Monomino-Domino Tatami Coverings

by

Alejandro Erickson
B.Sc., Simon Fraser University, 2007
M.Math., University of Waterloo, 2008

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

DOCTOR OF PHILOSOPHY

in the Department of Computer Science

Monomino-Domino Tatami Coverings

by

Alejandro Erickson
B.Sc., Simon Fraser University, 2007
M.Math., University of Waterloo, 2008

Supervisory Committee

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

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

Dr. Venkatesh Srinivasan, Departmental Member
(Department of Computer Science)

Dr. Jing Huang, Outside Member
(Department of Mathematics and Statistics)

Supervisory Committee

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

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

Dr. Venkatesh Srinivasan, Departmental Member
(Department of Computer Science)

Dr. Jing Huang, Outside Member
(Department of Mathematics and Statistics)

ABSTRACT

We present several new results on the combinatorial properties of a locally restricted version of monomino-domino coverings of rectilinear regions. These are *monomino-domino tatami coverings*, and the restriction is that no four tiles may meet at any point. The global structure that the tatami restriction induces has numerous implications, and provides a powerful tool for solving enumeration problems on tatami coverings. Among these we address the enumeration of coverings of rectangles, with various parameters, and we develop algorithms for exhaustive generation of coverings, in constant amortised time per covering. We also consider computational complexity on two fronts; firstly, the structure shows that the space required to store a covering of the rectangle is linear in its longest dimension, and secondly, it is NP-complete to decide whether an arbitrary polyomino can be tatami-covered only with dominoes.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

My family and friends for all of your Love.

Prof. Frank Ruskey for the support, encouragement, pressure, and freedom that you provided. I am fortunate to have such a wise, reasonable, and personable advisor who does world class research and cares for the well-being of his students.

The Natural Sciences and Engineering Research Council of Canada for recommending my Vanier Canadian Graduate Scholarship (CGS) to the Vanier Selection Board. It is an encouraging endorsement of this research. I also thank NSERC for supporting me indirectly, through my advisor's grant.

The University of Victoria for funding me through my first two years, and recognising my work through scholarships and NSERC nominations, including the Vanier CGS.

The McGill–INRIA Workshop on Computational Geometry hosted at the Bellairs Research Institute in Holetown, Barbados. Some of this research was conducted there.

DEDICATION

To Marina Marshall Silva.

EPIGRAPH

Μὴ μου τοὺς κύκλους τάραττε.

– Archimedes[1]

[1]Noted in one of several different accounts of Archimedes' death.

Chapter 1

Introduction

Tatami mats are a common floor furnishing, originating in aristocratic Japan, during the Heian period (794-1185). These thick mats, once hand-made with a rice straw core and a soft, woven rush straw exterior, are now machine-produced in a variety of materials, and are available in mass-market stores. They are so integral to Japanese culture, that a standard sized mat is the unit of measurement in many architectural applications (see [18]).

Here we depart considerably from traditional layouts, but we retain two essential features. The first of these is aspect ratio; a full mat is a 1×2 domino, and a half mat is a 1×1 monomino. The second is the 17th century rule for creating auspicious arrangements: *no four mats may meet*.

Counting domino coverings is a classic area of enumerative combinatorics and theoretical computer science, but less attention has been paid to problems where the local interactions of the dominoes are restricted in some fashion. The tatami restriction is perhaps the most natural of these, and it imposes a visually appealing structure with nice combinatorial properties (see Figure 1.1). It is the subject of exercise 7.1.4.215 in Volume 4 of "The Art of Computer Programming" (see [19]), where Knuth reprints a diagram from Jinkōki, by the renowned 17th century Japanese mathematician, Mitsuyoshi Yoshida, and recently the tatami restriction has been studied in several research papers (see [1, 10–12, 14, 26]).

The *integer grid* is the set of unit grid squares arranged on the integer lattice with their corners on lattice points.

