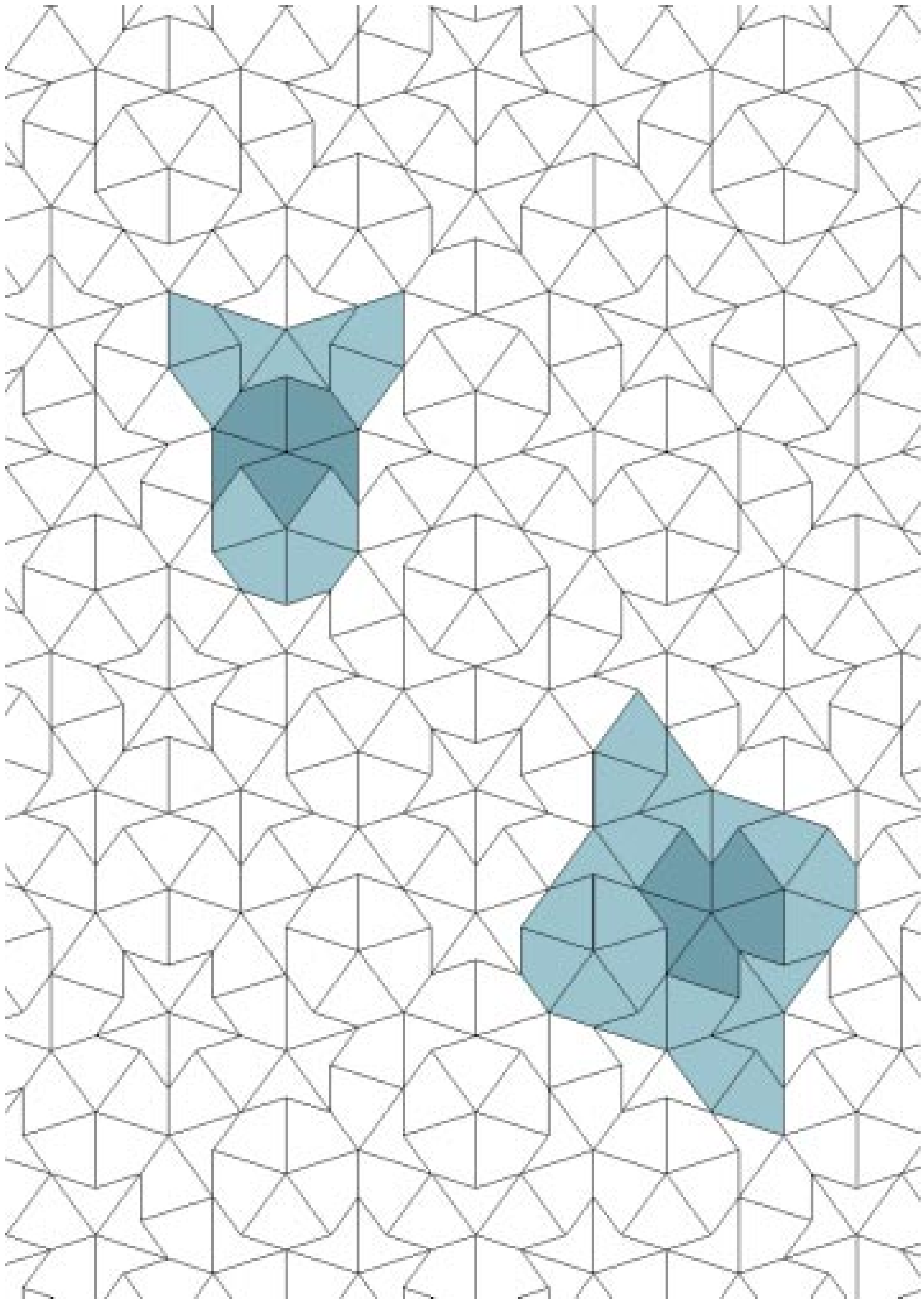
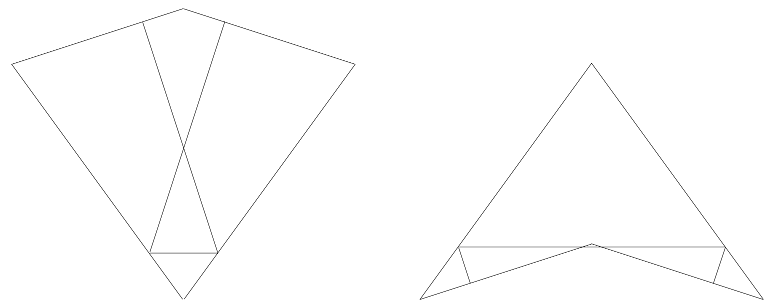


Figure 1.5: The queen and jack (dark) and the adjacent tiles they force.



(a) kite　　(b) dart

Figure 1.6: A Penrose kite and dart decorated with Ammann lines.

the adjacent tiles forced by a jack is a coffin, or regular trapezoid. It turns out that almost every initial patch of tiles forces a patch of adjacent tiles which is a variation of one of these two shapes. This will be important in our analysis of forced tiles in later chapters.

1.4 Ammann Lines — Decorations for Matching

Penrose kites and darts may be decorated with markings known as *Ammann lines*. The Ammann lines for these two prototiles are given in Figure 1.6. Arranging the tiles such that these markings form continuous lines is sufficient to force aperiodicity in the prototiles. In this sense, the lines may actually take the place of the vertex matching rules given in Figure 1.3. In any infinite tiling, the Ammann lines will be extended through the tiling to form a grid of parallel lines in five directions, separated by angles of 72°. The pairs of parallel lines themselves are separated by either long or short intervals, and the sequence of long and short is a quasiperiodic sequence known as a Fibonacci sequence. Finite subsections of these infinite sequences have been named "musical sequences" by Conway. We will study musical sequences at length in the next chapter.

In fact, it may be argued that the Ammann bars are fundamental, and the only function of the tiles is to give practical realization to them. From this position, the correct approach to the discovery of new aperiodic sets will be to find tiles which permit systems of bars with the required properties [GS87].

In order to understand two-dimensional tilings, it is helpful for us to first study some properties of one-dimensional aperiodic sequences. One class of aperiodic sequences which is instructive in this pursuit are the musical sequences. These aperiodic sequences, which are constructed using the irrationality of τ as a basis for their non-periodic nature, may be used to predict and explain many interesting properties of tilings. This thesis expands our understanding of the power of Ammann lines in finding forced tiles in aperiodic tilings.

Chapter 2

Musical Sequences and Forced Bars

2.1 Musical Sequences

In the thirteenth century, mathematician Leonardo Fibonacci conducted a thought experiment involving the proliferation of a population of rabbits. He imagined a scenario in which each month a new baby rabbit would be born for each adult rabbit, and every baby rabbit already in the population would grow into an adult rabbit. The population would undergo the same transformation each month, and no rabbits would ever die. Fibonacci showed that the ratio of adults to babies approached the golden ratio as this experiment was carried out to the limit [St86].

These ideas became the basis for a quasiperiodic sequence, known as a Fibonacci sequence or a *musical sequence*, of long (**L**) and short (**S**) intervals. These sequences may be generated by repeatedly applying grammar-like rewriting rules. In each iteration of substitutions, each **L** in the sequence is replaced by **LS** and each **S** is replaced by **L**. This is directly analogous to Fibonacci's experiment as described above. This information is given as Rule 1 (see Appendix B).

Another method of generating quasiperiodic sequences involves passing a line **I** with irrational slope through a regular two-dimensional lattice. For every integer value of x, the line is projected down to the nearest integer value for y. If the difference between two consecutive y values is 2, it is substituted by an **L**, and if the difference is 1, it is replaced by an **S**. Because the slope of the line is irrational, the sequence is quasiperiodic. If the slope is τ, then the sequence is a musical sequence.

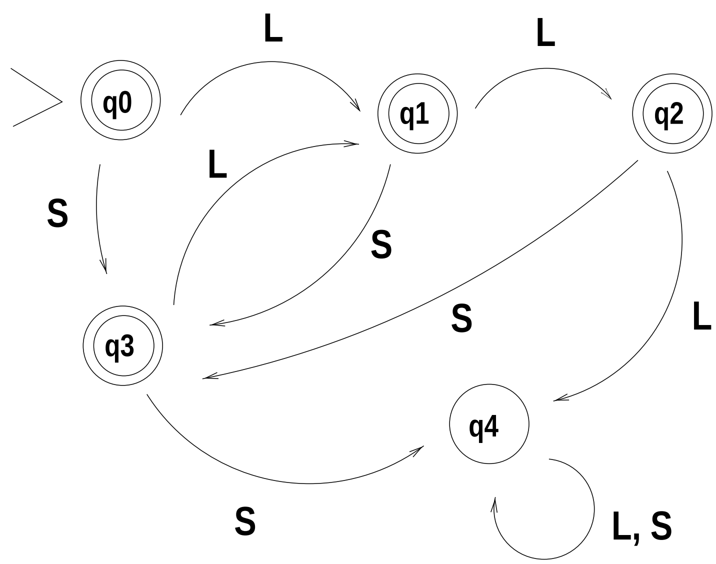
Musical sequences have two basic properties. First, musical sequences may be decomposed into other musical sequences by deleting every instance of a single **L**, replacing every **S** with an **L**, and replacing every **LL** with an **S**. The grammar is given as Rule 2. Second, in any musical sequence, there will be at most two consecutive **L**'s and **S**'s will be isolated between **L**'s. This is given as Rule 3. Decomposition under Rule 2 must produce a sequence which is also musical. We may test to see if any finite subsequence is musical by applying Rule 2 to the sequence, then checking that Rule 3 still holds. We continue to do this until we break Rule 3 or we have decomposed the sequence to length one. As base cases, both **L** and **S** are considered to be musical.

For example, consider the following two sequences and their decompositions:

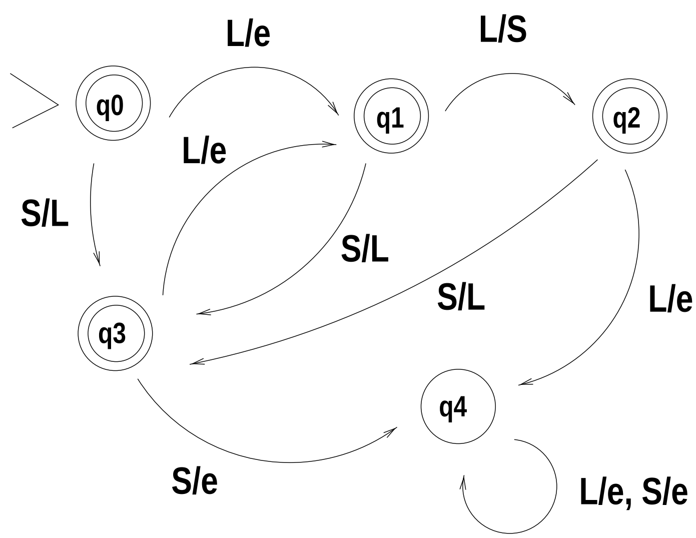
LSLSLLSLSLLSLLSL	**LSLSLLSLLSLLSLSL**
LLSLLSLSL	**LLSLSLSLL**
SLSLL	**SLLLS**
LLS	not musical (Rule 3)
SL	
L	

This gives us a recursive definition for musical sequences. Any sequence of length one is musical, and a sequence is musical if and only if it obeys Rule 3 and its decomposition under Rule 2 is also musical. In the above examples, both initial sequences obey Rule 3, so we repeatedly decompose them until either we obtain a sequence of length one, as in the first case, or we eventually obtain a sequence which contradicts Rule 3, as in the right case after two decompositions. We thus conclude that the left sequence is musical, while the right is not.

We may also construct deterministic finite automata (DFA) both to scan a sequence to ensure that it obeys Rule 3 and to decompose a sequence using the grammar of Rule 2. Figure 2.1(a) is a DFA which tests if the input string follows Rule 3. In this DFA, **q0** is the start state. Whenever the machine encounters a single **L**, it moves to state **q1**; if it reads two consecutive **L**'s, it will be in state **q2**. If the machine ever encounters three consecutive **L**'s, it will move into state **q4**, which is an error state. Once the machine is in the error state, it stays there regardless of the input. Similarly, if the machine



(a) DFA which verifies that a string follows Rule 3.



(b) DFA which decomposes a string using Rule 2.

Figure 2.1: Two DFA's which may be run simultaneously to perform one step of verification and decomposition.

ever encounters a single **S**, it will move to state **q3**, and if it ever reads two consecutive **S**'s, it will move into state **q4**. Since every state is final except for **q4**, the machine will accept any string which obeys the rules of Rule 3. Conversely, any string accepted by this machine must obey Rule 3, since a string with either three consecutive **L**'s or two consecutive **S**'s will move the machine into **q4**, and the machine will never accept once it reaches this state.

Figure 2.1(b) is a DFA which decomposes the input string using the grammar of Rule 2. Once again, **q0** is the start state. A single **L** will move the machine into state **q1**, writing nothing on the output string. Two consecutive **L**'s will move the machine into **q2**, writing an **S** on the output string. Finally, if the machine encounters an **S**, it will move to state **q3**, writing an **L** on the output string. Since this is a *deterministic* finite automaton, it accounts for the possibility of a violation of R3 with an error state similar to that of the previous DFA. In fact, a comparison of the two automata reveals that they behave identically with regard to input; upon reading any given string, both DFA's will always be in the same state. This realization implies that both steps (verifying Rule 3 and decomposing under Rule 2) may actually be done using only one pass through the string.

It is also interesting to observe that performing these operations requires only a small degree of context. In other words, although it is customary to read the string from beginning to end, we would get the same result by reading from the end of the string to the beginning. In fact, it is possible to break the sequence into **n** chunks, as long as we ensure that none of the breaks separate portions of a consecutive sequence of either **L**'s or **S**'s in the original string, and perform one step in parallel using **n** identical machines. In some sense, this is an important quality, since it ensures that only local rules and context are necessary to perform critical operations. This, of course, is a defining characteristic of aperiodic tilings, which must use local matching rules to grow a tiling aperiodically.

2.2 Forced Bars

We are interested in musical sequences because they give us information about forced bars and hence forced tiles in a tiling. The Ammann bars decorate the kites and darts so as to create a grid of parallel bars in five directions. These parallel bars are arranged such that they are always separated by either a long interval or a short interval; the ratio of the widths of the two types of intervals is 1 to τ. At first glance, these sequences of long and short intervals appear to be random. However, Conway has proven that these sequences are musical sequences.

We may use this fact to determine when the positions of bars are fixed by the presence of certain other bars. Whenever we know where at least two adjacent parallel bars are positioned, we can fix the positions of other bars. Figure 2.2 gives the next two bars which are forced by the presence of two initial bars separated by a long interval. We are shown in this figure all sequences of length six which are musical and which begin with an **L**. For the most part, the generation of these sequences is obvious. The main question involves the last sequence — why is the last **L** the only possibility? The answer to this question involves the requirement that the sequences be musical. A sequence of **LSLSLS** would satisfy Rule 3, but it's decomposition under Rule 2 is **LLL**, which is not musical. Therefore, these are the only four possible musical sequences of length six beginning with **L**. We can therefore see that the positions of two bars are fixed - one a short and a long interval distance from the previous fixed bar, and the second another short and two long interval distance from that one.

2.2.1 A Grammatical Algorithm

We have developed an algorithm which generalizes this method to any initial starting sequence and any integer number of bars which are to be fixed. The algorithm is given in Figure 2.3. The input to the algorithm is a string, *startSeq*, which must be a musical sequence, and an integer k, representing the number of fixed bars that we wish to find. The output will be a string, of the form $\mid SL \mid SLL \mid SL \mid$ ..., where the vertical lines represent fixed bars and the letters between the bars are an abstract representation of distance. They are not necessarily the actual sequence; rather, they reflect the distance between successive bars.

We first convert startSeq to the desired format, saving this information on a template string. We will eventually return this string at the end of our algorithm. We then add startSeq to the queue which will store possible sequences. Our goal is to generate all possible sequences which begin with startSeq and which are musical. We have a surrounding while loop which exits when we have found k fixed bars. Then, for each element of the queue, we remove it, make a copy of it, add an **S** to the end of one copy and an **L** to the end of the other copy, and check to see if either or both is musical using an

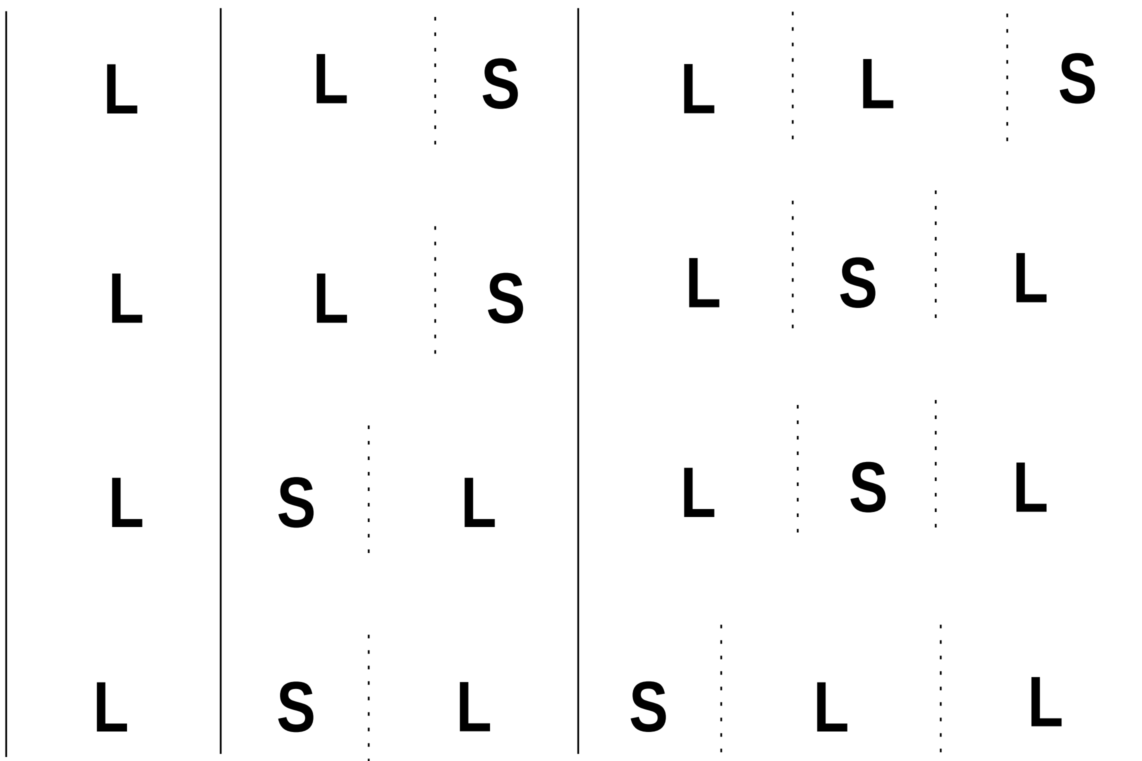


Figure 2.2: Ammann bars forced by the existence of two parallel lines separated by a long interval.