Definition 1.1 (Monomino-domino tatami covering)**.** *Let R be a subset of the integer grid. A* monomino-domino tatami covering *of R is an exact covering of R with non-overlapping 1×1 monominoes, 1×2 horizontal dominoes, and 2×1 vertical dominoes*

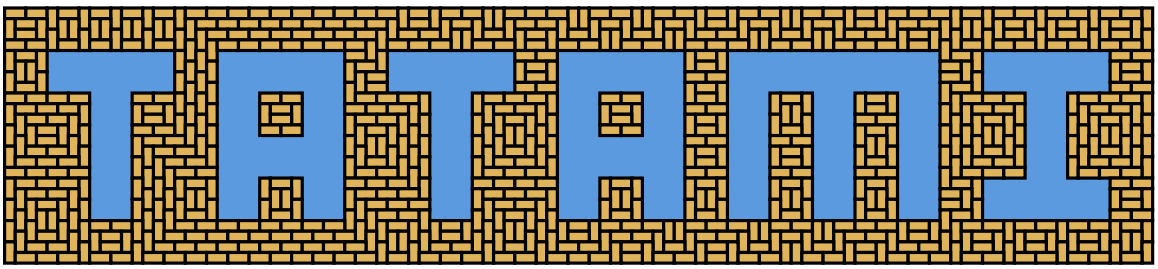


Figure 1.1: A domino tatami covering of a rectilinear region, produced by a SAT-solver.

in which no four tiles meet. The terms covering and tatami covering refer to monomino-domino tatami covering in the remainder of this dissertation, except in Chapter 6, where only domino tatami coverings are considered.

Further insight may be gained from the following graph theoretic interpretation. Let G be a graph, and let H be a subgraph of G. A matching in G *meets* H, if H contains at least one edge in the matching. A tatami covering is a matching on an induced subgraph of the infinite grid-graph, which meets all 4-cycles. In this setting, tatami coverings are a special case of *H-transverse matchings*, in which a matching of the graph, G, meets every instance of the subgraph, H, that occurs in G. Ross Churchley and Jing Huang show, in [4], that deciding whether or not G has a C_4 transverse matching is NP-complete. Consider the physical properties of a matching in which matched edges are rigid bonds between vertices, while unmatched edges are weaker bonds. Intuitively, a tatami restricted matching has some structural advantage over any non-tatami restricted matching, because there are no 4-cycles consisting only of weak bonds.

Our primary concern, however, is with the enumerative combinatorics of tatami coverings. This is the subject of Chapters 3 and 4, followed by natural extensions to combinatorial algorithms, in Chapter 5. Chapter 6 diverges from this pattern, and relates a computational complexity result similar to the one mentioned above.

1.1 Main results

The combinatorial richness of tatami coverings lies in the surprising global structure that is imposed by the tatami restriction. Tatami coverings of rectangular grids are determined by four local configurations of tiles, called *features*, which

may be rotated or reflected. The features themselves "force" the placement of other tiles, and must be arranged such that the forced tiles are not in conflicting places.

In Chapter 2 we describe the structure in detail, which was largely unknown before the publication of [12]. Prior to this, the structure of domino-only tatami coverings was resolved by Ruskey and Woodcock, in [26]. However, there was expressed interest — in the literature and by the present author — about the extension to monomino-domino coverings. Specifically, Alhazov et al. followed up on the aforementioned domino-only research with a treatment of odd-area tatami coverings which include a single monomino (see [1]). They closed with this remark:

> However, the variety of coverings with arbitrary number of monominoes is quite "wild" in [the] sense that such coverings cannot be easily decomposed, see Figure 11; therefore, most results presented here do not generalise to arbitrary number of monominoes, the techniques used here are not applicable, and it is expected that any characterisation or enumeration of them would be much more complicated.

The structure discovered by the present author (see [12]), however, reveals the opposite; the coverings with an arbitrary number of monominoes *are* easily decomposed. The decomposition has a satisfying symmetry, it is ammenable to inductive arguments, and it shows that the space complexity of a tatami covering is linear in the dimensions of the grid (see Figures 2.1 and 2.8).

The main results of this thesis are listed below. Let $T(r, c, m)$ be the number of tatami coverings of the $r \times c$ rectangle, with m monominoes.

Theorem 3.2 (Erickson, Ruskey, Schurch, Woodcock, 2011, [12])**.** *If $T(r, c, m) > 0$, then m has the same parity as rc and*

$$m \leqslant \max\{r + 1, c + 1\}.$$

Theorem 3.11. *If n and m have the same parity, and $m < n$, then $T(n, n, m) = m2^m + (m + 1)2^{m+1}$.*

This result is by Mark Schurch, but we give a new proof using the transfer matrix method (see [28]).

Theorem 3.17 (Erickson, Ruskey, Schurch, Woodcock, 2011, [12])**.**

Let $T(r,c) = \sum_{m \geqslant 0} T(r,c,m)$. *The generating function*

$$T_r(\lambda) = \sum_{c \geqslant 0} T(r,c)\lambda^c,$$

is a rational function.

In Section 3.2.1 we show that every $n \times n$ covering with n monominoes has monominoes in exactly two corners, and those corners share a boundary. Let $\mathbf{T}_n$ be those coverings with monominoes in their top corners. Let $V(n,k)$ and $H(n,k)$ be the number of coverings of $\mathbf{T}_n$ with exactly k vertical and horizontal dominoes, respectively.

Theorem 4.2 (Erickson, Ruskey, 2013, [11])**.** *The generating polynomial*

$$\mathcal{VH}_n(z) := 2 \sum_{i=1}^{\lfloor \frac{n-1}{2} \rfloor} S_{n-i-2}(z) S_{i-1}(z) z^{n-i-1} + \left(S_{\lfloor \frac{n-2}{2} \rfloor}(z) \right)^2,$$

where $S_n(z) = \prod_{i=1}^{n} (1 + z^i)$, *is equal to*

$$\sum_{k \geqslant 0} V(n,k) z^k \text{ and } \sum_{k \geqslant 0} H(n,k) z^k,$$

for even and odd n, *respectively.*

Don Knuth was the first to produce $\mathcal{VH}_n(z)$ for small n, and we have generalised his observations.

Theorem 4.6 (Erickson, Ruskey, 2013, [11])**.** *The generating polynomial* $\mathcal{VH}_n(z)$ *has the factorisation*

$$\mathcal{VH}_n(z) = P_n(z) D_n(z)$$

where $P_n(z)$ *is a polynomial and*

$$D_n(z) = \prod_{j \geqslant 1} S_{\lfloor \frac{n-2}{2^j} \rfloor}(z).$$

Three classes of tatami coverings, which are related to those we have enumerated, can be exhaustively generated in constant amortised time per covering; the number of operations required to generate all the coverings in each class is a con-

stant factor of the size of the class. Each theorem follows from the construction of a data structure and an algorithm.

Theorem 5.2. *There exists a Gray code for listing the elements of* $\mathbf{T}_n$ *such that successive coverings differ by exactly one diagonal flip. The list can be created in constant amortised time per covering.*

Theorem 5.4. *The coverings in* $\mathbf{T}_n$ *with exactly k vertical dominoes can be exhaustively generated in constant amortised time.*

A rectangular grid of height r and infinite width is called a *strip*. Tatami coverings of the height r strip — called *strip coverings* — encapsulate some of the combinatorial properties of coverings of the rectangle. Here, isomorphism refers to the topological arrangement of certain structural features, by ignoring their precise horizontal location.

Theorem 5.5. *If* $S(r, n)$ *is the number of non-isomorphic strip coverings with exactly n features, then it satisfies the system of homogeneous linear recurrence relations,*

$$\begin{aligned} V_r(n) =& 4(r-1)V_r(n-1) + 2H_r(n-1), \text{ where } V_r(0) = 1, V_r(1) = 4r-2; \\ H_r(n) =& 2V_r(n-1), \text{ where } H_r(0) = 1; \\ S(r,n) =& V_r(n) + H_r(n). \end{aligned}$$

Our final result is on the computational complexity of Domino Tatami Covering, defined below.

INSTANCE: A rectilinear region R, on the integer lattice, represented, say, as n line segments joining the corners of the polygon, which need not be simply connected.

QUESTION: Can R be covered by dominoes such that no four of them meet at any one point?

The solution makes use of `SAT`-solvers to find the gadgets used in a polynomial reduction from an NP-complete problem called planar `3SAT`.

Theorem 6.2. *Domino tatami covering is NP-complete*

1.2 Outline

Chapter 2 describes the structure of tatami coverings of rectangles, and a useful abstraction, called the T-diagram. We prove that such a covering can be recovered from the tiles on its boundary, and thereby show that the data required to store a covering is at most linear in the length of its perimeter.

Chapter 3 gives all of the new and known formulas for $T(r, c, m)$, defined in Section 1.1. In particular, we give closed form formulae for all instances where $r = c$, in Section 3.2. When $r < c$, general formulae for $T(r, c, m)$ are unknown for fixed m, besides certain cases where $T(r, c, m) = 0$, by Theorem 3.2, and $m = 0$, which is the result of [26], and $m = 1$, from [1]. We do, however, give conjectured formulae for $T(r, c, m)$ when m is maximum, via Jennifer Woodcock (private communication), in Conjecture 3.14. The values of $T(r, c)$, defined above, are given as a rational generating function. The chapter concludes with the aforementioned alternate proof of Theorem 3.17.