Algorithm FindForcedBars (*startSeq*, *k*)
Input: *startSeq* (a musical sequence of L's and S's) and *k* (an integer indicating the number of forced bars user wishes to find)
Output: string indicating positions of forced bars in the following form:
| L | S | L | SL | ...

```
begin
    initialize template using startSeq;
    insert template into Queue;
    numPoss = 1;
    while (number of known bars < k) {
        for (i = 0; i < numPoss; i++) {
            tempL = removeFromHead(Queue);
            tempS = tempL;
            concat(tempL, L);
            concat(tempS, S);
            if (isMusical(tempL)
                addToTail(Queue, tempL);
            if (isMusical(tempS)
                addToTail(Queue, tempS);
        }
        numPoss = length(Queue);
        temp1 = removeFromHead(Queue);
        {count L's and S's in temp since last forced bar}
        addToTail(Queue, temp);
        match = true;
        numChecked = 1;
        while ((match) and numChecked < numPoss) {
            temp = removeFromHead(Queue);
            match = {number of L's and S's in temp same as for temp1};
            addToTail(Queue, temp);
            numChecked++;
        }
        if (match) {
            {found position of next forced bar - add information to template}
        }
    }
    return template;
end
```

Figure 2.3: Algorithm to determine positions of k fixed Ammann bars given an initial sequence of bars separated by long and short intervals.

implementation of our deterministic finite automata from Figure 2.1. If so, we add them back on to the end of our queue. In effect, at the end of this process we will have increased the length of every string in the queue by one, so that we will have all strings of length n which are musical and which begin with startSeq. We then check each string in the queue to see if there are the same number of **L**'s and **S**'s since the last forced bar. If this is the case, the width of the sequence is the same for all legal strings. In this case, we have found our next forced bar, and so we add this information to the template. Once we have found k forced bars, we exit the loop and return the template string. We also note that we have all possible sequences available in the queue, so if this information is useful, as is often the case, we may pass back these values as well.

2.2.2 A Geometric Algorithm

We recall from Section 2.1 that a musical sequence may be generated by passing a line with slope τ through a two-dimensional grid. This fact gives us another method for generating musical sequences. Any musical sequence may be determined algebraically by generating the sequence $\lfloor n\tau + c \rfloor$ for $n \geq 0$, and then substituting S and L for differences of 1 and 2 respectively. Here c is any real number, and it may be taken to satisfy $0 \leq c < 1$. This is given in Rule 4. For example, if we take c to be 0, then we have the following sequence for n = 0,1,2,...:

0, 1.618..., 3.236..., 4.854..., 6.472..., 8.090..., 9.708..., ...

If we take the integer part of each element of the sequence, we have:

0, 1, 3, 4, 6, 8, 9, ...

For the difference between each pair of consecutive integers we have:

1 2 1 2 2 1 ...

This gives us a sequence as follows:

S L S L L S ...

The choice of c is, in some sense, a measure of the inverse density of the bars. Thus when c is 0 we have a sequence where the bars that delineate intervals are as close together as is allowed. Similarly, when c approaches 1, the bars are spread apart as far as possible. For example, when c is 0.999, the sequence which is generated begins as follows:

L L S L L ...

We may generate sequences which begin with certain patterns by restricting the value of c. For example, if we know that the first interval in a musical sequence is a long interval, then we know that $2 \leq \tau + c < 3$. Solving for c, we can see that this restricts c to values such that $c \geq 2 - \tau$, and also such that $c < 3 - \tau$. The second inequality does not prove too restrictive; we already knew that $c < 1$. However, the first inequality is an improvement; we now know that whenever the first interval is an L, c must be greater than or equal to 0.382...

We would like to use this information to predict which bars will be forced. As an example, consider the case when $n = 3$. This is equivalent to the position of the fourth bar. For c equal to 0.382... we have that $3\tau + c = 5.236...$, and for c equal to .999, we have $3\tau + c = 5.853....$ The integer part of either of these numbers is 5. Thus, every musical sequence beginning with L has the fourth bar fixed in a position which is a short interval and a long interval away from the second forced bar. This situation is pictured in Figure 2.2. An equivalent analysis may be performed for any integer n. This gives us a constant time algorithm for finding whether or not the n^{th} bar is forced, and a linear algorithm for finding the first n forced bars for sequences beginning with an L.

This algorithm may be generalized to any initial starting sequence of bars which are fixed. We simply use each piece of information that we know about the fixed bars to restrict the bounds of c. A given bar may or may not improve the bounds. If it does not, that implies that its position was already forced by the preceding bars. Once we have the minimum and maximum values for c for a given initial sequence, $\mathcal{S}$, we need only plug these numbers into the equation in order to either determine if a given bar is fixed by $\mathcal{S}$, or to find any number of bars forced by $\mathcal{S}$.

Graphically, we may interpret this process as given in Figure 2.4. A line with slope τ is passed through a two-dimensional grid. A *c-interval*, shown as a darkened line segment, gives the possible intercepts for this line on each vertical line.

Theorem 2.1 *Any musical sequence may be generated by choosing* $0 \leq c < 1$.

Proof: In order for the k^{th} Ammann bar to be forced, the c-interval must lie completely within a vertical section of the grid between two consecutive y values on the integer lattice for $x = k - 1$. Since the upper bound for c may be arbitrarily close to 1, this will only fit if the lower bound of the c-interval lies on a point on the integer lattice. However, if this were to occur, that

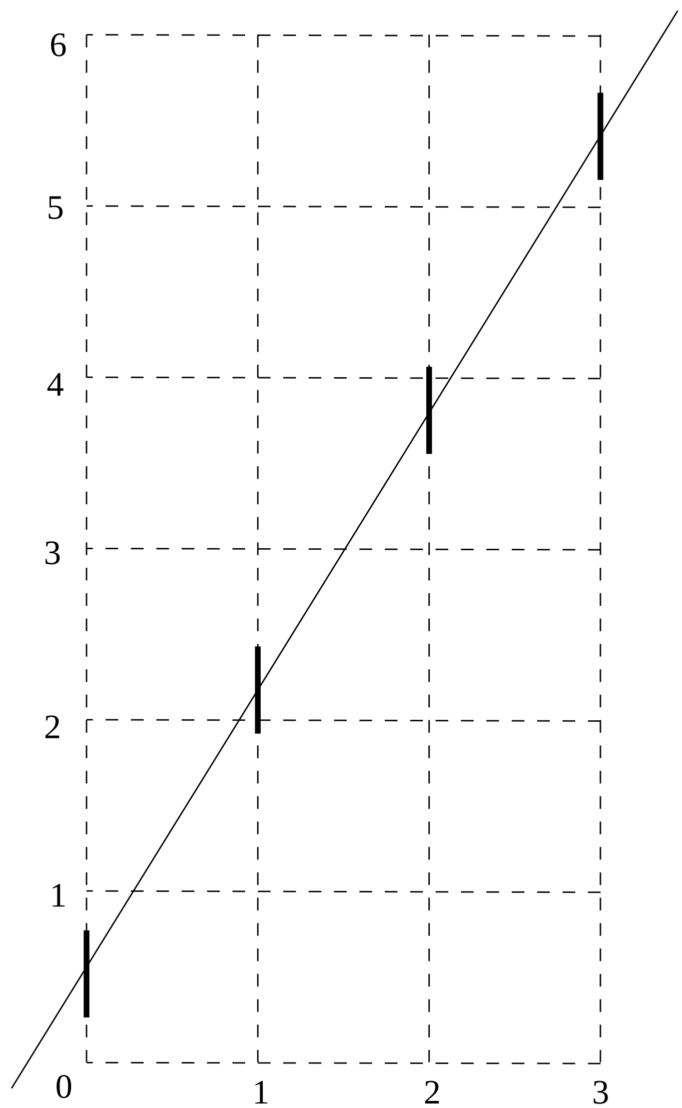


Figure 2.4: c-intervals for $.25 \leq c < .75$. The fourth Ammann bar (x=3) will be forced for any musical sequence within this range.

would imply that τ was rational. The lower bound is a line with slope τ, and that would be equal to the slope between two integer lattice points. Since τ is irrational, this is impossible, and hence, since no bars are fixed, any sequence may be generated. **Q.E.D.**

As a specific example, Figure 2.4 gives the c-intervals when $.25 \leq c < .75$. This is a somewhat contrived example, as the true upper and lower bounds will always be $i - j\tau$ for some $i, j \in Z$, but it will suffice to demonstrate the concept. The first bar is always forced — it is the bar which is placed down before the sequence of intervals begins. We can see that the second bar is not forced since the c-interval spans the third horizontal line. Therefore, the first interval may be either an **L** or an **S**. By a similar argument, we can see that the third bar is not forced, and so the second interval may be either an **L** or an **S** as well. Notice that it is not possible for **both** the first and second intervals to be short — if the first is short it is because the integer part of $\tau + c$ is 1, and in all cases, the integer part of $2\,\tau + c$ is at least 3. Thus, depending upon the value of c, our first two intervals must be **LS**, **LL**, or **SL**. The fourth bar is a different story, however. The minimum and maximum values for c both lie between $y = 5$ and $y = 6$, so the integer part of $3\,\tau + c$ will be 5 for any value of c within our allowed range. Hence the position of the fourth bar is fixed, forced by our constraints on the c-interval. Since the fourth bar is fixed at a distance 5 away from the origin, it implies that the sum of the first three intervals must be five, which translates to some permutation of **SLL**. We can see that all three are possible — for the minimum value of c, we will have **SLL** for our first three intervals, at the maximum value for c, the sequence will begin with **LLS**, and for c in the middle range (for example, the line drawn in the figure), we will have **LSL**.

2.2.3 Comparison

The first algorithm is more accurate: we will always be able to compute the correct answer as long as we have enough memory to store the possible combinations. It is less efficient, however, in use of both time and space.

The second algorithm is capable of finding the first n forced bars in linear time, which is an improvement in complexity over the first algorithm (the first algorithm has complexity of O(nlog(n))). It also uses only four integers worth of storage space. From these perspectives, it is a considerable improvement over the preceding algorithm. Automated generation of sequences and forced bars using this method, however, is subject to error caused by rounding off even a double-precision number. This will eventually create a problem, especially as the upper and lower bounds for c tighten.

Chapter 3

Ammann Lines and Vertex Stars

3.1 Ammann Lines

Penrose kites and darts may be decorated with markings known as Ammann lines. The Ammann lines for these two prototiles are given in Figure 3.1 and Figure 3.2. In any infinite tiling, the Ammann lines will be extended through the tiling to form a grid of parallel lines in five directions, separated by angles of $72°$. In Figures 3.1 and 3.2, the dashed lines extend the lines as they would be found in an actual tiling. The dotted lines with arrow heads are vectors from each vertex of each prototile extending orthogonal to each Ammann line found on that particular tile. The lengths of all of the vectors and line segments may be determined using elementary trigonometry. The data is given in Table 3.1.

We notice that each vertex of a prototile has three vectors, each orthogonal

a	$sin(54°)(\frac{2\tau+1}{2\tau})$	1.059017	v	$\frac{1}{4}(3-\frac{tan(36°)}{tan(18°)})$	0.190983
b	.25	.25	x	$\frac{2\tau+1}{2\tau}$	1.3090169
c	$cos(36°)+.75$	1.559017	y	$\frac{1}{2\tau}$	0.3090169
d	$a-1$	0.059017	z	.25	.25
e	$\tau-b$	1.368034	w	.75	.75

Table 3.1: Lengths of vectors and line segments in Figures 3.2 and 3.1.

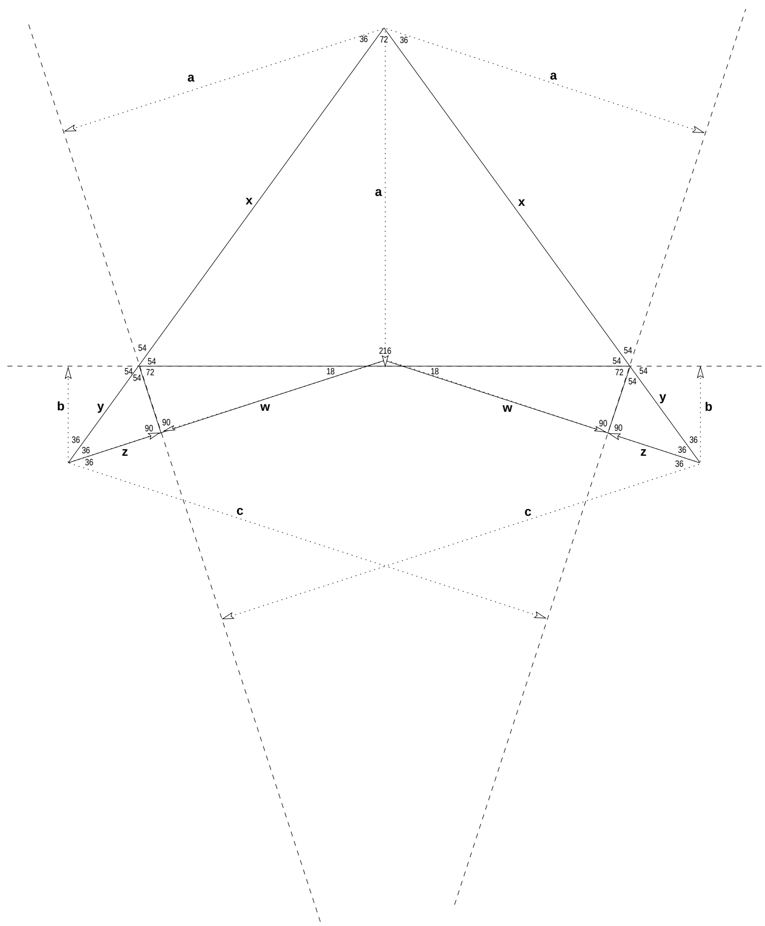


Figure 3.1: A dart decorated with Ammann lines and the vectors from each vertex orthogonal to each line fixed by the presence of a dart.

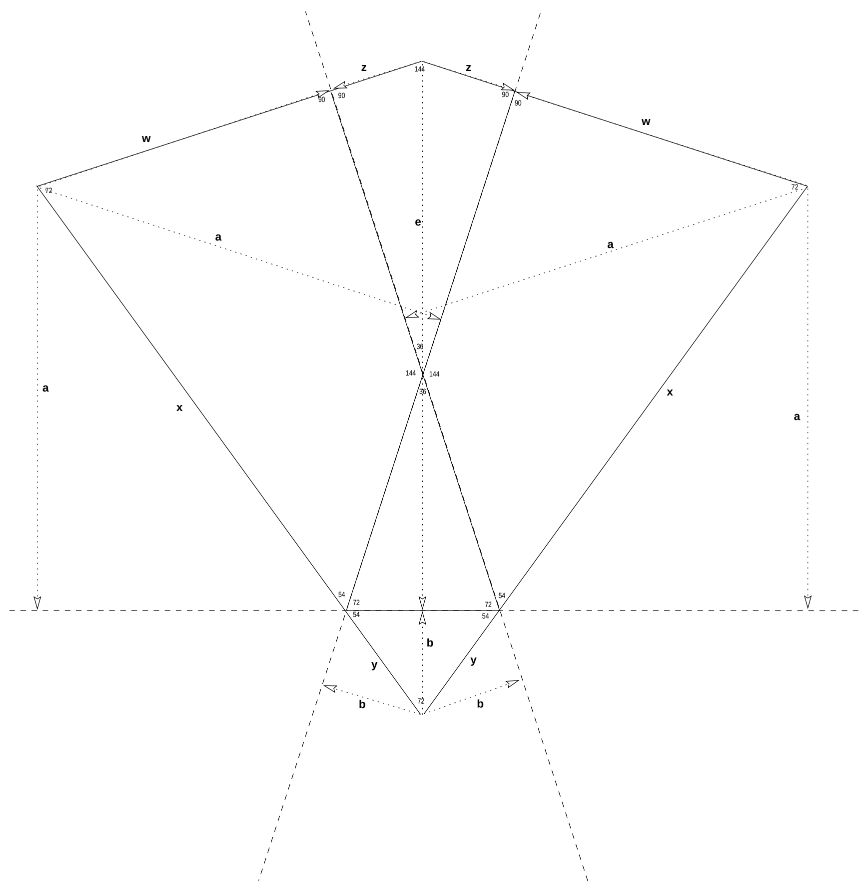


Figure 3.2: A kite decorated with Ammann lines and the vectors from each vertex orthogonal to each line fixed by the kite.

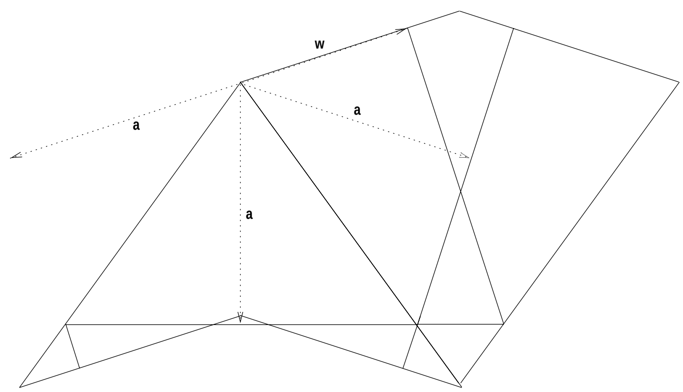


Figure 3.3: The **a** vectors overlap when a kite and dart share a long edge.

to an Ammann line. Whenever two tiles are adjacent to one another in a vertex configuration, they will share either one or two of these vectors, depending on how many Ammann lines are continued from one tile to the next. For example, in Figure 3.3 we notice that when a kite and a dart are placed such that the long edges touch, both **a** vectors overlap. When a kite and dart are oriented so that the short edges touch, only one Ammann line is continued and only one pair of vectors (the **w** vectors) coincide. This is true in general for the Penrose kites and darts.

We may now examine the vector stars produced by each of the vertex configurations. They are given in Figure 3.4. Since we know that there are two overlapping lines for each long edge and one overlapping line for each short edge, to find the number of points in each vertex star, we merely count the number of tiles in the vertex configuration, and then add one for every short edge that we find. We see that the number of points in the vertex stars ranges from five to seven, and that the number of tiles in a vertex configuration ranges from three to six.

The configurations of the vector stars lend some insight to the observation that some vertex configurations seem likely to be forced while others are almost never forced. An *empire* is the set of tiles, both adjacent and non-adjacent, which are forced by the presence of an initial patch of tiles. This is the terminiology which is used by Conway. If we consider the empires which are given in Appendix A, we see that the king vertex configuration is never forced in any of the empires associated with the seven configurations given in Figure 1.4. Inspection of the vector star gives us a hint as to why this may

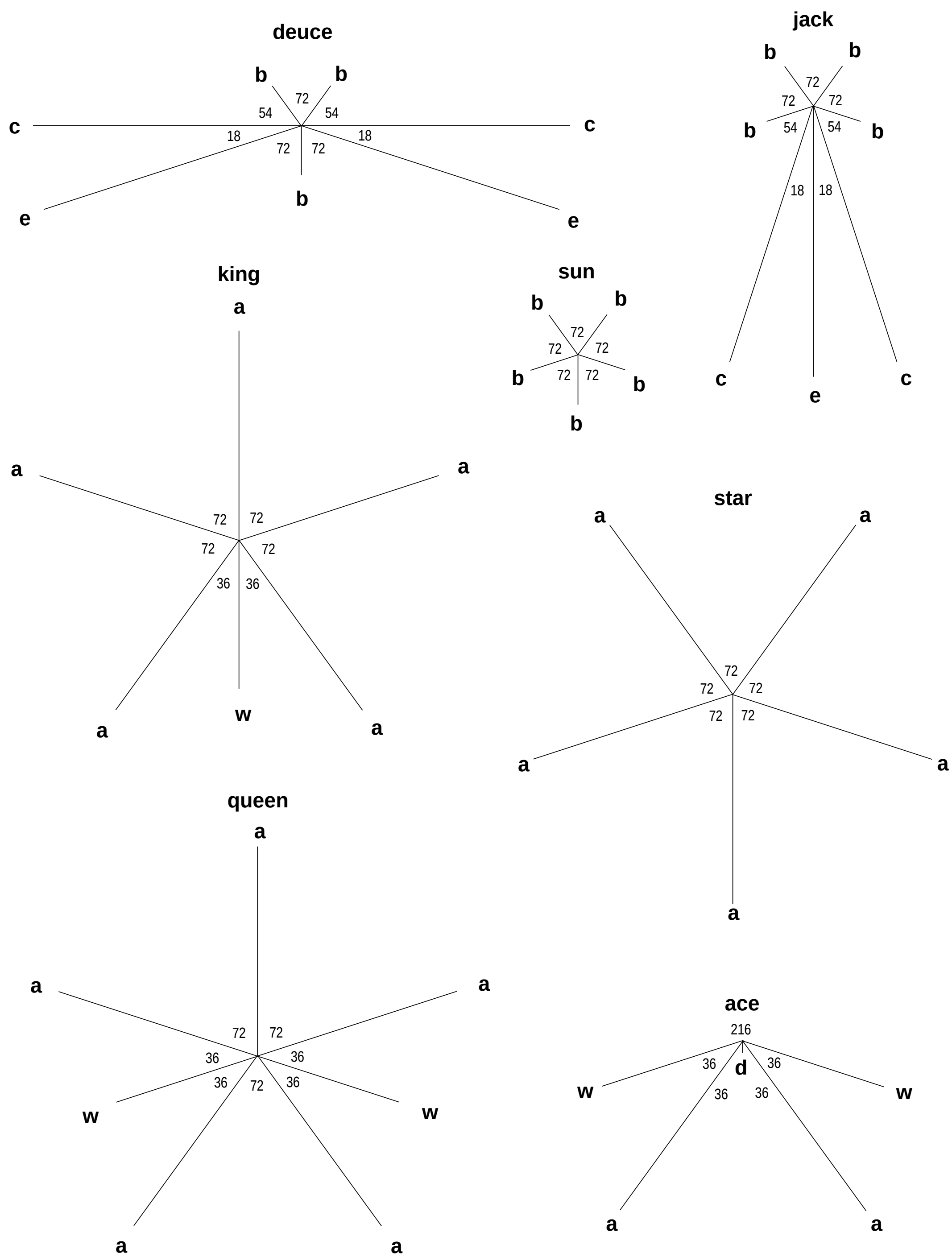


Figure 3.4: Vertex stars — vectors from vertices perpendicular to Ammann lines for each of the seven vertex configurations in Figure 1.4.

be true. The king's vector star consists of six vectors. If we consider any five of the six vectors, we notice that that arrangement is found in some other vector star. Therefore, the presence of all six vectors, and therefore bars, is necessary in order to force the king. By contrast, consider the two pairs of **a** and **w** vectors in the ace vector star. The only other star which contains this configuration is the queen, so we need only fix the position of one additional line in order to determine whether an ace or a queen is forced.

Information about vertex stars is useful for computation. A computer is capable of exhaustively checking each vertex to see if some required number of bars is forced, and using that information to mark tiles which are forced. This underscores the difference between generating a list of forced tiles manually or automatically. The human eye can see patterns and shapes, but a program must have information which is much more precise. However, this task requires exhaustive seach techniques which are much more suited to automatic methods than manual ones. In this sense, vertex stars improve our ability to allow a program to do the work of finding forced tiles for us.

Chapter 4

Forced Tiles

Our analysis of musical sequences and Ammann bars was motivated by the idea that this information could be used to improve our understanding of forced tiles in an infinite Penrose tiling. In this chapter, we will explore how information about forced Ammann lines can provide information about forced non-adjacent tiles.

4.1 Individual Tiles

We recall from Figure 1.6 that the kite and dart may be decorated with Ammann lines. Since the markings are the same for every copy of each tile in a tiling, it follows that certain arrangements of forced lines in a tiling will pin-

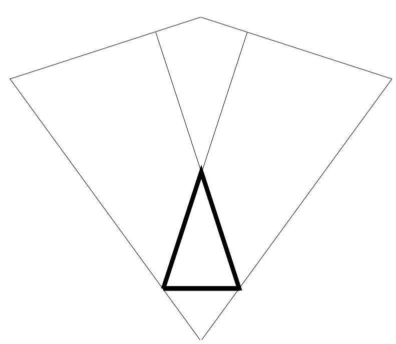


Figure 4.1: A kite marked with Ammann lines. The heavy lines form an isosceles triangle which indicates the presence of a forced kite in a tiling.

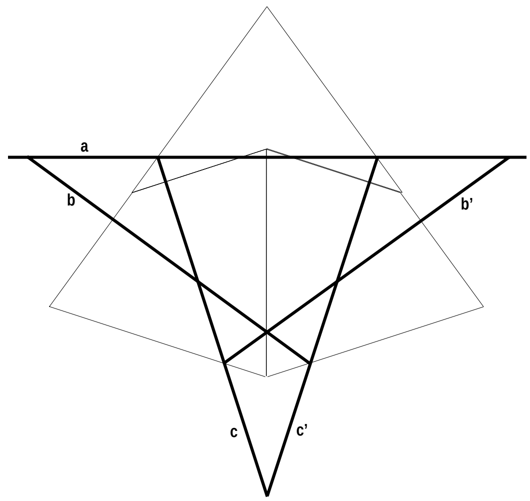


Figure 4.2: An ace marked with Ammann lines. An ace is automatically forced by a dart, so this gives us the lines which determine the presence of a forced dart.

point the placement of specific tiles in the tiling. A concrete example makes this easier to visualize. Figure 4.1 gives a kite marked with Ammann lines. We notice that the decorations form an isosceles triangle with angles of 72°, 72°, and 36°. Triangles which are similar will also be found, but we know the lengths of the sides of the particular triangle which forces a kite by the relative size of the original tiles. This arrangement of lines is sufficient to guarantee the presence of a kite in a tiling. In other words, if, after our analysis of forced bars in a tiling from Chapter 2, we find this arrangement of lines, we know that we have found a location in the tiling that must contain a kite, and so that kite is forced. This will be our first rule for finding forced tiles. It is given as Rule 5.

We recall from Chapter 1 that two kites are immediately forced by the presence of a dart in a tiling. It turns out that lines from all three of these tiles are necessary in order to force the presence of a dart, and hence an ace, in a tiling. Figure 4.2 gives an ace marked with Ammann lines. We notice that there are five lines present on these three tiles. The only line which is not essential in order to force this cluster of tiles is one of the pair marked **c** and **c'**. If one of these lines is removed, the remaining lines are still sufficient to force an ace. We may prove this graphically.

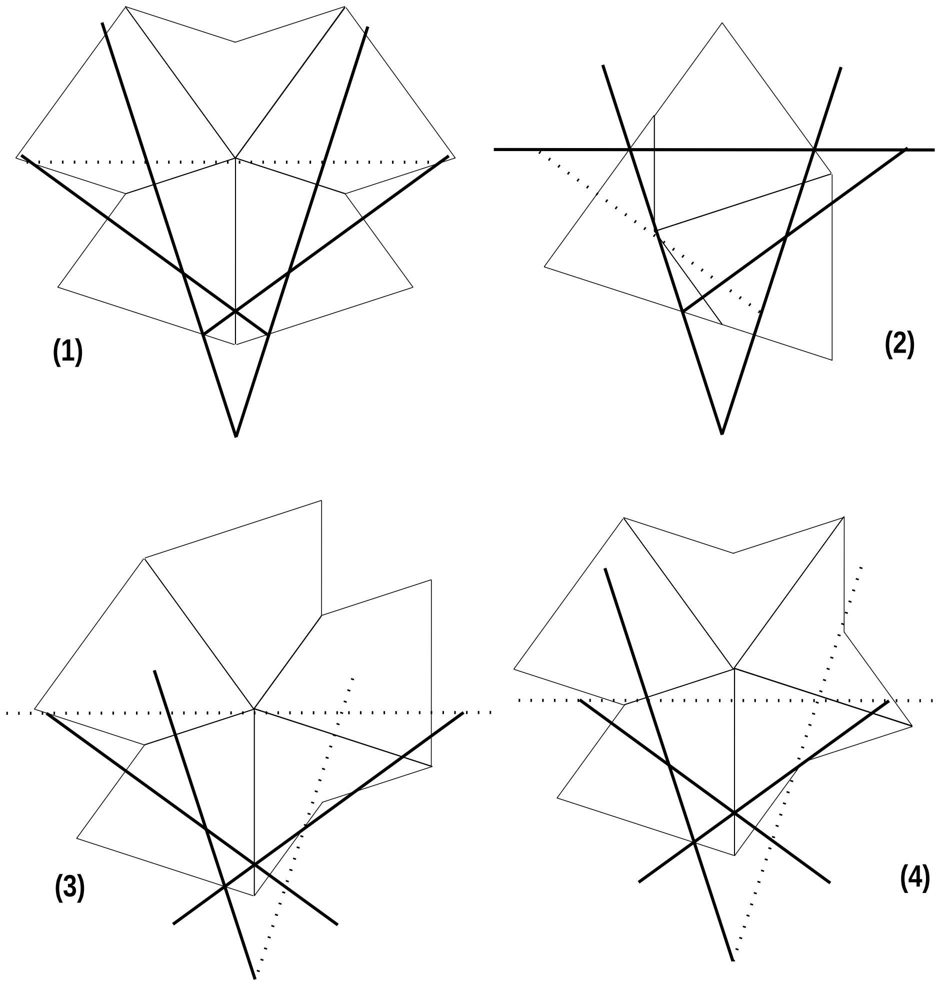


Figure 4.3: Counter examples to forcing an ace which are consistent with fewer lines. Dotted lines are inconsistent with one or more of the forced lines.

Theorem 4.1 *Lines a, b, b', and either c or c' from Figure 4.2 are necessary and sufficient to force an ace in a tiling of Penrose kites and darts.*

Proof: In Figure 4.3(1) we see that if line a is not forced, then a queen may exist in the position where $\mathbf{b}$, $\mathbf{b}'$, $\mathbf{c}$ and $\mathbf{c}'$ meet. Similarly, if line $\mathbf{b}$ is not forced, we see in (2) that an ace which is rotated 108° may exist in the location where $\mathbf{a}$, $\mathbf{b}'$, $\mathbf{c}$, and $\mathbf{c}'$ meet. By symmetry, we may also see that $\mathbf{b}'$ is necessary to force an ace — simply consider the mirror image. We now wish to consider removing line $\mathbf{c}'$. We note that lines $\mathbf{b}$, $\mathbf{b}'$, and $\mathbf{c}$ form the isosceles triangle necessary to force a kite, as shown above. If $\mathbf{c}'$ were forced, this would force another kite adjacent to the first kite, rotated 72°. Therefore, we need to look at clusters of tiles where $\mathbf{b}$, $\mathbf{b}'$, and $\mathbf{c}$ are forced, but not $\mathbf{c}'$, which occurs when we have a kite adjacent to a dart along the long edge. This particular arrangement occurs in the queen, king, and jack vertex configurations. We may consider only the king and queen, since the jack forces either of these two configurations in the location in which we are interested. We see in (3) that the queen fixes $\mathbf{b}$, $\mathbf{b}'$, and $\mathbf{c}$, but neither $\mathbf{c}'$ nor $\mathbf{a}$ will occur in the necessary places. Thus, if $\mathbf{c}'$ is the only line which is not forced, the queen cannot occur in that place. We can make a similar argument for the king, as seen in (4). Therefore, $\mathbf{c}'$ is not necessary to force an ace — if we have the other four lines, then the ace is the only patch of tiles which may occur in that location. Once again, we may argue that, by symmetry, $\mathbf{c}$ is also not necessary to force an ace if the other four lines are present. Thus, an ace is forced whenever all five lines occur in the given arrangement, or when either $\mathbf{c}$ or $\mathbf{c}'$ is absent, but all four of the other lines are present. **Q.E.D.**

This is given as Rule 6. In practice, all five lines are usually forced wherever an ace is forced.

4.2 General Algorithm

Our algorithm for determining which tiles are forced by a given patch is quite simple and intuitive. We first place the patch, then use knowledge about allowed vertex configurations to place any adjacent forced tiles. After finding which Ammann bars are fixed by the presence of this (possibly extended) patch of tiles, we use one of the algorithms from Chapter 2 to determine which other bars are forced. From these forced bars, we may use Rule 5 and Rule 6 to place forced tiles.

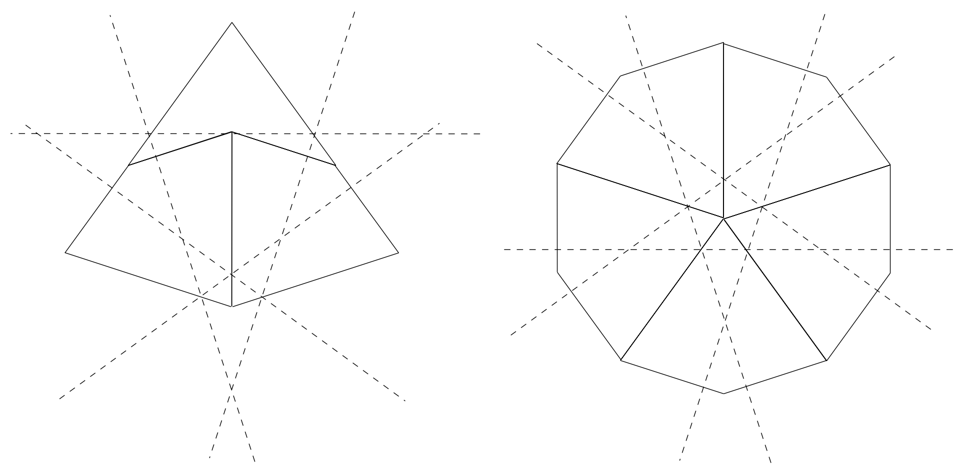


Figure 4.4: The sun and ace vertex configurations decorated with Ammann bars. There are no forced parallel lines, thus neither of these patches of tiles forces any non-adjacent tiles.

4.3 Empires of Vertex Configurations

Two of the vertex configurations, the ace and the sun, do not force any tiles, adjacent or non-adjacent. This stems from the fact that the vertex configurations force a single ine in each of the five directions. This is pictured in Figure 4.4.

4.3.1 Deuce

In Figure A.1, a deuce is colored purple, and the grey tiles are forced by the deuce. We notice that the original patch of tiles forces two adjacent kites. Among the Ammann lines fixed by these six tiles are two pairs of parallel lines, each separated by a long interval. Two adjacent bars separated by a long interval forces a series of bars beginning with the following pattern:

| **L** | **SL** | **SLL** | **SL** | **SLL** | **SLL** | **SL** | **SLL** | **SL** | **SLL** | **SLL** | **SL** | ....

We can see that the two pairs of parallel lines are oriented so as to force a kite along the horizontal line wherever a bar is forced, creating a collinear set of kites forced by the deuce, stretching to infinity in both directions.

We would like a method of describing the empire of the deuce in such a way that it is possible for us to be able to tell, arbitrarily distant from the deuce, whether a specific kite is forced or not. We notice that all of the forced kites are separated by either a distance which consists of a long interval and a short interval, or two long intervals and a short interval. We notice that the ratio of **SL** to **SLL** is 1 to τ, and so it is likely that these forced kites actually form a Fibonacci sequence. This particular sequence may be generated using the grammar of Rule 1.

If we take SL to be "S" and SLL to be "L", the sequence is constructed as follows:

S L
S LS
S LSL
S LSLLS
S LSLLSLSL

We notice that the leading S is completely ignored in our iterative process — it is there simply as a placeholder. The above grammar is repeatedly applied to the right-hand portion of the sequence. This gives us the ability to predict forced tiles arbitrarily far from the initial patch using an iterative process. This is not a closed form solution, but it does allow us to predict where kites will be forced arbitrarily distant from the initial patch. A finite amount of information thus encodes an infinite portion of the tiling, a fact which is crucial to storing information about infinite tilings on finite systems, such as computers. This description gives all of the forced kites which are collinear with the deuce. We will see in the next chapter, however, that many more kites which are not collinear are actually forced by the deuce.

4.3.2 Star

It is natural for us to next consider the star vertex configuration, since the star is in essence five overlapping deuces, each rotated 72° from the previous one. In Figure 4.5, we notice that the star and the ten adjacent kites which are immediately forced fix five pairs of parallel lines, each separated by a long interval. These lines force bars in the same pattern as for the deuce, and the intersections of the forced bars in turn force tiles.

4.3.3 Jack, Queen, and King

The forcing patterns of the remaining three vertex configurations are complex because the bars which are fixed by the original patch of adjacent tiles are not all separated by the same distance, and so the lines which are forced do not

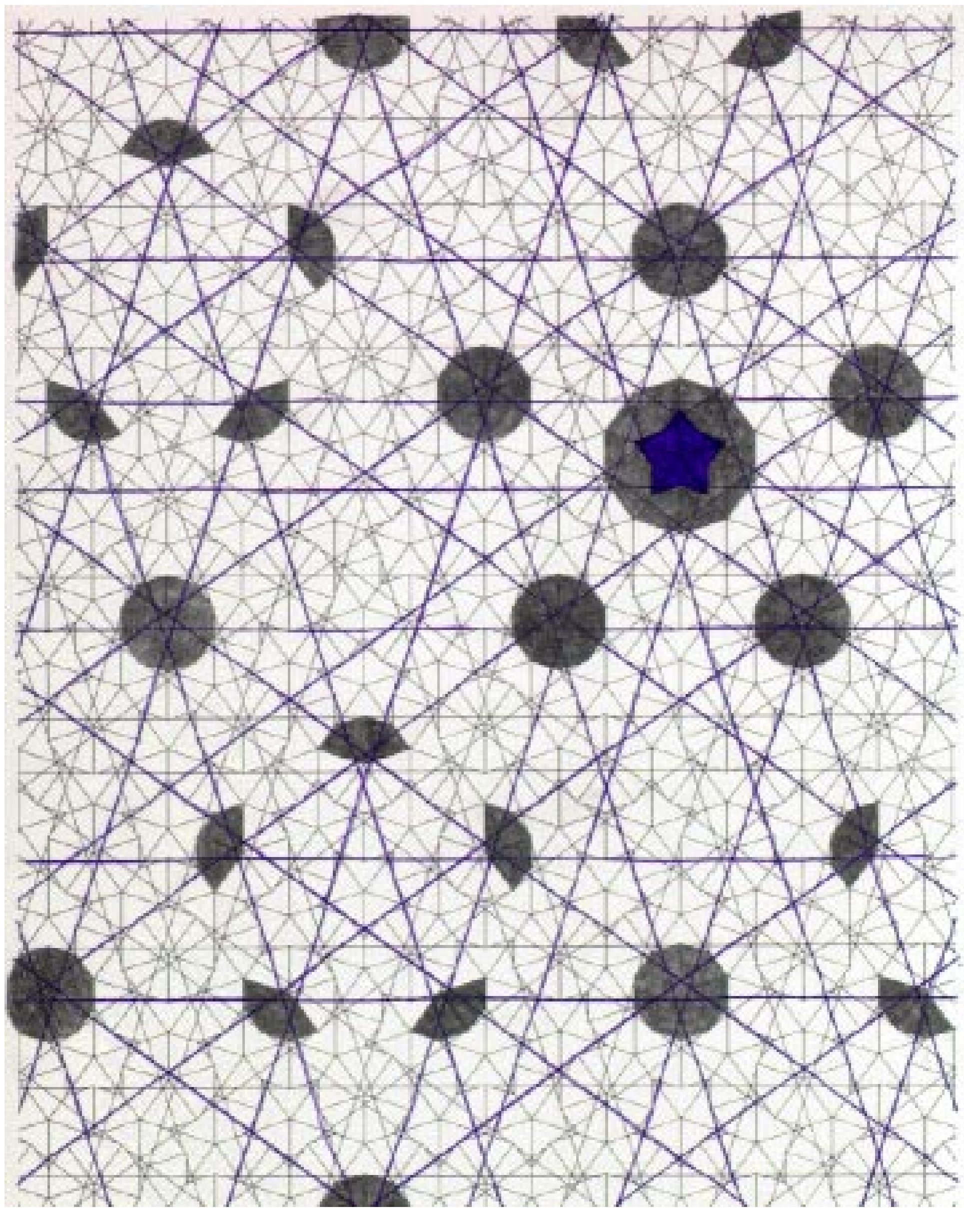


Figure 4.5: A subset of the tiles forced by a star vertex configuration.

intersect in patterns which are as straightforward to describe. For the moment we will simply note several properties about the patterns of non-adjacent forced tiles.

First, the sizes of the patches which are forced are related to the size of the original patch. The forced tiles adjacent to the jack in Figure A.4 form a relatively small patch, and the jack forces single kites, clusters of two kites, and aces. By contrast, the adjacent empire of the king in Figure A.7 is effectively twice as large, and tiles are correspondingly forced in larger clusters. We see aces, suns with two darts on non-adjacent pairs of kites, the adjacent empire of the jack, and other clusters of equivalent size.