Chapter 4 expands on Theorem 3.6 — that is, $T(n, n, n) = n2^{n-1}$ — to derive the generating polynomial in Theorem 4.2, which counts coverings with k vertical dominoes. From this, we describe a natural partition of the $n2^{n-1}$ coverings into n classes of size 2^{n-1}, and we pursue the factorisation of $VH_n(z)$, observed for small n by Knuth. The polynomial, $P_n(z)$ mentioned above in Theorem 4.6, has several compelling properties, which are demonstrated both with proof and empirical evidence in its coefficients.

Chapter 5 develops some of the enumeration results of the previous two chapters into combinatorial algorithms. The first three of these are based on the results of Chapter 4, generating all coverings counted by $T(n, n, n)$, or the subset of these with exactly k vertical dominoes. The fourth and final combinatorial algorithm generates strip coverings.

Chapter 6 describes a polynomial reduction from planar 3-SAT to tatami domino covering (see Definition 6.1). This is a computer aided proof, where SAT-solvers were used to find gadgets in the reduction, however, the gadgets that were found can be verified by hand.

Chapter 7 summarises and discuses open problems encountered, some of which are mentioned in previous chapters.

Chapter 2

Structure of tatami coverings

The classical brick laying pattern is fundamental to tatami coverings, and it is defined precisely below. The *checkered integer grid* is the integer grid with its grid squares coloured black or white, with no adjacent squares of the same colour. Up to a permutation of the colours, the checkered integer is unique. A *checkered grid*, C, is a subset of the checkered integer grid. Consider a tatami covering of C in which every black square together with the white square to its right is covered by a horizontal domino, and the remaining squares are covered by monominoes. This is one of the four possible *running bonds*, or simply *bonds*, on C. The other three are defined as above, but with the white square above, below, or to the left of the black square. *Vertical bond* is bond with no horizontal dominoes, and *horizontal bond* is bond with no vertical dominoes.

All tatami coverings have an underlying structure which partitions the grid into regions filled with bond, and isolated monominoes. For example, in Figure 2.1 there are 10 regions, plus two isolated monominoes. The partition has some special properties which are described later in the present chapter.

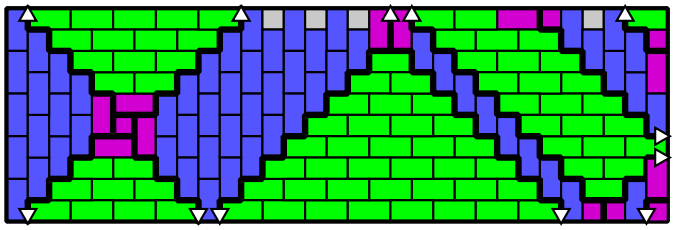
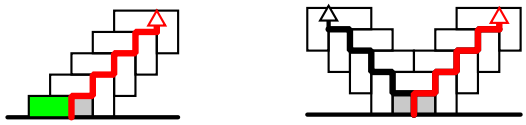


Figure 2.1: A covering showing all four types of sources. Coloured in magenta, from left to right they are, a clockwise vortex, a vertical bidimer, a loner, a vee, and two more loners.

Wherever a horizontal and vertical domino share an edge (), either the placement of another domino is forced to preserve the tatami condition, or the tiles make a T with the boundary of the grid (). In the former case, the placement of the new domino again causes the sharing of an edge (), and so on (), until the boundary is reached.

This successive placement of dominoes constructs a skinny herringbone formation, called a *ray*. Observe that once a ray is started, it propagates to the boundary. But how do they start? In a rectangular grid, we will show that a ray starts at one of four possible types of *sources*. In our discussion we use inline diagrams to depict the tiles that can cover the grid squares at the start of a ray. We need not consider the case where the innermost square (denoted by the circle in) is covered by a vertical domino () because this would simply move the start of the ray.