Figure A.5 gives a portion of the queen's empire. The shape of the adjacent tiles forced by the queen is roughly a diamond, or rhombus. By contrast, the shape of the adjacent tiles forced by both the jack and king form coffin shapes, or regular trapazoids.

Third, identical patches of tiles oriented in the same direction are present in patterns which appear to be musical sequences, although the actual patterns are more difficult to describe because many sets of lines contribute to the forcing, as opposed to the simple pattern which we have seen so far for the deuce.

4.4 Further Influences

The method presented here does indeed find tiles which are forced from an initial patch. This is not the entire story, however. Published figures of the star's empire [GS87] indicate that many more tiles are forced than would be predicted by taking an initial patch, finding which bars are fixed, determining forced bars from any pairs of parallel bars, and then using the forced bars to fix the position of non-adjacent tiles as given in Figure 4.5. There is another step involved, and that will be the subject of the next chapter.

Chapter 5

Empires Revisited

In order to account for certain patches of non-adjacent forced tiles in the star's empire, we must both define an additional rule and insert another step into our algorithm for finding forced tiles.

5.1 Rogue Patches in the Star Empire

Figure 5.1 is a portion of a tiling containing a star vertex configuration. The purple tiles are the original patch, and the solid purple lines are the Ammann bars which are both fixed by the original patch and the adjacent forced tiles as well as those which are subsequently forced by the pairs of parallel lines. The light grey tiles are those forced tiles which are determined by the solid purple lines. According to a published figure of the empire of the star [GS87], the ace which is colored green should also be forced by this star. Three of the lines necessary to force an ace, a, c, and c', are forced by our previous method. However, this is not enough to force the presence of this ace. We recall from Figure 4.3 (2) that if either b or b' is not forced, then the ace is not forced — it is possible for an ace rotated $108°$ to exist in that location. Therefore, we must extend our model to include an explanation for why b and b', the two Ammann lines colored red in the figure, are forced.

5.1.1 Explanation

Since the star has five-fold rotational symmetry, it is enough for us to explain why just one of these lines is forced; this will suffice as an explanation for both

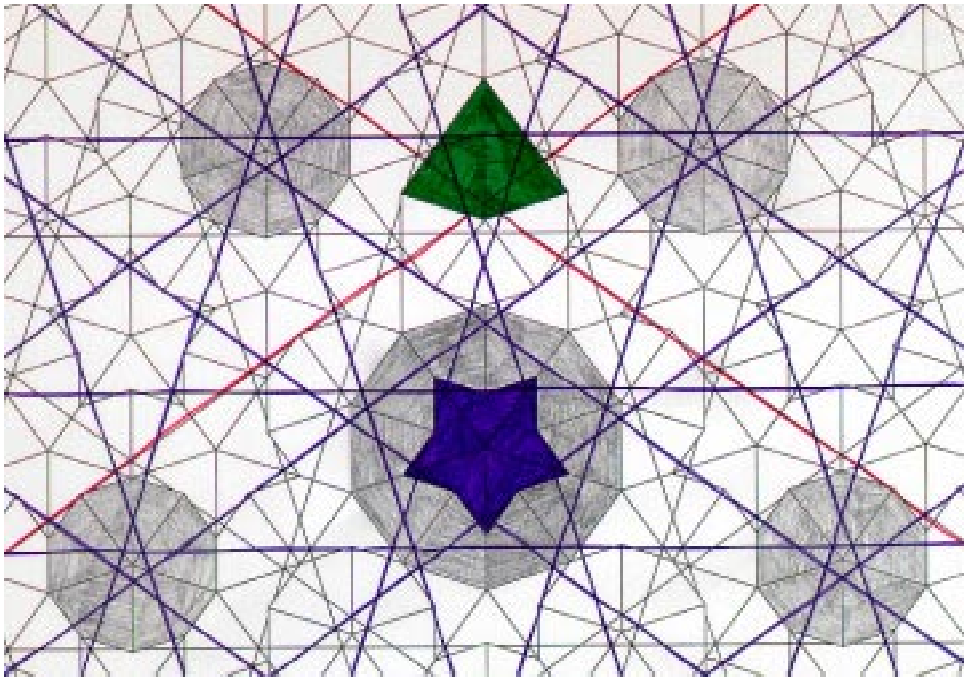


Figure 5.1: A subset of the tiles forced by a star vertex configuration.

of them. Working under the assumption that the red lines are indeed forced, we notice that they are adjacent to a pair of parallel lines separated by a long interval. We know that, in a musical sequence, a single L does not force an S as the next element. On the contrary, it may be extended either with LLS... or LSL... . Therefore, it is not the bars fixed by the patch which are forcing this line. It follows that perhaps some property of the underlying tiles results in the forced placement of the line.

The line which must be forced is orthogonal to two kites which meet along their respective short edges. If we consider the seven legal vertex configurations, as given in Figure 1.4, we notice that in only two of the configurations, the king and the queen, we find two adjacent kites which share a short edge along with more tiles which exist "above" the two in which we are interested. We can see in Figure 5.2 that in both of these arrangements, two parallel bars which form a short interval are fixed. Note that the mirror image of the queen is another possible arrangement of tiles, but that does not change the result. Thus, these two tiles force the bar above them, although none of the three tiles themselves is forced. The result is given in Rule 7.

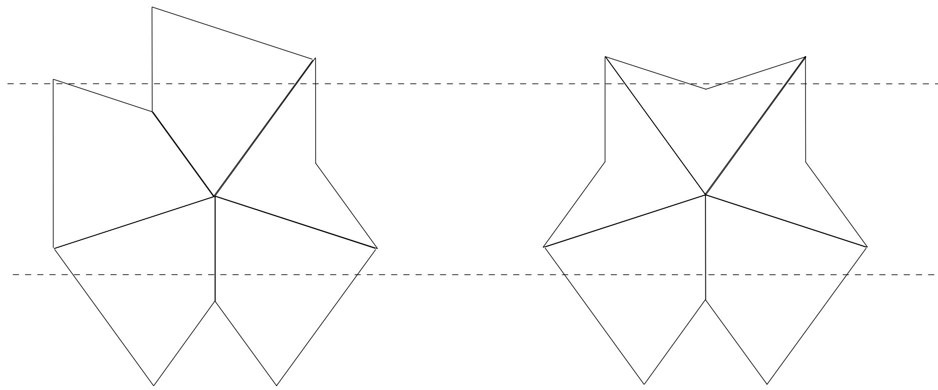


Figure 5.2: When two adjacent kites share a short edge, an Ammann line is forced whose distance from the parallel line already determined by the tiles is equivalent to the distance between “short” intervals.

5.1.2 Rule and Algorithm Modification

We now have a rule which does not fit in the same class as the rules we have previously discussed. Other rules involve either sets of parallel lines which force additional parallel lines or patterns of line intersections which force certain tiles to exist in those locations. This rule falls in neither of these two catagories. It states that wherever two kites which share a short side exist, a bar is forced which forms a short interval with the parallel bar already fixed by the two tiles, as shown in Figure 5.2. This will be significant in determining non-adjacent forced tiles whenever this arrangement of tiles occurs on the boundary of a patch of tiles, and we wish to know which non-adjacent tiles are forced by the patch.

Our modified algorithm for finding forced tiles uses these concepts. Given a patch of tiles, we first find any adjacent tiles which are forced due to knowledge of the allowed vertex configurations. Once this process is complete, we mark any Ammann lines which are fixed by this possibly extended patch. We next scan the edges of the patch for instances of Rule 7, and, if they occur, we mark the line which is forced as a result. The next steps are identical to the previous algorithm — we find the bars which are forced by any sets of parallel lines, and then use the rules given in Chapter 4 to fix the locations of any forced tiles. This method generates the empire of the star, given in Figure A.3.

5.2 The Expanded Empire of the Deuce

Our method has ramifications which affect the empires of even simple patches of tiles. The deuce vertex configuration contains two kites which share a short edge. In fact, two kites arranged in this way is enough to force a deuce. We can see in Figure A.2 that two lines separated by a short interval are forced by the deuce, and these two lines force infinitely many parallel lines. We can see that more kites are forced by this process. Thus, the empire of the deuce is actually two-dimensional instead of one-dimensional. This greatly complicates our ability to describe this empire. It is no longer a collinear set of kites streching to infinity in two directions. Thus our description of the empire must be modified to be accurate as a result of this discovery. This is a new result. The red tiles are not mentioned as elements of the deuce's empire in previously-published figures [GS87]. In fact, previous descriptions explicitly state that the empire of the deuce consists of a collinear set of kites, which, as we can see from Figure A.2, is certainly not the case, once we factor in the effects of Rule 7.

5.3 Additional Tiles in the Queen's Empire

We note in Figure A.5 that the queen forces deuces throughout its empire. Thus, we may utilize Rule 7 to fix bars in the queen's empire. Figure A.6 gives the expanded empire of the queen. The red bars are those fixed via Rule 7, and the red tiles are the tiles which are subsequently forced. The red tiles are new elements of the empire of the queen — they are not present in previously-published figures of this empire [GS87].

Chapter 6

Algebraic Interpretations of Forced Bars

6.1 Introduction

We will begin by considering aperiodicity in the one-dimensional case. We recall from Chapter 2 that musical sequences are aperiodic sequences of long and short intervals. Musical sequences may be generated by passing a line with slope τ and y-intercept c, where $0 \leq c < 1$, through a two-dimensional grid of integer points. The particular sequence is generated by taking the integer part of y for each integer x. This generates a sequence of integers in which consecutive elements are separated by either one or two. We may replace each one-difference with an S and each two-difference with an L. This generates the musical sequence.

We may use this method to understand forced bars in a musical sequence. If no elements of the sequence are given, any sequence is possible. We can think of this as a band with slope τ with a vertical width of one. A bar is only forced when the c-interval passes completely between two consecutive y values on the integer lattice, though it may be incident to the lower value. We can see that this will never happen if the vertical width of the c-interval is one. We know from Theorem 2.1 that if no bars are forced initially, no subsequent bars are forced, and so any sequence is possible. Once we begin to set the positions of certain bars, this has the effect of improving the bounds of c. Graphically, we may interpret this as the width of the c-interval decreasing. We can see in Figure 6.1 that the improvements in the width of the c-interval

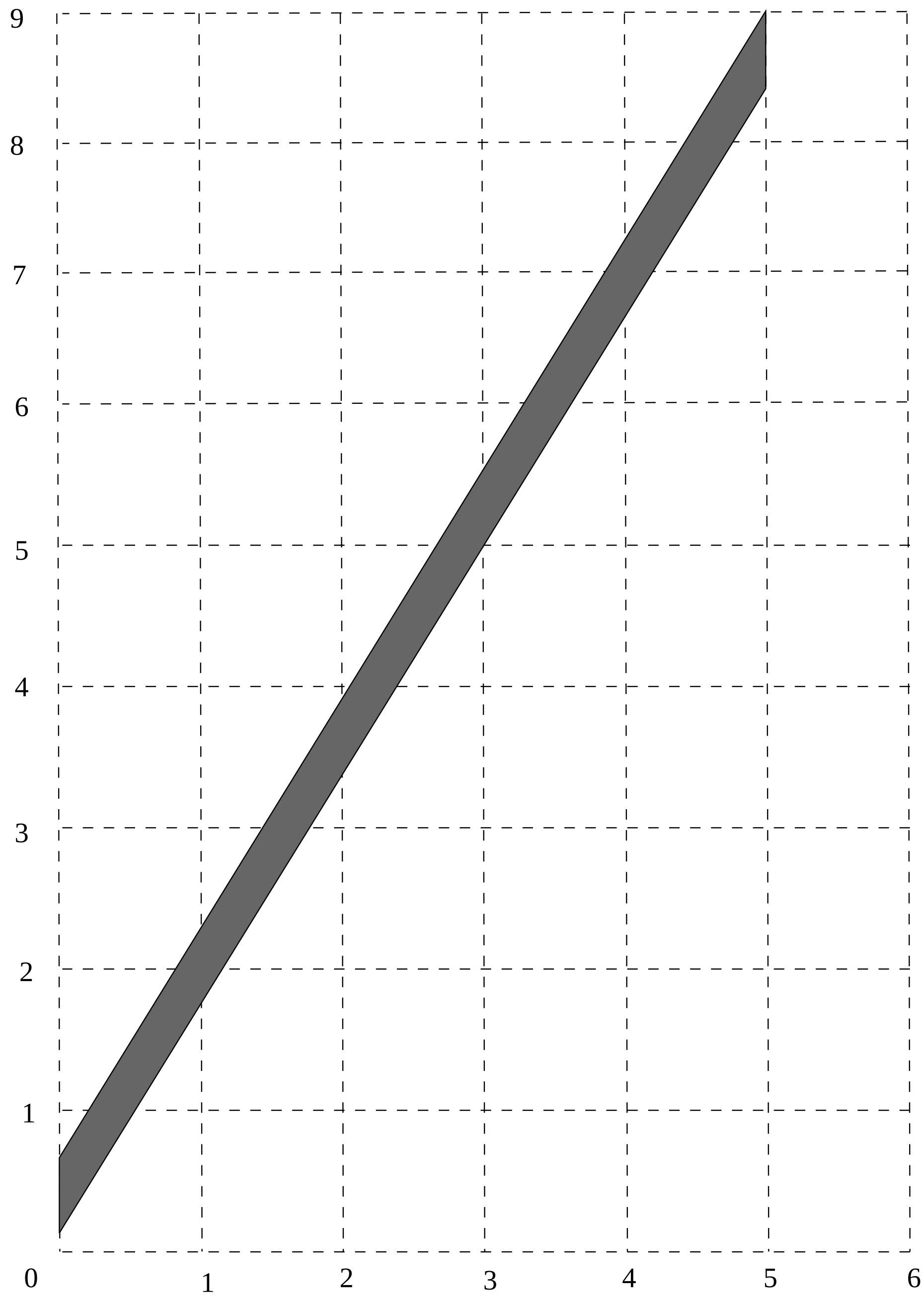


Figure 6.1: A c-interval for a sequence in which bars are forced for x=2, x=3, and x=5.

in this case have occurred where $x = 3$ and where $x = 5$. We can see that the second bar is not forced, since the c-interval passes through the point (1,2) on the lattice. On the other hand, the third bar is forced since when $x = 2$, the entire c-interval passes between $y = 3$ and $y = 4$. When $x = 3$, the lower bound of the c-interval passes through the point $y = 5$, which indicates that this was the last point at which the lower bound for c was improved. We can see that when $x = 4$, the c-interval once again passes through the lattice point, so the Ammann bar is not fixed. Finally, when $x = 5$, the upper bound of the c-interval passes through the point $y = 9$, and so this bar is fixed, being the last point at which the upper bound for c was improved.

This representation of musical sequences is quite powerful, in part because it allows us to predict certain properties of forcing patterns given some initial sequence. This is valuable to our pursuit because forcing in two-dimensional aperiodic tilings is caused to some degree by forcing of musical sequences in five directions. In practice, it is much more complicated as a result of the need to take the relationships between different sequences into account. However, it is possible to predict some properties of forcing patterns.

6.2 Motivation

Theorem 6.1 *The number of consecutive bars forced by an initial patch of n bars will be less than n.*

Proof: In order for n or more bars to be forced, it must be the case that a c-interval of the given width could be placed such that it lay completely between two consecutive integer values for y for n consecutive values of x. We know, however, that there is some value for x, call it i, between $x = 0$ and $x = n - 1$ where the upper bound of the c-interval passes through some integer point, and some value for x, call it j, where the lower bound of the c-interval passes through a point on the integer lattice. Since the c-interval has slope τ, and τ is irrational, it is not possible for either the upper bound or the lower bound of the c-interval to pass through any additional points on the integer lattice. Using this information, we can see that it is not possible for x values equivalent to both i and j to be forced. Assuming without loss of generality that $i < j$, it is only possible to force either $j-1$, or $n-i-1$ bars. In addition, it is possible that bars could be unforced before the bar at $x = 0$. In this case, it would be possible to force a number of bars greater than $j - 1$, but this number will always be significantly less than n. **Q.E.D.**

Intuitively, this proof works because the c-interval cannot "fit" between the two points on the integer lattice which bound it above and below in the second instance of n forced bars. For this to happen, either the upper and lower bounds of the c-interval would have to pass through integer values of y, which is impossible since τ is irrational, or some point on the interior of E would have to pass through an integer point, in which case the bar would not be forced.

6.3 Forced Bars

The method of this proof suggests that we may in fact improve this bound. If $j = n$, where n is the length of the original sequence, then the upper bound for the number of forced bars is indeed $n - 1$. Our next task will be to show that j must actually be significantly less than n.

We approach this problem by considering c-intervals of decreasing width. It is intuitive that as the width of the bar decreases, the distance between successive "choices" in the positions of bars will increase. A choice corresponds to a value for x for which the c-interval spans a lattice point. It is an unforced bar, so the upper bound for the c-interval and the lower bound for the c-interval straddle an integer value. A thick bar with constant irrational slope is more likely to intersect a point on the integer lattice than a thin bar with the same slope. In some sense, this is due to the probability that the area covered by the bar will include certain types of points.

Table 6.1 gives the results of an experiment in which a c-interval with width w was created whose lower bound was the origin. The number in the second column, b, corresponds to the number of bars whose positions were fixed before the first unforced bar was encountered. The values in the first column are related — beginning at one, each value is the product of the previous value with $\frac{1}{\tau^2}$. The a_i's which are given are the largest value for w for which the corresponding b_i is the number of bars which are forced. The b_i's are also related — they form a sequence which is composed of the even-indexed Fibonacci numbers. It is interesting to note that the number of forced bars jumps from each b_i to the next. There are no values of w for which any intermediate values of b exist.

Similarly, Table 6.2 gives the results of an experiment in which the upper bound of a c-interval was improved so that the width became w. We notice

Width (w)	Number of Forced Bars (b)
$a_0 = 1$	$b_0 = 1$
$a_1 = \frac{a_0}{\tau^2} \approx 0.381966$	$b_1 = 3$
$a_2 = \frac{a_1}{\tau^2} \approx 0.145898$	$b_2 = 8$
$a_3 = \frac{a_2}{\tau^2} \approx 0.055728$	$b_3 = 21$
$a_4 = \frac{a_3}{\tau^2} \approx 0.0212862$	$b_4 = 55$
$a_5 = \frac{a_4}{\tau^2} \approx 0.0081306$	$b_5 = 144$
$a_6 = \frac{a_5}{\tau^2} \approx 0.0031056$	$b_6 = 377$
$a_7 = \frac{a_6}{\tau^2} \approx 0.0011862$	$b_7 = 987$
$a_8 = \frac{a_7}{\tau^2} \approx 0.0001531$	$b_8 = 2584$
...	...
$a_n = \frac{a_{n-1}}{\tau^2}$	$b_n = round(b_{n-1} * \tau^2)$

Table 6.1: Number of forced bars (b) immediately following an improvement of the lower bound of the c-interval such that its width is w.

Width (w)	Number of Forced Bars (b)
$d_0 = 1$	$c_0 = 1$
$d_1 = \frac{a_0}{\tau} \approx 0.618034$	$c_1 = 2$
$d_2 = \frac{a_1}{\tau^2} \approx 0.2360679$	$c_2 = 5$
$d_3 = \frac{a_2}{\tau^2} \approx 0.0901699$	$c_3 = 13$
$d_4 = \frac{a_3}{\tau^2} \approx 0.0344418$	$c_4 = 34$
$d_5 = \frac{a_4}{\tau^2} \approx 0.0131556$	$c_5 = 89$
$d_6 = \frac{a_5}{\tau^2} \approx 0.0050249$	$c_6 = 233$
$d_7 = \frac{a_6}{\tau^2} \approx 0.0019193$	$c_7 = 610$
$d_8 = \frac{a_7}{\tau^2} \approx 0.0007331$	$c_8 = 1597$
...	...
$a_n = \frac{a_{n-1}}{\tau^2}$	$b_n = round(b_{n-1} * \tau^2)$

Table 6.2: Number of forced bars (b) immediately following an improvement of the upper bound of the c-interval such that its width is w.

that in this case, the bounds of the widths, the d_i's, are the inverses of odd multiples of τ, and that the number of bars which are forced are the odd-indexed Fibonacci numbers.