If it is covered by a horizontal domino (), the source, which consists of the two dominoes that share a long edge, is called a *bidimer*. Otherwise the circle is covered by a monomino () in which case we consider the grid square beside it (). If the circle in its new location is covered by a monomino, then the source is called a *vee* (); if the circle is covered by a vertical domino, then the source is called a *vortex* (); if the circle is covered by a horizontal domino, then the source is called a *loner* (). Each of these four types of sources forces at least one ray in the covering and all rays begin at either a bidimer, vee, vortex or loner. The union of a source and the rays propagating from it is called a *feature*. The different types of features are depicted in Figures 2.2-2.4, where the sources are coloured and the rays are shown in white. A bidimer or vortex may appear anywhere in a covering, as long as the coloured tiles are within its boundaries. The vees and loners, on the other hand, must appear along a boundary, as shown in Figure 2.2.



(a) (b)

Figure 2.2: *(a)*, A loner feature and, *(b)*, a vee feature, each overlaid with its feature diagram. These two types of sources must have their coloured tiles on a boundary, as shown, up to rotational symmetry.

The bold staircase-shaped arrows overlaying each ray in Figures 2.2-2.4 are called *ray diagrams*. The precise start and finish of the arrows is shown for one

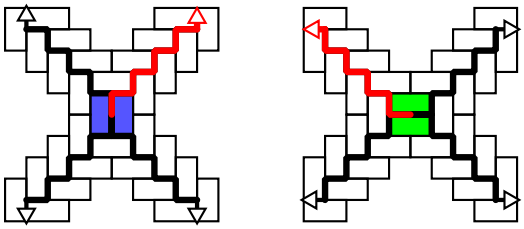


Figure 2.3: A vertical and a horizontal *bidimer* feature, each overlaid with its feature diagram. A bidimer may appear anywhere in a covering provided that the coloured tiles are within the boundaries of the grid.

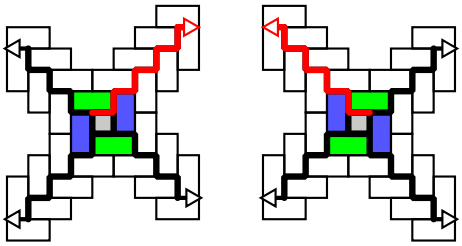


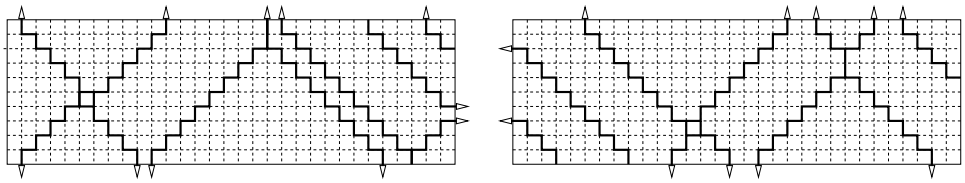
Figure 2.4: A counter clockwise and a clockwise *vortex* feature, each overlaid with its feature diagram. A vortex may appear anywhere in a covering provided that the coloured tiles are within the boundaries of the grid.

ray diagram in each depicted feature, from which the general definition can be inferred. Ray diagrams of the four possible orientations are represented by the symbols $\nearrow$, $\nwarrow$, $\searrow$, and $\swarrow$ regardless of the length of the ray diagram.

The union of ray diagrams in each of the four types of source-ray drawings in Figures 2.2-2.4 is called a *feature diagram*. Note that feature diagrams do not intersect. A collection of feature diagrams partitions the rectangle, and those parts with unit area greater than 2 are called *regions*. A collection of feature diagrams is called a *T-diagram* (see Figure 2.5) if there is a bijective correspondence between the features of a tatami covering and the diagrams of the collection. If a collection of feature diagrams is a T-diagram, then its regions can be covered with vertical and horizontal bond to obtain a tatami covering.

A ray diagram may meet more than two regions of a collection of feature diagrams, so we say that a ray diagram *bounds* (at most) the two regions of bond containing the tiles of the ray it represents. Specifically, one region contains the vertical tiles and the other region contains the horizontal tiles, provided the regions exist.

If a ray diagram bounds a region, then its ray determines the position of at least one domino in that region. The bond in this region is therefore determined



(a) (b)

Figure 2.5: (a) The T-diagram of Figure 2.1. (b) A collection of feature diagrams that is not a T-diagram.

uniquely, and must agree with that of the other ray diagrams bounding the region. Rays bounding the same region are *adjacent*. Lemma 2.1 gives conditions for a collection of feature diagrams to be a T-diagram; conditions that can each be checked in a constant number of operations.