This result is illuminating. Returning to our previous discussion, it gives us some information about where in a group of forced bars the i and j values may exist. In order to continue this discussion, we must first assume that our original patch of forced bars was complete, meaning that if the size of the patch was n, then the bar where $x = n + 1$ was **not** forced.

We can see that we have an instance of a case from both Table 6.1 and Table 6.2 within our n forced bars. At some point, the upper bound of the c-interval passes through an integer point, and at some point the lower bound of the c-interval passes through some integer point.

This idea is pictured graphically in Figure 6.2. In this case, i is the point at which the upper bound was improved and j is the point at which the lower bound was improved. We know that the number of bars between i and j is some odd-indexed Fibonacci number, and that the number of bars between j and the end of the sequence of forced bars is some even-indexed Fibonacci number. Furthermore, the number of bars between the beginning of the sequence of forced bars and i must be less than or equal to the number of bars between j and the end of the sequence. This is true because the width of the c-interval must be constant. So, if the lower bound was improved at j, then rotating the lattice by $180°$ gives us that i, which used to be an improvement of the upper bound, becomes an improvement of the lower bound. Hence, since some number of bars, possibly 0, were forced before i, that number must be less than or equal to the number of bars forced after j, by symmetry.

Consider the case where $n = 30$. We need to decide which even-indexed and which odd-indexed Fibonacci number to use. If we were to use 13 and 21, their sum would be greater than 30, which is impossible. If we were to use 5 and 8, their sum plus the 8 possible bars before i would be less than 30, which is impossible. Thus, the two boxed Fibonacci numbers, 8 and 13, must be used. We can see in this case that it is impossible for $i < j$ since the number of bars before either of the two must be 9, which is more than 8. Hence, it must be the case that we have 9 forced bars, then a bar where the lower bound is improved (j), then 8 more forced bars, then a bar where the upper bound is improved (i), and finally 13 additional forced bars.

In other examples, when $n = 25$ we must once again use 8 and 13, but in this case both orders of i and j are possible. When $n = 34$, we use 13 and 21. Since 34 is itself a Fibonacci number, it is the sum of the two previous

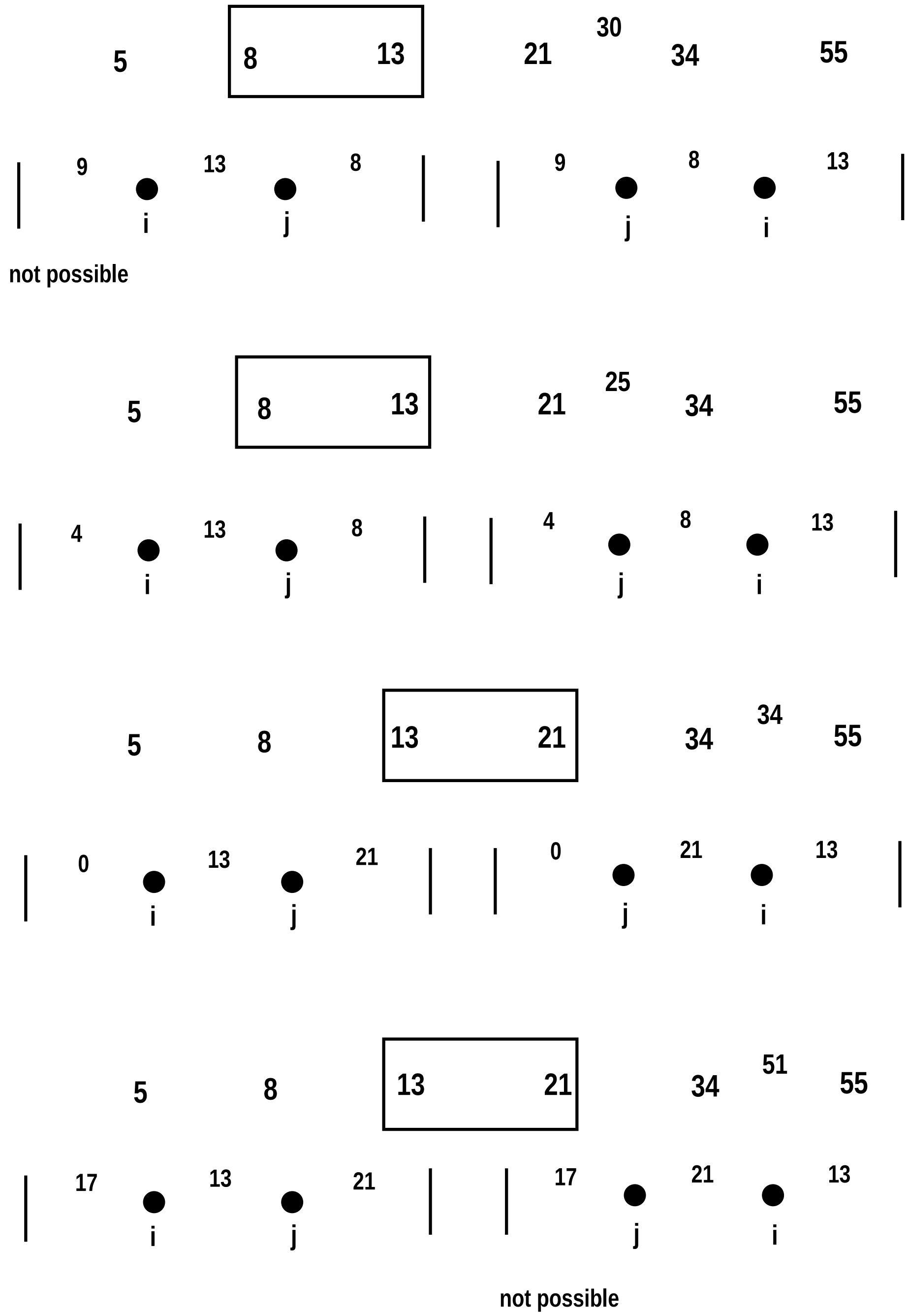


Figure 6.2: Numbers of bars between improvements in the upper and lower bounds for the c-interval.

Fibonacci numbers. Therefore, there are no bars before the first point at which a bound was improved. When $x = 51$, we once again use 13 and 21. In this case, it is not possible for $j < i$. We can see from the data that if n is between two Fibonacci numbers, then the width of the c-intervals must be in the range that will force a number of bars equivalent to the two Fibonacci numbers before the ones which bound n.

We can use this information to find bounds on the width of the c-interval which may force n bars. We can see that for any integer n, there are Fibonacci numbers f_1 and f_2 such that $f_1 \leq n < f_2$. There are two cases to consider.

First, suppose that $f_1 = c_m$ and $f_2 = b_m$ for some integer m. Then the Fibonacci numbers that we will need to use are c_{m-1} and b_{m-1}. We can see from Table 6.1 that in order to force b_{m-1} bars, we must have a width w such that $a_m < w \leq a_{m-1}$. Similarly, from Table 6.2, we can see that to force c_{m-1} bars, we must have $d_m < w \leq d_{m-1}$. We can see that for any integer i, $a_i < d_i$, so by combining these two restrictions to give one which will force both b_{m-1} bars and c_{m-1} bars, we have that $\mathbf{d_m < w \leq a_{m-1}}$.

Second, suppose that $f_1 = b_m$ and $f_2 = c_{m+1}$ for some integer m. Then we need to restrict the width so that b_{m-1} bars and c_m bars will be forced. From Table 6.1, to force b_{m-1} bars, we need $a_m < w \leq a_{m-1}$. From Table 6.2, to force c_m bars we need $d_m + 1 < w \leq d_m$. We notice that for any integer i, $d_i < a_{i-1}$. Thus, combining the restrictions, we have that $\mathbf{a_m < w \leq d_m}$.

Therefore, given a number of bars in a complete patch of forced bars, we may determine both an upper bound and a lower bound for the width of the c-interval for the patch of bars. Now that we know the width of the c-interval, we may perform another experiment in which we determine how many bars may be forced by a c-interval of a certain width. We do this by ensuring that the bar preceding the patch of forced bars is *not* forced — we do this by beginning with the c-interval passing through the origin. We start with the bottom of the c-interval incident to the origin, and repeatedly move the entire bar down by some small amount until the top of the c-interval is incident to the origin.

The results of this experiment provide us with a great deal of information. It is helpful to consider each case separately. In the first case, when $c_m \leq n < b_m$, $d_m < w \leq a_{m-1}$, and the number of bars forced will always be c_{m-1}-1, b_{m-1}-1, or c_m-1. c_m-1 bars are only forced when n is close to b_m. In the second case, when $b_m \leq n < c_{m+1}$, $a_m < w \leq d_m$, and the number of bars forced is always b_{m-1}-1, c_m-1, or b_m-1. As before, b_m-1 bars are only forced when n is close to c_{m+1}. Thus the result is the same in each case — if $f_m \leq n < f_{m+1}$, then the number of consecutive bars which are forced may be either f_{m-2}-1, f_{m-1}-1, or f_m-1, where f_i is the ith Fibonacci number. This is stated as Rule 9.

We have found an upper and lower bound to the number of consecutive bars which may be forced by a patch of a given size. In addition to bounds, we have exact values for the number of consecutive bars which may be forced by any given initial patch of forced bars. Since forced bars position forced tiles, this information gives rigor to our intuition that larger initial patches of tiles force large patches of tiles relative to smaller initial patches. As an example, consider Figures A.4 and A.7. We see that the adjacent tiles which are forced by the jack is much smaller than the adjacent patch forced by the king. Consequently, the patches of non-adjacent tiles which are forced by the king are larger than those non-dajacent patches of tiles forced by the jack.

6.4 Proof[1]

Whenever we have a series of n consecutive forced bars we develop a c-interval with bounds $c_{\min}$ and c_{max}. (Recall the c-interval is a half-open interval of offsets, c, that uniquely distinguish the aperiodic sequence of intervals suggested by $\lfloor i\tau + c \rfloor$, where i is integral, and τ is the golden section.) If the n consecutive bars (without loss of generality, located at $x = 0, 1, 2 \ldots (n-1)$) are the only contraints on the tilings, we know that $c_{\min}$ and $c_{\max}$ are determined by lattice points $(i, i\tau + c_{\min})$ and $(j, j\tau + c_{\max})$, where $0 \leq i, j < n$. We cannot be certain, however, of the ordering of i and j and, as we shall see, it is generally unimportant.

We turn to a geometric interpretation of continued fractions (see, for example, Stark). The golden ratio is the simplest infinite, periodic continued fraction, written as

$$a_0 + \frac{1}{a_1+}\ \frac{1}{a_2+}\ \frac{1}{a_3+} \ldots = 1 + \frac{1}{1+}\ \frac{1}{1+}\ \frac{1}{1+} \ldots$$

and we have that $a_i = 1$ for all $i \geq 0$. Geometrically, if we define

$$V_i = (x_i, y_i) = \begin{cases} (1,0) & i = -2 \\ (0,1) & i = -1 \\ (a_i x_{i-1} + x_{i-2}, a_i y_{i-1} + y_{i-2}) & i \geq 0 \end{cases}$$

[1]The analysis in this section is due to Duane A. Bailey

then the points V_i are successively closer approximations to a line with slope determined by the value of the continued fraction. In the case of τ, this simplifies to

$$V_i = (x_i, y_i) = \begin{cases} (1,0) & i=-2 \\ (0,1) & i=-1 \\ (x_{i-1}+x_{i-2}, y_{i-1}+y_{i-2}) & i \geq 0 \end{cases}$$

If we define the Fibonacci number F_i by the recurrence

$$F_i = \begin{cases} i & i=0,1 \\ F_{i-1}+F_{i-2} & \text{otherwise} \end{cases}$$

then we see that

$$V_i = (x_i, y_i) = \begin{cases} (F_1, F_0) & i=-2 \\ (F_0, F_1) & i=-1 \\ (F_{i-1}, F_{i-2}) & i \geq 0 \end{cases} .$$

In this case, the points V_i are successively closer to a line of slope τ, giving us successively better approximations $\frac{y_n}{x_n}$ to τ (see Figure 6.3). We can get a better grasp on what this means, with the following definitions from Stark:

Definition 1 *Let L be the line $y = \tau x$. We define the C,* set of points closest *to L, to be the set of all points (p,q) such that p and q are integers, $p \geq 1$, and having the property that the distance from (p,q) to L is less than or equal to the distance from (m,n) to L with these distances being unequal if $m < p$.*

We may now make a statement about the points selected by the continued fractions algorithm for a line of slope τ.

Theorem 6.2 *V_0, V_1, V_2,... are members of C, the set of points closest to L, and are successively closer to L. Furthermore, V_0, V_2, V_4,... are all below L, while V_1, V_3, V_5,... are all above L.*

The proof follows, inductively, the construction of the set of points V_n using the continued fractions algorithm. In general, the set of points closest to L may contain points other than V_i, in the case of constructing an approximation to τ, we can see that that C and the set of points V_i are the same.

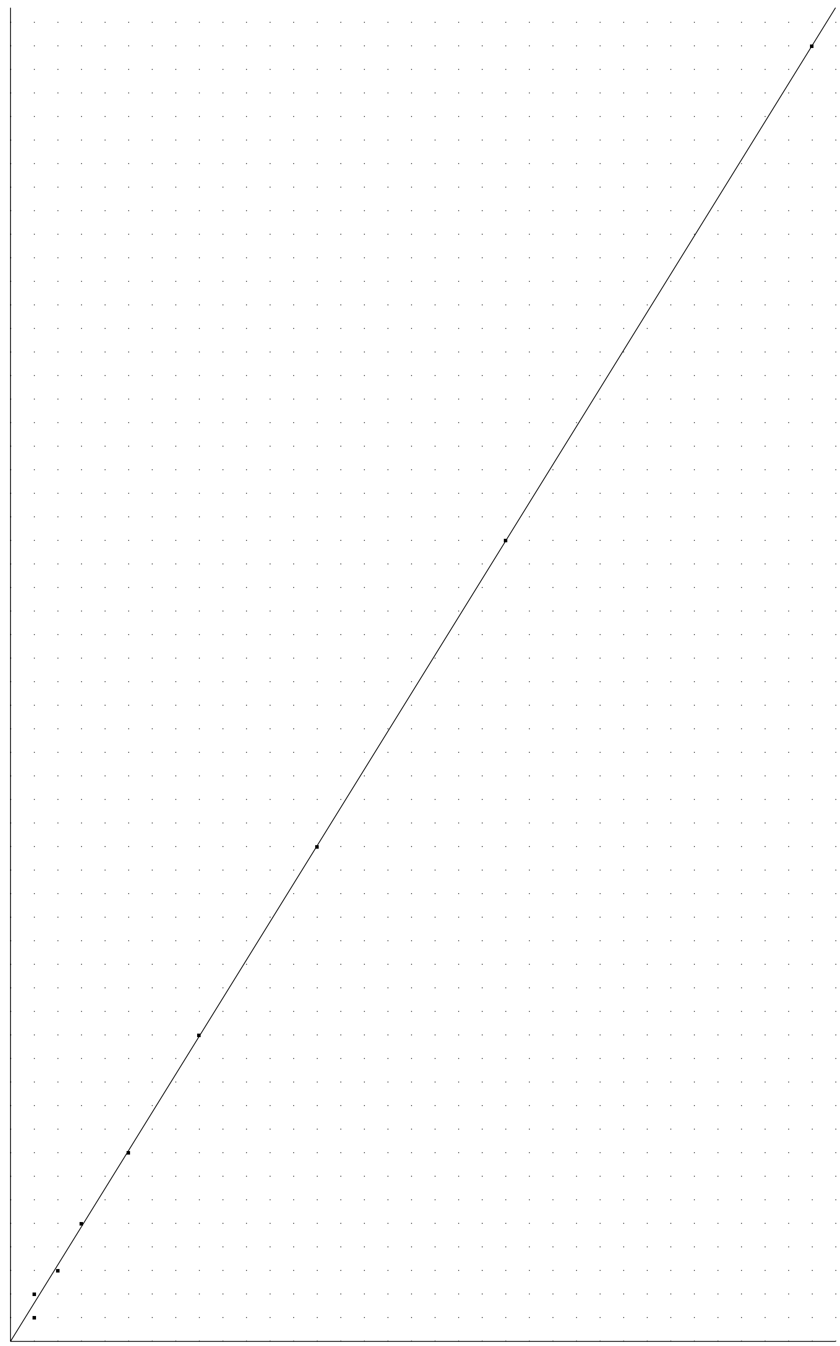


Figure 6.3: The convergents of the continued fraction approximating τ. Highlighted points $\frac{F_{i+1}}{F_i}$ are successively closer to the line $y = \tau x$.

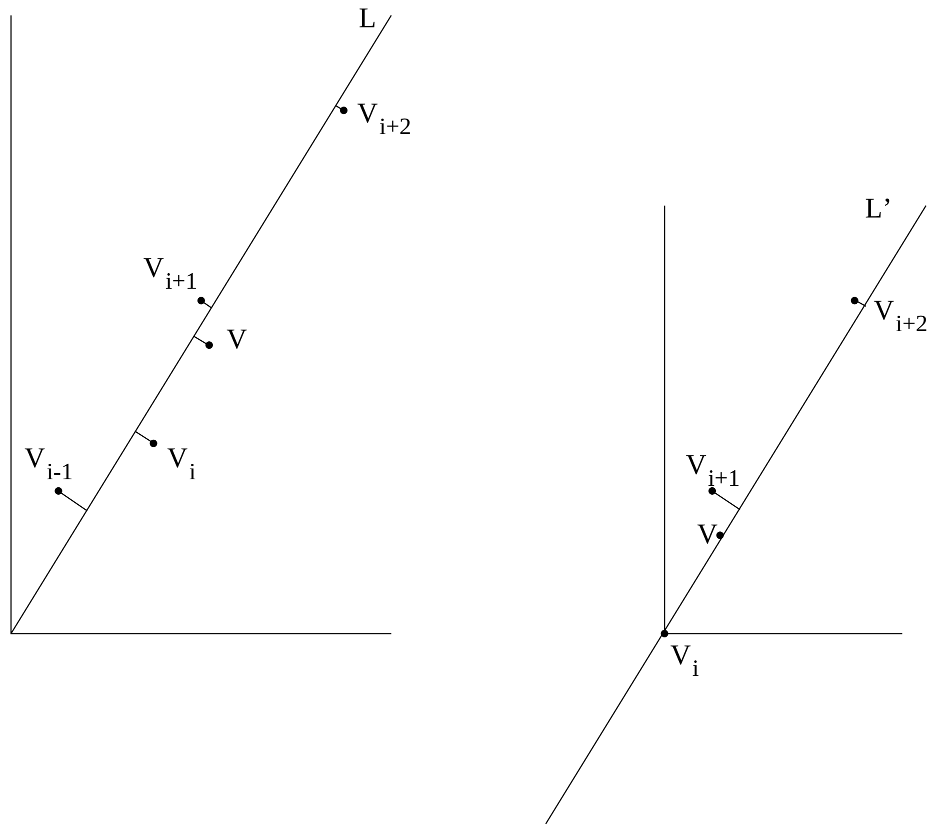


Figure 6.4: V cannot be a closest point because it contradicts the closest nature of point V_{i+1} to L (or V_{i+2} to L').

Theorem 6.3 *Let L be the line $y = \tau x$. The set of points V_i, $i >= 0$ is precisely the set C of points closest to L.*

Proof: First, by Theorem 6.2, we know that for $i \geq 0$, $V_i \in C$.

We need only demonstrate that each point of C is a point V_i. Suppose this were not the case. Then let $V = (x, y)$ be a point of C that is not a V_i for any $i \geq 0$. Suppose, without loss of generality, that V falls below the line. Let each V_i be distance d_i from L, while V is distance d from L. For some i V falls between V_i and V_{i+2}, both of which are also below L. Then $x_i < x < x_{i+2}$ (it is easy to see these equalities are strict if V_i is a closest point), and $d_i > d > d_{i+2}$.

We note that because the continued fraction algorithm gives us coefficients $a_i = 1$, we have $V_{i+1} = V_{i-1} + V_i$ and $V_{i+2} = V_i + V_{i+1}$. Geometrically, we can think of the vector from the origin to V_{i+1} as being the same as the vector from V_i to V_{i+2} (see Figure 6.4). Suppose, now, we draw a second coordinate system with origin at V_i and include L', a line of slope τ through V_i. Clearly, V_{i+2} is a closest point to L' in this system. However, V, which sits on a lattice point, is farther from L than V_{i+2}, but is *closer* to L', contradicting the fact that V_{i+1} and V_{i+2} are closest points to L and L', respectively. It must be that any point of C is also a V_i. **Q.E.D.**

Now that we know all the closest points to line L, we gain valuable information about the number of bars forced by c-intervals of various sizes.

Suppose, for example, that a c-interval is tightly bounded below by a lattice point $(i, i\tau + c_{\mathrm{m}in})$, and tightly bounded above by $(j, j\tau + c_{\mathrm{m}ax})$. What is the relationship between i and j?

Suppose $i < j$. We simply recenter the coordinate system at point $(i, i\tau + c_{\mathrm{m}in})$. The point that corresponds to $(j, j\tau + c_{\mathrm{m}ax})$ is a closest point above the line that determines the lower bound by running through the origin. Since these points have the form (F_{2k}, F_{2k+1}), we know $j - i = F_{2k}$ for some $k > 0$, and thus there are F_{2k} bars forced from i to the left of j. In addition, we know the next point at which a integer lattice point falls within the bar is at (F_{2k+2}, F_{2k+3}). There are, then, F_{2k+2} bars forced to the right of i and past j.

By symmetry, we can see that a similar argument can be made for forcing $F_{2k+2} - F_{2k} - 1$ bars to the left of i. The total is, then $2F_{2k+2} - F_{2k} - 1$ bars. In the case, for example, when $i = 0$ and $j = 1$, $2k$ is 2 and we get a short interval that forces a long on either side, for a total of $2F_4 - F_2 - 1 = 2 \cdot 3 - 1 - 1 = 4$ bars.

Similarly, if $j < i$, then we recenter the coordinate system about $(j, j\tau + c_{\mathrm{m}ax})$ and find a closest point below the top of the c-interval at F_{2k+1} units away and $i - j = F_{2k+1}$ for $k > 0$. F_{k+1} consecutive bars are forced at j and to the right until i. The next choice point is at location (F_{2k+3}, F_{2k+4}), and we see that this combination of i and j forces a total of F_{2k+3} bars to the right of j. Counting another $F_{2k+3} - F_{2k+1} - 1$ the left of j we have a total number of forced bars $2F_{2k+3} - F_{2k+1} - 1$ bars. For example, if we start with a long interval, say by setting $j = 0$ and $i = 1$, then $2k+1$ is 1 and the total number of bars is $2F_{2k+3} - F_{2k+1} - 1 = 2 \cdot 2 - 1 - 1 = 2$, only the bars that bound the long interval. The sequence short-long-long is forced with $j = 1$, $i = 3$, $F_{2k+1} = 2$, $F_{2k+3} = 5$ and the total number of forced bars is $2 \cdot 5 - 2 - 1 = 7$. Indeed the sequence forced by this configuration is of the form $\mid L \mid S \mid L \mid L \mid S \mid L \mid$, which has 7 bars.

We define the width of a c-interval determined by i and j to be the vertical width. If i (a point at which $c_{\mathrm{m}in}$ is set) is to the left of j, we may place

i's lattice point at the origin and run a line of slope τ through it. The point j is a lattice point above this line by the distance $d_{2k} = F_{2k+1} - F_{2k}\tau$. To improve this, we make use of form of Cassini's identity developed by Knuth (see Knuth's Concrete Mathematics):

Theorem 6.4 *For $n \geq 0$, $F_{n+1} = F_n\tau + \left(-\frac{1}{\tau}\right)^n$.*

This theorem is easily verified by induction.

Given this, and the knowledge that $2k$ is even, we now know that

$$\begin{aligned} d_{2k} &= F_{2k}\tau + \left(-\frac{1}{\tau}\right)^{2k} - F_{2k}\tau \\ &= \tau^{-2k} \end{aligned}$$

For the widths of c-intervals with $j < i$ (where the upper bound is determined before the lower), the width of the interval is

$$\begin{aligned} d_{2k+1} &= F_{2k+1}\tau - F_{2k+2} \\ &= F_{2k+1}\tau - F_{2k+1}\tau - \left(-\frac{1}{\tau}\right)^{2k+1} \\ &= \tau^{-(2k+1)} \end{aligned}$$

6.5 Extension

Theorem 6.5 *A patch $\mathcal{P}$ of contiguous tiles will not have a copy of itself as an element of its empire.*

Proof: This theorem is a direct result of Theorem 6.1. There are two cases to consider. First, suppose that $\mathcal{P}$ forced a copy of itself, $\mathcal{Q}$, oriented in the same direction. We know that $\mathcal{P}$ fixes a certain number of Ammann bars in each of the five directions. Let n be the number of bars forced in any one direction. Then in order for Q to exist, n bars must be forced in that direction by the n original bars. We know from Theorem 6.1 that this is not possible. Hence $\mathcal{P}$ cannot force $\mathcal{Q}$. Second, suppose that $\mathcal{P}$ forced a copy of itself, $\mathcal{R}$, oriented in a different direction. If the number of bars fixed by $\mathcal{P}$ is the same in each of the five directions, then we can see, for the same reasoning as above, that $\mathcal{P}$ cannot force $\mathcal{R}$. If the number of bars is different in all or some of the five directions, then let m be the maximum of the five. Then, regardless of in which direction $\mathcal{R}$ is oriented, it will be impossible for any of the original groups of fixed bars from $\mathcal{P}$ to force m bars by Theorem 6.1. Hence $\mathcal{P}$ cannot force $\mathcal{R}$. **Q.E.D.**

Chapter 7

Forcing Generalized

We would like to use information from previous chapters to be able to understand forcing patterns for initial patches of arbitrary size. This problem may be approached from several angles.

7.1 Algebraic Interpretations of Musical Sequences

We recall from the previous section that a musical sequence may be generated algebraically by differencing the sequence $\lfloor n\tau + c \rfloor$ as c runs through the integers, and then replacing the differences of 1 with an S and differences of 2 with an L. Every legal sequence may be generated by choosing $0 \leq \mathbf{c} < 1$. Therefore, the value for c uniquely determines a specific sequence. Consider the following examples of how two bars which are separated by a certain distance force a pattern of bars:

$\mid S \mid L \mid SL \mid L \mid SL \mid SL \mid L \mid SL \mid L \mid SL \mid SL \mid ...$

$\mid L \mid SL \mid SLL \mid SL \mid SLL \mid SLL \mid SL \mid SLL \mid SL \mid SLL \mid SLL \mid ...$

$\mid SL \mid SLL \mid SLLSL \mid SLL \mid SLLSL \mid SLLSL \mid SLL \mid SLLSL \mid SLL \mid SLLSL \mid SLLSL \mid ...$

$\mid SLL \mid SLLSL \mid SLLSLSLL \mid SLLSL \mid SLLSLSLL \mid SLLSLSLL \mid SLLSL \mid SLLSLSLL \mid SLLSL \mid SLLSLSLL \mid SLLSLSLL \mid ...$

In each case, we can see that an initial interval of length l forces clusters of length $l * \tau$ and $l * \tau^2$. Furthermore, if we rename each cluster of length $l * \tau$ as **S** and each cluster of length $l * \tau^2$ as **L**, then each initial interval forces the **same** sequence, namely **SLSLLSLSLL...**. It turns out that this particular sequence is a sequence as described above where $c = 0$. This fact is tremendously useful in determining forced tiles. It is given as Rule 8.

7.2 Forced Tiles

We would like to use the information which was described above to help us determine which tiles are forced by a given patch. We will first consider two identical patches which are not adjacent. Figure 7.1 gives two aces as initial patches. Notice that both aces are oriented in the same way, and also that there is a bar which passes through both aces in the same relative position in each patch. Recall from Chapter 4 that an ace does not force any tiles, either adjacent or non-adjacent. By placing two aces, however, several sets of parallel lines are fixed.

We then see that these two aces force a sequence of collinear aces in both directions. By Rule 8, we know the pattern of forcing. If equivalent points on the two aces are separated by some distance $\mathbf{d}$, then two parallel lines are separated by that distance, and so they will force a pattern of bars in a musical sequence as described above. Since one line is common to the two aces, this implies that the other four pairs of lines will force the identical arrangement of bars. This will serve to force more aces. The pattern of the forced aces will be a musical sequence as defined in Rule 8 such that $c = 0$. In fact, tiles other than the aces are forced, but we have chosen not to mark them in order to clarify the point of this argument. This result may be generalized to arbitrary patches. The message is that two identical patches of tiles which are oriented in the same direction and which share corresponding lines in one of the five directions will force copies of the patch. The pattern will correspond to that of the forced aces — collinear sequences of the patch in the pattern generated by our algebraic method where $c = 0$.

This is a general fact which can be used to account for a large portion of non-adjacent forcing in Penrose tilings. In fact, Rule 8 is enough to determine all of the forced tiles except for the ones caused by Rule 7. We can see this by examining Figures A.1 and A.2. The first empire, though incomplete, is determined completely by the two adjacent kites which are forced by the deuce. This is interesting because it removes our dependence on the Ammann lines to determine which tiles are forced. We may, in this method, merely look at pairs of identical patches and extrapolate which tiles are forced from this information, without ever needing to draw in or mark the Ammann lines which

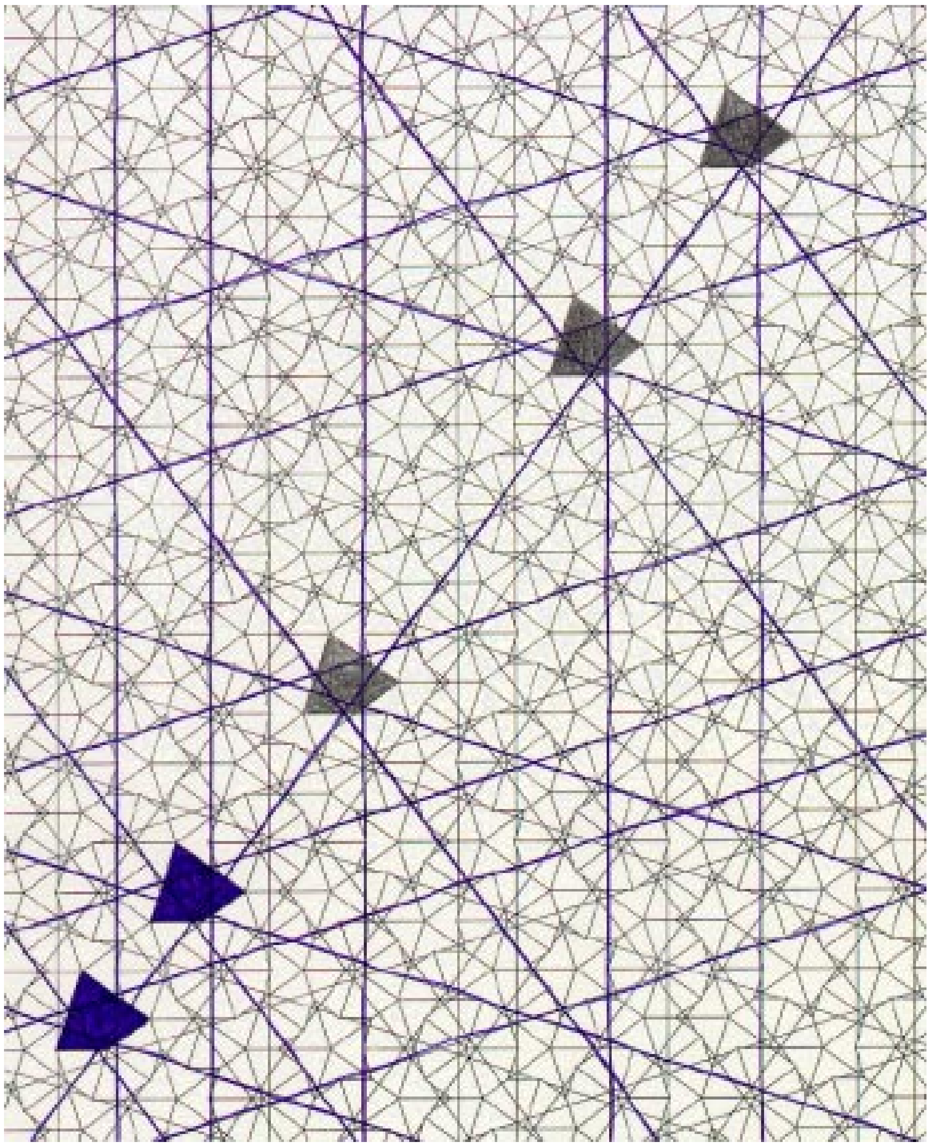


Figure 7.1: Partial forcing pattern of two aces oriented in the same direction.

motivate this knowledge.

We can see from the second figure (Figure A.2) that this does not work once Rule 7 is incorporated. It is reasonable to suspect, however, that this rule merely has the effect of removing the influence of the tiles themselves by one order. In other words, we may need to mark the Ammann lines within a certain distance of the original patch, but once the forced tiles in that region are determined, we may once again rely on the rule described above to describe the remainder of the forced tiles. This seems to be the case in practice.

7.3 Forcing Patterns

7.3.1 Skewed Patches

Figure 7.2 gives two aces which are oriented in different directions with the tiles which they force. Although an ace itself does not force any tiles, these two aces conspire to force a substantial number of tiles. We notice, however, that they do not force any additional aces - the tiles which are forced are patches of single kites (not within the area given in the figure) and clusters of two kites. It makes sense that there are forced tiles. Placing any additional tiles will fix the positions of at least one additional line, which will be parallel to an existing line, which will then go on to force more lines and possibly tiles. In some sense, we can think of these forced tiles as an interference pattern of the lines fixed by the two patches. The combination forcing is greater than the sum of its parts, and this is a property that recurs often when dealing with forcing patterns of aperiodic tilings.

7.3.2 Princess Patch

If we examine the empire of the king in Figure A.7, we notice that it includes many copies of a patch which is a sun with two darts capping non-adjacent pairs of kites. We call this patch the “princess”, in keeping with Conway’s metaphor. The princess fixes two pairs of parallel bars, so it seems reasonable that this patch would force non-adjacent tiles. However, Figure 7.3 gives a princess patch with all forced bars marked. We notice that no tiles are forced. This is a result of the fact that the intersections of forced lines do not occur close enough together to force tiles. In this case, the interference pattern of the forced bars cancels itself out.

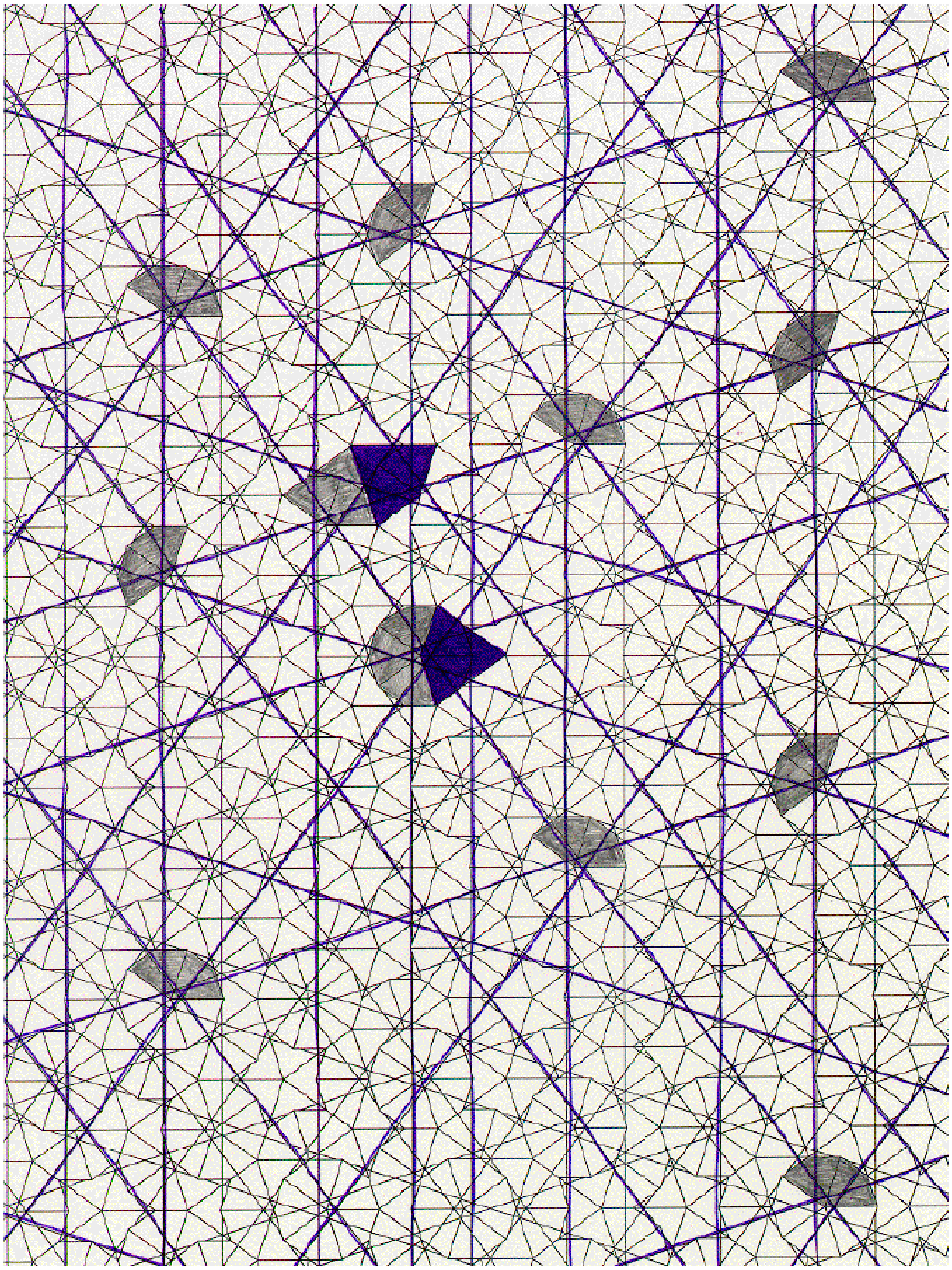


Figure 7.2: Tiles forced as a result of two aces oriented in different directions.

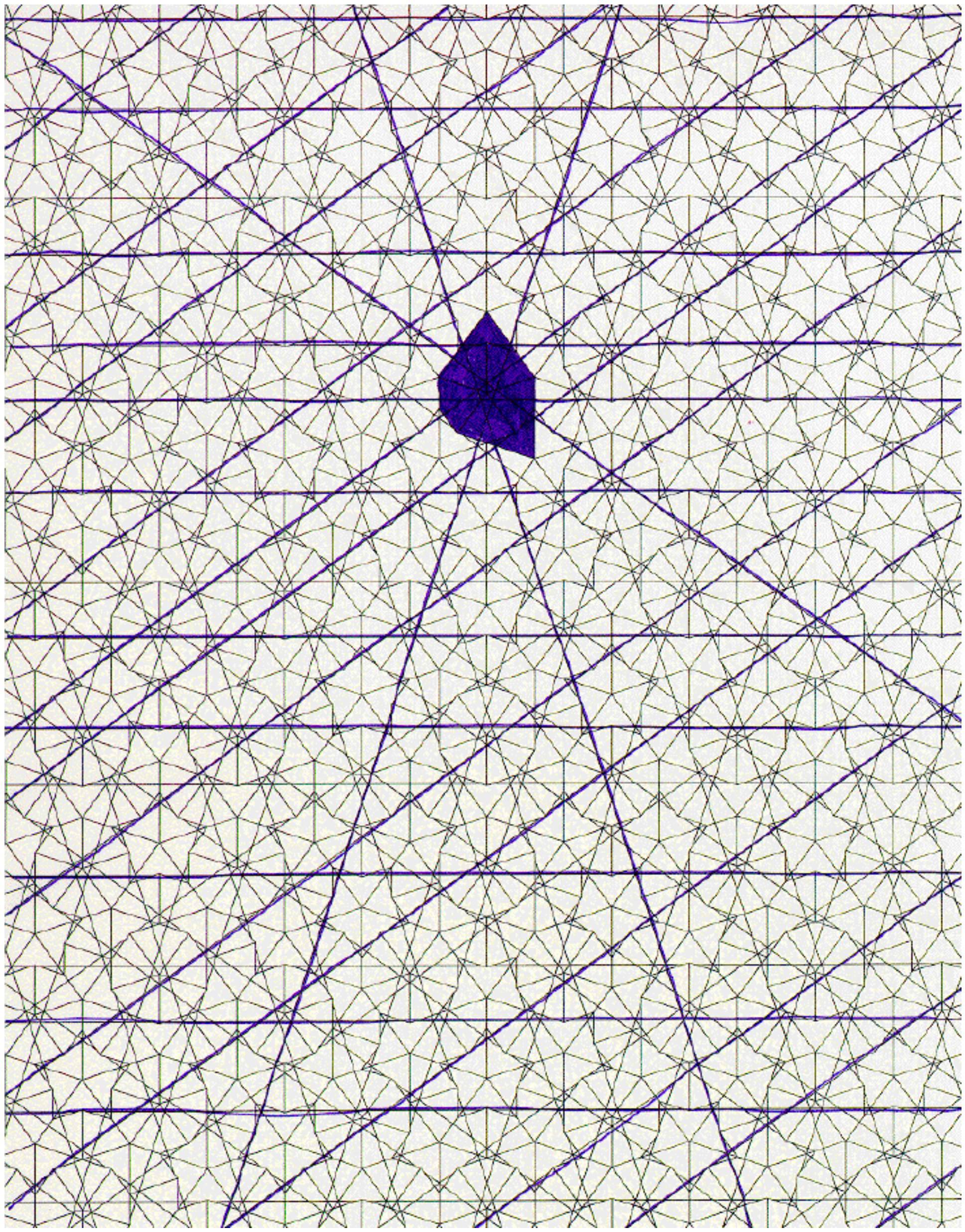


Figure 7.3: A "princess" patch of tiles: the largest contiguous patch of tiles which does not force any other tiles, either adjacent or non-adjacent.

7.3.3 Coffin and Diamond

We would like to be able to predict the tiles which are forced by initial patches of arbitrary size. At first glance, this problem seems imense. There are infinitely many inital patches, of all shapes and sizes, and any attempt to categorize them seems intractable. However, the Penrose tilings have several properties which make this task less difficult. First, and most importantly, almost every patch of kites and darts forces adjacent tiles whose rough outline is either a rhombus or a regular trapazoid with internal angles of 72° and 108°. This simplifies the task greatly. Instead of needing to determine forcing patterns of patches of all shapes, we need only describe the forcing patterns of two shapes in all sizes. If we examine the empires of the jack and king in Figures A.7 and A.4, we notice that the adjacent forced tiles in each case is a coffin shape. If we mentally superimpose the two empires, we notice that a jack is present in the king's empire in the same relative position to every ace in the jack's empire. Similary, if we compare the empires of the deuce and the queen, we notice that the adjacent tiles which are forced in each case roughly form a rhombus, or diamond. Deuces are forced by the queen in the same relative position to where kites are forced by the deuce.

This suggests that the forcing patterns of patches of similar shapes will be similar in the major aspects, regardless of size. Once the major forced patches have been placed, the remainder of the forced tiles may be determined recursively by finding the forced tiles of each of the forced patches. This process ends with the aces and kites, as they do not force any tiles. Thus, we need only a description of the major tiles forced by each of the two types of patches, and that will aid in the determination of the remainer of the forced tiles. Moreover, our analysis from Chapter 6 gives us an upper and lower bound for the sizes of patches which will be forced by initial patches of a given size.

Chapter 8

Conclusions and Future Work

This work is useful because it allows us to encode an infinite amount of information in a finite amount of space. For example, suppose that we wanted to store information about the structure of an aperiodic tiling. Since tilings themselves are infinite, and any storage mechanism must be finite, this presents a problem. However, an understanding of forcing patterns allows us to know the locations of infinitely many tiles in a tiling given some finite initial patch. Thus, incorporating information about forcing into a data structure designed to store a tiling would increase the amount of information which could be known by a great deal.

Similarly, much of this work is not limited to two-dimensional tilings of the plane. For example, we believe that the prototiles of three-dimensional tilings of space, such as the Danzer tiles, may be decorated with Ammann planes. These Ammann planes would be expected to form musical sequences in ten directions in three-dimensional space. Thus, an initial patch of tiles, if it fixed pairs of parallel planes, would force additional planes, which would in all likelihood go on to force additional tiles. The results of Chapter 7 give us an indication as to the size of patches which would be forced by an initial patch of a given radius. The analysis is the same in higher dimensions as it is for two dimensions. Unfortunately, forcing patterns in three dimensions are much more difficult to visualize. This is most likely the reason that they have not been documented.

We have increased the amount of information which may be deduced from a tiling by the presence of certain patches of tiles. Our incorporation of Rule 7 has expanded the known empires of the duece and the queen significantly. Our work in Chapter 7 has provided not only an upper and lower bound for the size of patches which may be forced, but also exact values for the possible sizes. Finally, our realization that an arbitrary initial patch will force adjacent tiles whose rough outline is either a coffin or a diamond simplifies a problem which at first glance may seem intractable. We postulate that the pattern of forcing is the same for each coffin and each diamond in the major aspects, and that the remainder of the forcing may be determined recursively using this knowledge. Thus, we have an improved understanding of how forcing occurs in aperiodic tilings.

We believe that forcing does not end with the incorporation of Rule 7. It seems likely that this is merely one step towards complete understanding. Forcing of adjacent tiles can be considered zeroeth order forcing, forcing of non-adjacent tiles via our method from Chapter 4 may be considered first order forcing, and Rule 7 may be thought to improve our knowledge to include second order forcing. It is conceivable, and indeed likely, that this is not the end of the line. More rules in which certain arrangements of the underlying tiles force bars probably exist and are yet to be discovered.

Finally, we believe that a hierarchy exists in forcing patterns. We can see from the figures in Appendix A that certain types of patches are forced by each initial patch. For the most part, coffins force coffins and diamonds force diamonds, with some shapes, such as the ace, existing in both types of empires. An investigation into this property of the tilings seems worthwhile.

Appendix A

Empires of Vertex Configurations

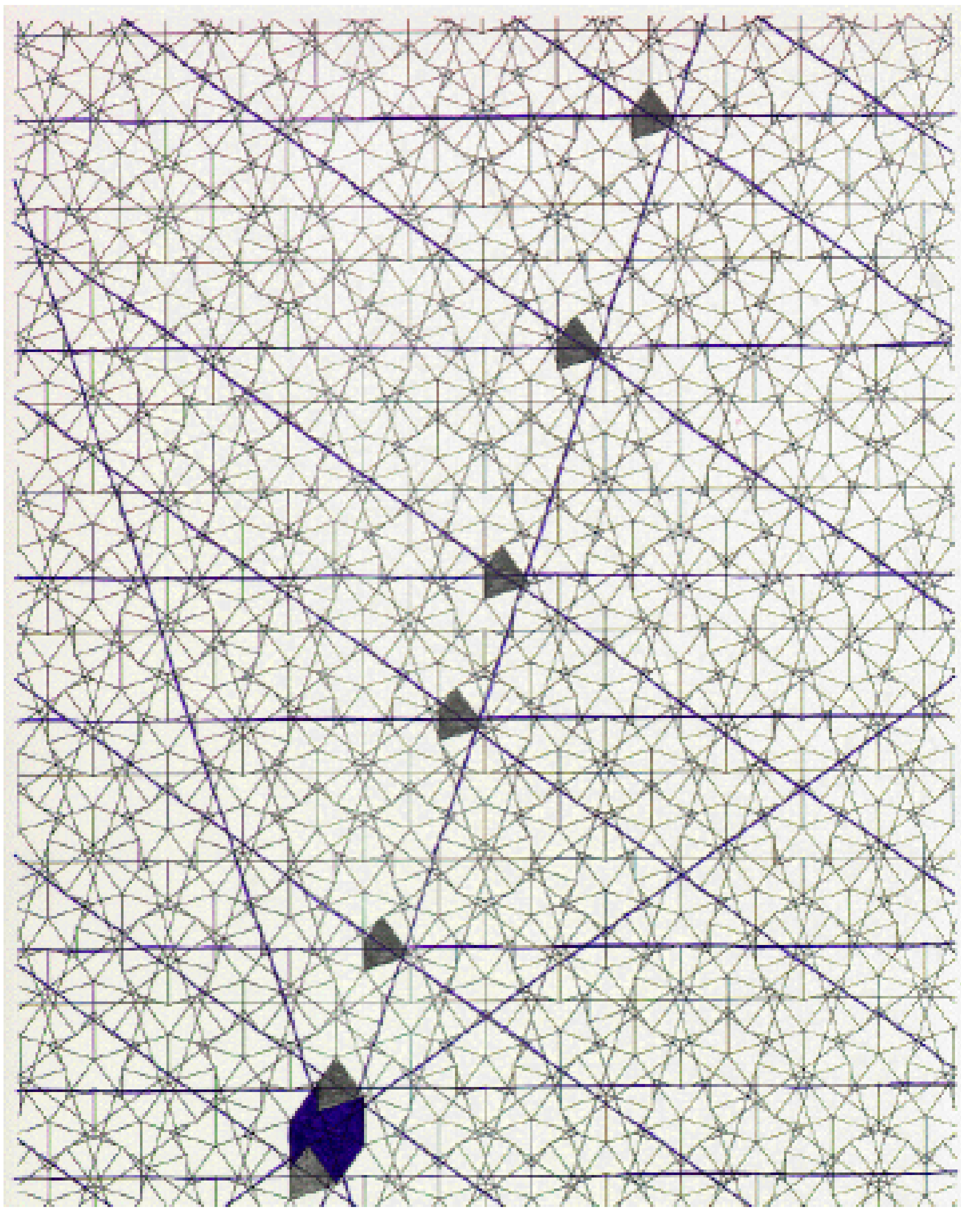


Figure A.1: Tiles forced by a deuce vertex configuration are shaded.

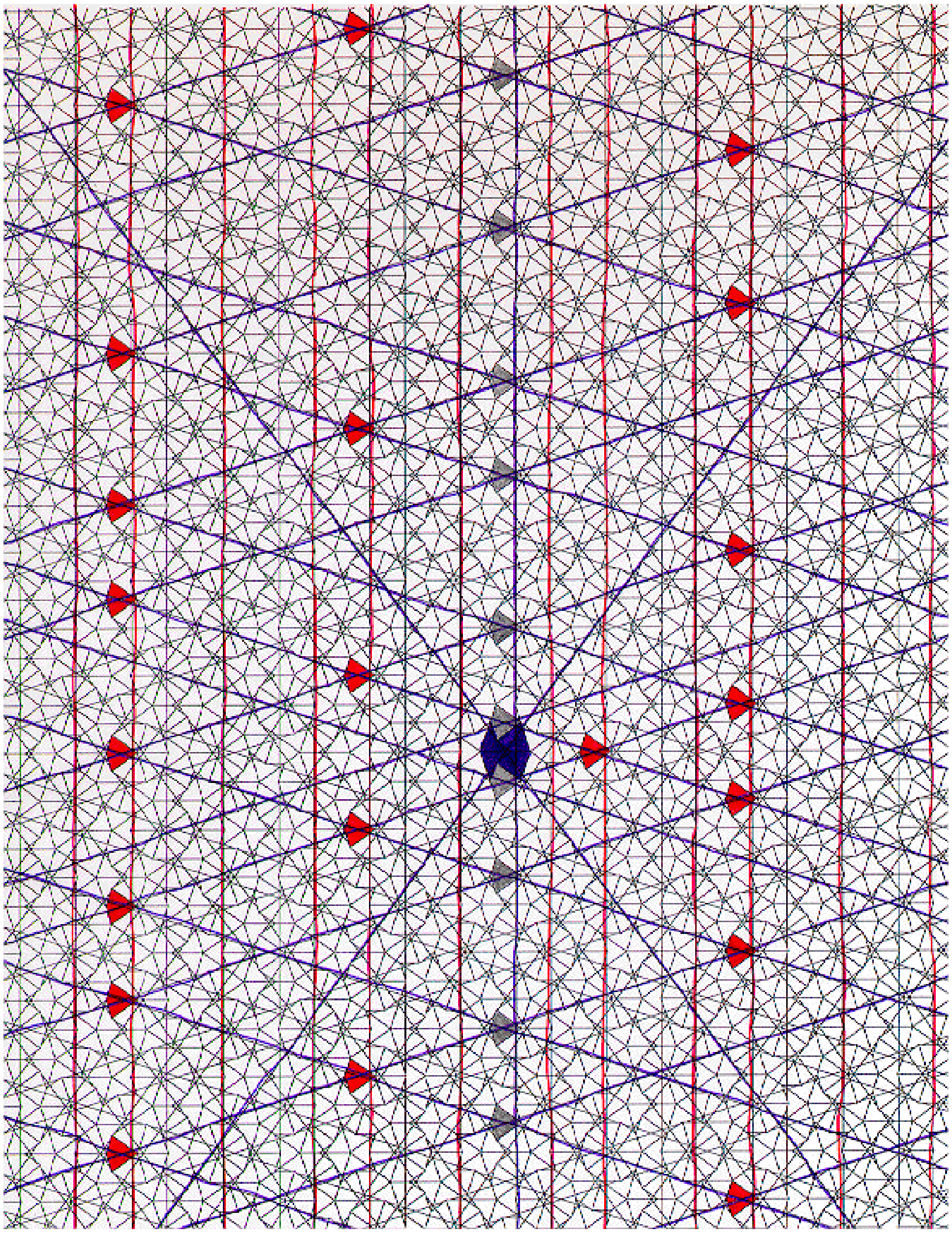


Figure A.2: A portion of the empire of the deuce vertex configuration. The red Ammann lines were determined to be forced by Rule 7. The red tiles are new discoveries; they were not included in the empire in previously published figures.

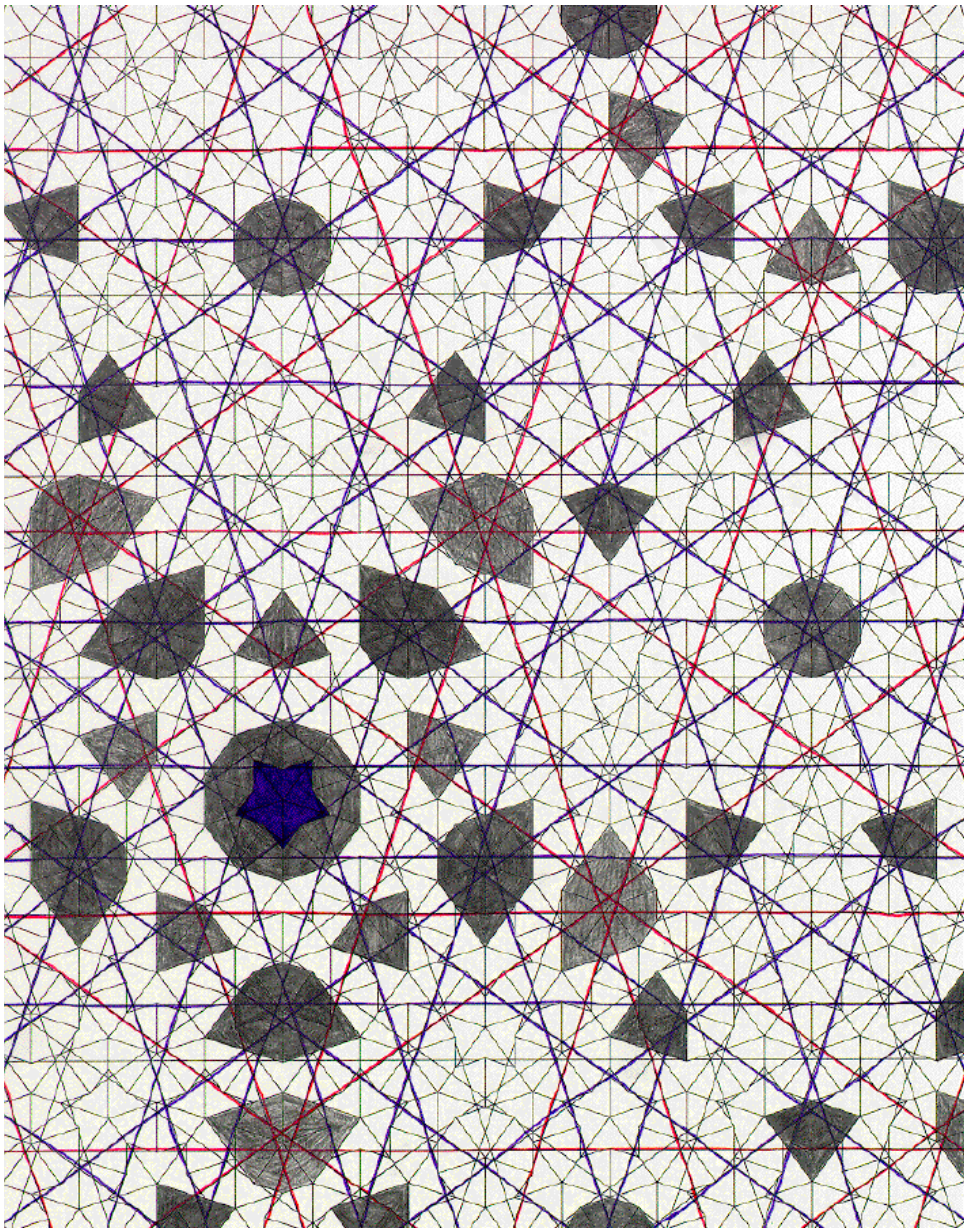


Figure A.3: A portion of the empire of the star vertex configuration. The red lines are determined via Rule 7.

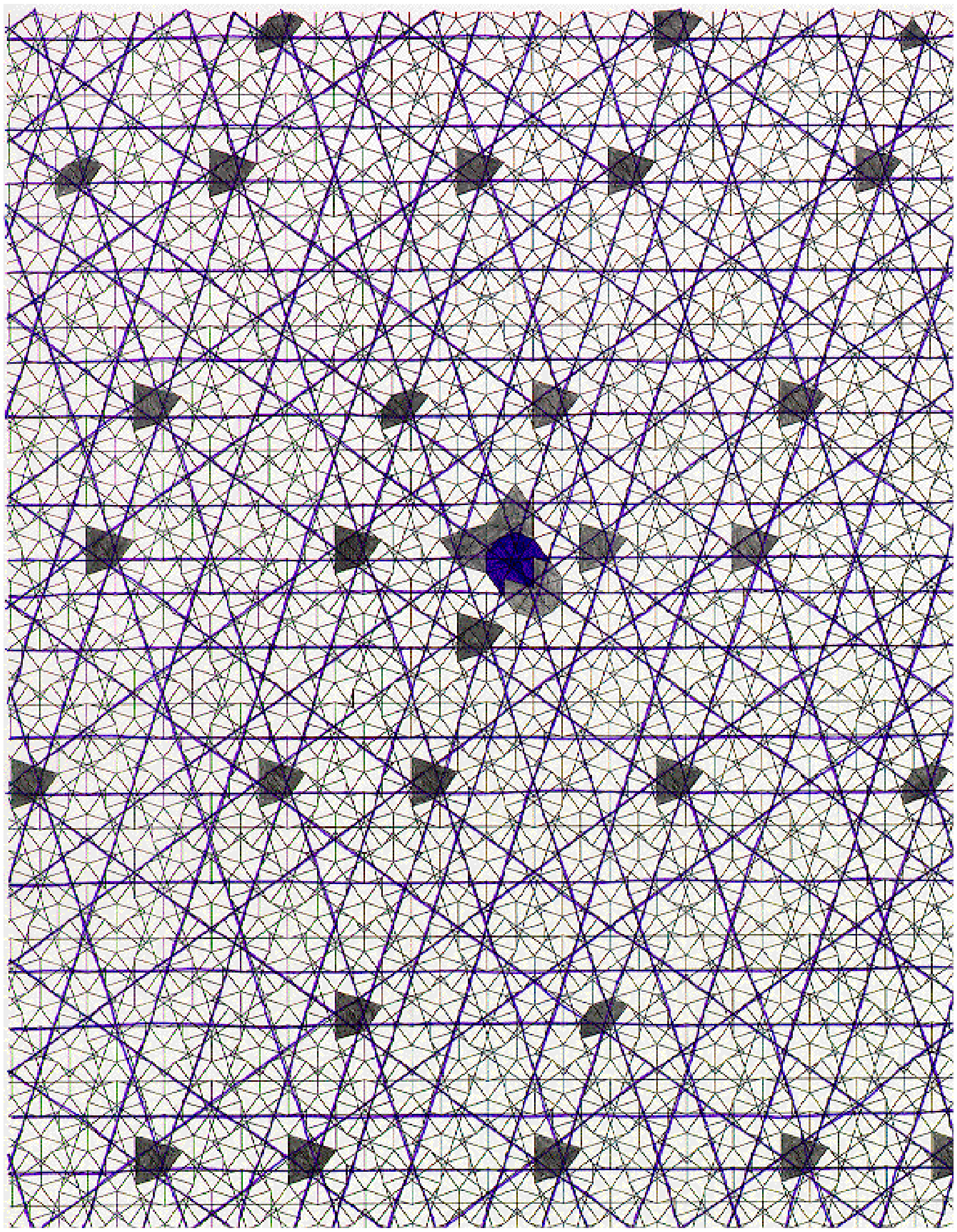


Figure A.4: A portion of the jack's empire. Shaded tiles are forced.

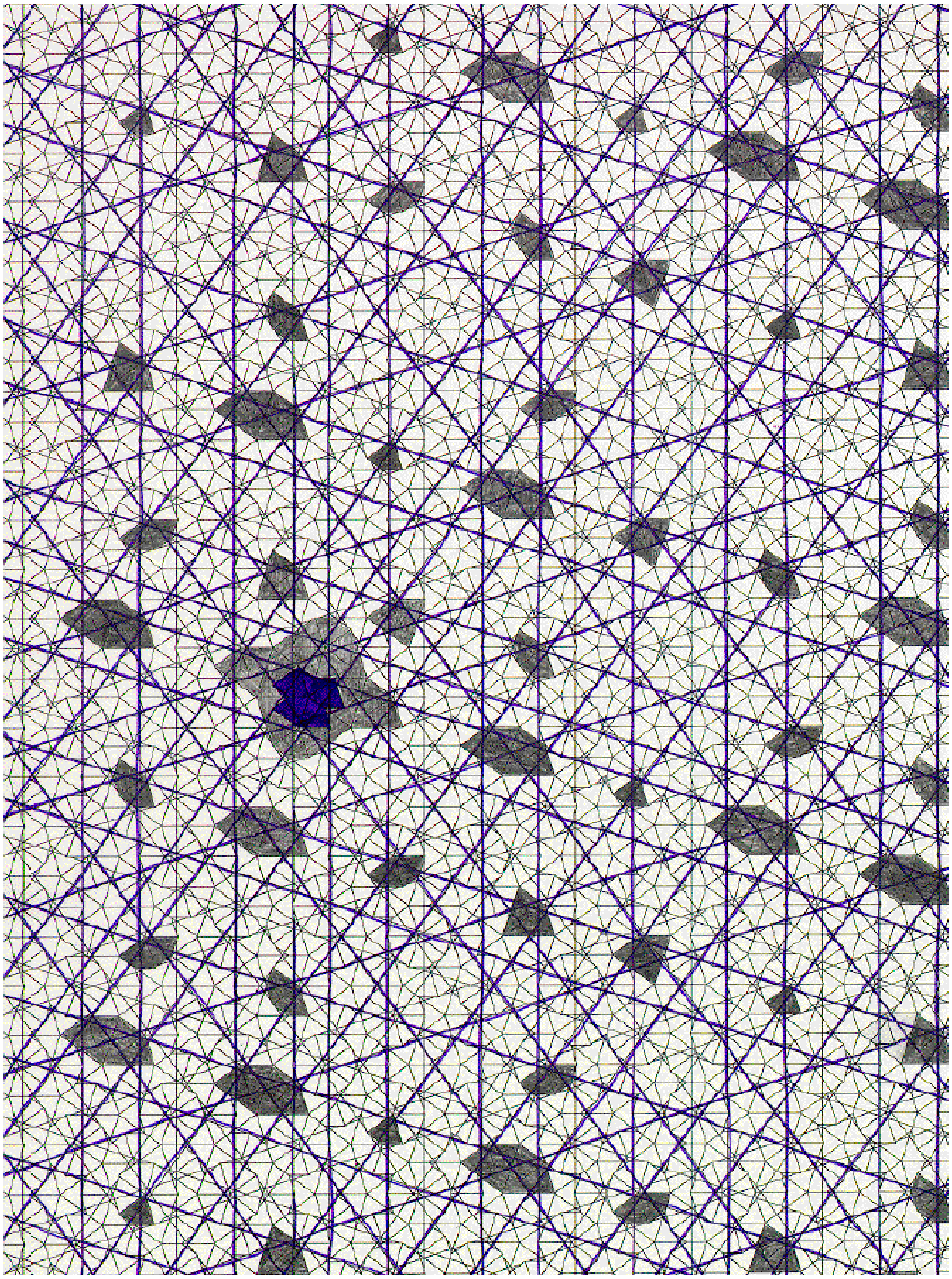


Figure A.5: Tiles forced by a queen vertex configuration.

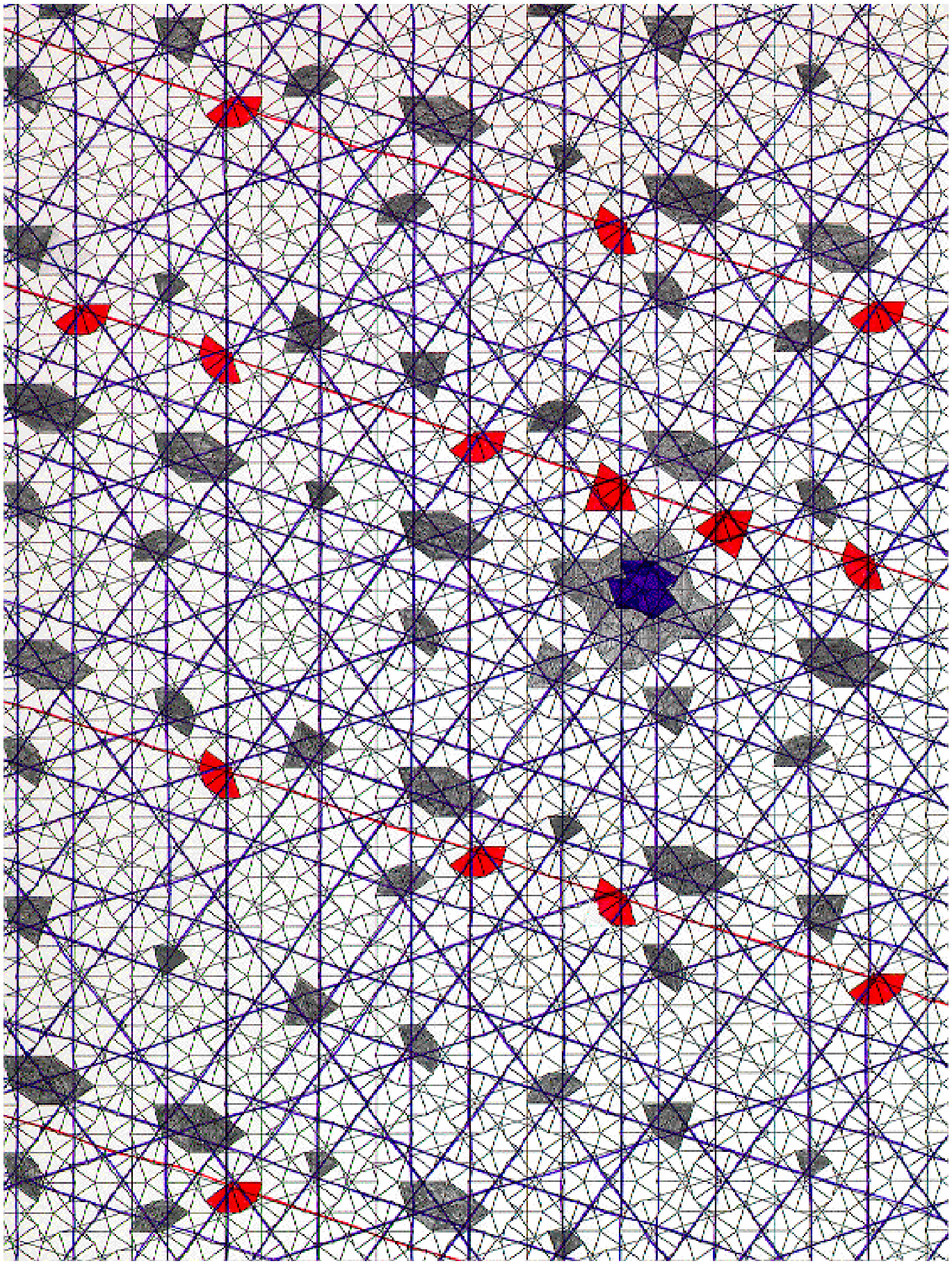


Figure A.6: Tiles forced by a queen vertex configuration. The red tiles and lines are those forced by Rule 7. The red tiles are new discoveries.

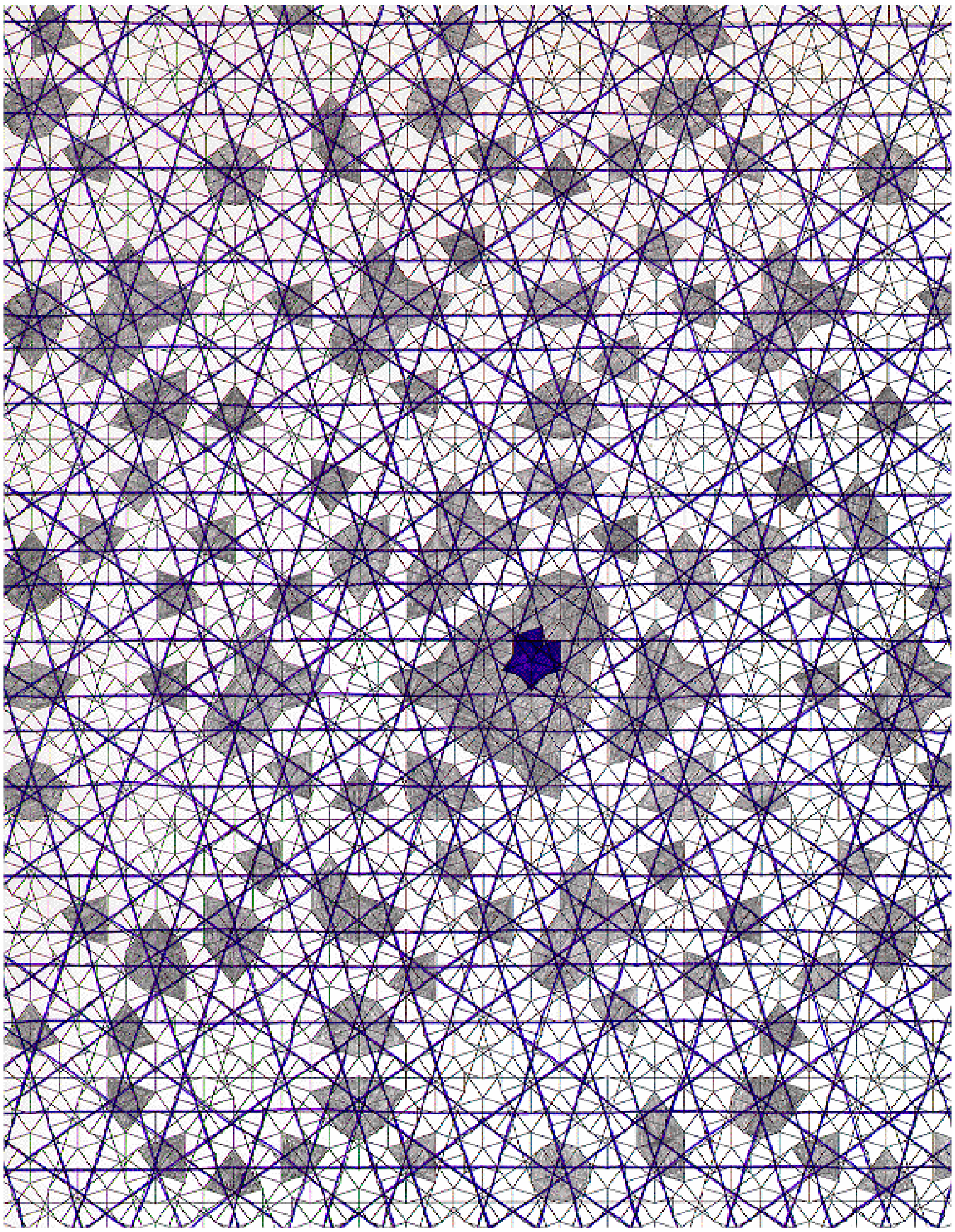


Figure A.7: A portion of the king's empire. Shaded tiles are forced.

Appendix B

Rules

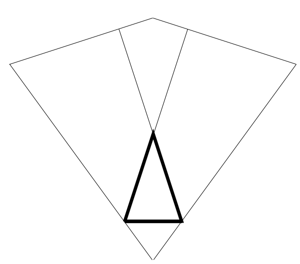
Rule 1 *Inflation of Musical Sequence*
$L \to LS$
$S \to L$

Rule 2 *Decomposition of Musical Sequence*
$LL \to S$
$S \to L$
$L \to e$

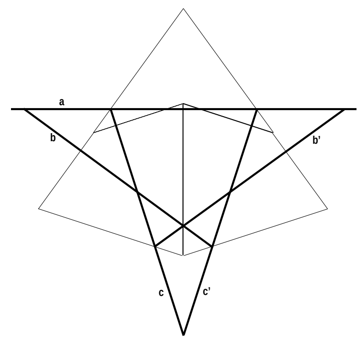
Rule 3 *A musical sequence may have no more than one consecutive S and no more than two consecutive L's*

Rule 4 *Any musical sequence may be generated algebraically by differencing the sequence $[n\tau + c]$ as n runs through the positive integers for some $0 \leq c < 1$ and substituting* **S***'s for 1's and* **L***'s for 2's.*

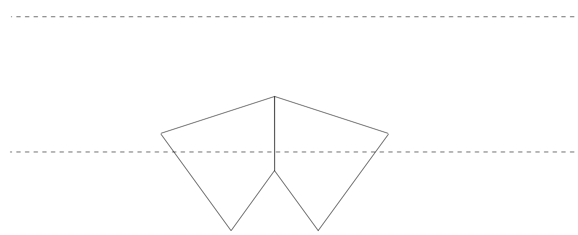
Rule 5 *A kite is forced whenever three forced bars intersect to form an isosceles triangle with angles 72°, 72°, and 36° of the size appropriate to the inflation level of the tiling as given below.*



Rule 6 *An ace is forced whenever lines a, b, b', and either c or c' as given below are forced.*



Rule 7 *Whenever two kites arranged as shown below are found in a tiling, the dashed Ammann line is forced, forming a pair of parallel bars separated by a short interval.*



Rule 8 *Two bars separated by some distance l will force a musical sequence of bars. The distance between successive bars will be either $l * \tau$ or $l * \tau^2$. The musical sequence formed will be the one generated by Rule 4 for c=0.*

Rule 9 *If $f_m \leq n < f_{m+1}$, where f_i is the ith Fibonacci number, then a sequence of n forced bars will force patches of bars of lengths f_{m-2}-1, f_{m-1}-1, or f_m-1. In addition, if n is very close to f_m, then it will only force patches of length f_{m-2}-1 or f_{m-1}-1, and if n is very close to f_{m+1}, then it will only force patches of length f_{m-1}-1 or f_m-1.*

Bibliography

[AGS92] Robert Ammann, Branko Grünbaum, and G. C. Shephard. Aperiodic tiles. *Discrete and Computational Geometry*, 8:1–25, 1992.

[BM94] S. Berman and R. V. Moody. The algebriac theory of quasicrystals with five-fold symmetries. *Journal of Physics A*, 27:115–129, 1994.

[Cha98] Benjamin C. Chaffin. Exploring the danzer tiling. Technical report, Williams College Department of Computer Science, 1998.

[Dan89] L. Danzer. Three-dimensional analogs of the planar penrose tilings and quasicrystals. *Discrete Mathematics*, 76:1–7, 1989.

[dB81] N. G. de Bruijn. Algebraic theory of Penrose's non-periodic tilings. *Nederl. Akad. Wetensch. Proc.*, 84:39–66, 1981.

[DS91] D. P. DiVincenzo and P. J. Steinhardt, editors. *Quasicrystals: The State of the Art*, volume 11 of *Directions in Condensed Matter Physics*. World Scientific, Singapore, 1991.

[GS87] Branko Grünbaum and G. C. Shephard. *Tilings and Patterns*, chapter 10. W. H. Freeman and Company, New York, 1987.

[Gum96] Petra Gummelt. Penrose tilings as coverings of congruent decagons. *Geometriae Dedicata*, 62(1):1–17, August 1996.

[JDdB89] Christian Janot, Jean-Marie Dubois, and Marc de Boissieu. Quasiperiodic structures: Another type of long-range order for condensed matter. *American Journal of Physics*, 57(11):972–987, November 1989.

[LDK94] X. Z. Li, M. Duneau, and K. H. Kuo. On the symmetries of two-colour Penrose quasi-lattices. *Europhysics Letters*, 26(8):589–594, June 1994.

[LP87] W. F. Lunnon and P. A. B. Pleasants. Quasicrystallographic tilings. *Journal de Math. Pures et Appl.*, 66(3):217–263, 1987.

[Mac82] A. L. Mackay. Crystallography and the Penrose pattern. *Physica*, 114A:609–613, 1982.

[McI93] Matthew J. McIrvin. The Fibonacci ruler. *American Journal of Physics*, 61(1):36–39, January 1993.

[Nel86] David R. Nelson. Quasicrystals. *Scientific American*, 255(2):42–51, August 1986.

[OSDS88] George Y. Onoda, Paul J. Steinhardt, David P. DiVincenzo, and Joshua E. S. Socolar. Growing perfect quasicrystals. *Physical Review Letters*, 60(25):2653–2656, June 1988.

[Sen95] Marjorie Senechal. *Quasicrystals and Geometry*. Cambridge University Press, Cambridge, 1995.

[Soc91] Joshua E.S. Socolar. Growth rules for quasicrystals. In D. P. DiVincenzo and P. J. Steinhardt, editors, *Quasicrystals: The State of the Art*, volume 11 of *Directions in Condensed Matter Physics*. World Scientific, Singapore, 1991.

[Ste86] Paul Joseph Steinhardt. Quasicrystals. *American Scientist*, 74:586–597, November–December 1986.

[Tha95] Le Tu Quoc Thang. Local rules for petagonal quasi-crystals. *Descrete and Computational Geometry*, 14(1):31–70, 1995.

[Tre96] Forrest Trepte. Exploring aperiodic tilings with inflation. Technical report, Williams College Department of Computer Science, 1996.

[vO] Geert-Jan van Opdorp. Java penrose tiler. `http://www.aie.nl/~geert/java/public/Penrose.html`.