Consider a region, R, with a ray diagram, D, that bounds it. The orientation of any bond covering R is completely determined by D (see Figure 2.6). If all of the ray diagrams that bound D determine the same orientation of bond inside R, then we say the region R is consistent.

A *path* is a sequence of grid squares, with consecutive squares connected at edges, and no repeated squares. The *length of a path* is the number of grid squares it contains. Let p be a path that is entirely contained inside a region. If an edge of the first square of p borders a ray diagram, and an edge of the last square borders another ray diagram, then p *connects* these ray diagrams. Given two ray diagrams, the length of every path connecting them has the same parity, which can be checked in constant time by looking at the colour of the grid squares on the checkered grid using the first and last squares.

Lemma 2.1. *A collection of feature diagrams is a T-diagram if and only if the following conditions are satisfied.*

(a) Every region is consistent (see Figure 2.6).

(b) The lengths of paths connecting adjacent ray diagrams must satisfy the parity requirements tabulated below for each orientation of ray diagram (note that "×" entries in the table are impossible by (a), and see Figure 2.7).

vertical bond

	↗	↖	↙
↖	*odd*	×	×
↙	*even*	*even*	×
↘	*even*	*even*	*odd*

horizontal bond

	↗	↖	↙
↖	*even*	×	×
↙	*even*	*odd*	×
↘	*odd*	*even*	*even*

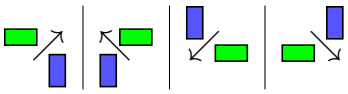


Figure 2.6: The spacial relationships between different types of ray diagrams and orientations of bond (see also Figure 2.7).

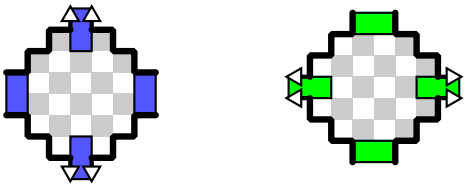


Figure 2.7: Example regions of vertical and horizontal bond, respectively.

Proof. Let R be a rectangular grid and let $\mathscr{F}$ be a collection of feature diagrams in R.

Suppose (a) and (b) are satisfied for each region of $\mathscr{F}$, and consider a checkered region, R, that must be covered with, say, vertical bond. Let α and β be ray diagrams that bound R, which are connected by p. The colour of the squares bordering α uniquely determines the colour of the squares bordering β (and vice versa) by the parity of the length of p. Together with the fact that p satisfies (b), we ensure that the same type of vertical bond meets both α and β in a way that does not conflict with the tiles of the rays they represent.

Thus every region of $\mathscr{F}$ can be covered with bond that is not in conflict with the tiles belonging to the features represented by $\mathscr{F}$. The tatami condition is satisfied in the bond(s) and in the features (where regions of bond meet), therefore $\mathscr{F}$ is a T-diagram, and this proves sufficiency.

A tatami covering of a T-diagram must cover its regions with bond because there can be no interior monominoes — these would create vortices not accounted for by the T-diagram — or vertical dominoes touching horizontal dominoes — these would create rays not accounted for by the T-diagram. Therefore (a) and (b) must be satisfied. □

2.1 Storage complexity

The characterisation of Lemma 2.1 has some implications for the space complexity of a covering.

Lemma 2.2. *Let G be an $r \times c$ grid, with $r < c$.*

(i) A tatami covering of G is uniquely determined by the tiles on its boundary.

(ii) The storage requirement for a tatami covering of G is $O(c)$; that is, a tatami covering can be recovered from $O(c)$ bits.

(iii) Whether a collection of feature diagrams in G is a T-diagram can be determined in time $O(c)$.

Proof. To prove (i), we need to show that we can recover the T-diagram from the tiles that touch the boundary. Those portions of the T-diagram corresponding to vees and loners, as well as bidimers whose source tiles are both on the boundary (⊔), are easy to recover. The black ray diagrams in Figure 2.8 show their recovery. Imagine filling in the remaining red ray diagrams, whose ends look like ⊏, by following them naïvely, backwards from their endings to the boundary. The ends of the four ray diagrams emanating from a bidimer or vortex will always form exactly one of the four patterns illustrated in Figure 2.9; in each case, it is straightforward to recover the position and type of source. This proves (ii).

Part (ii) follows from (i), because we can use a ternary encoding for the perimeter squares.

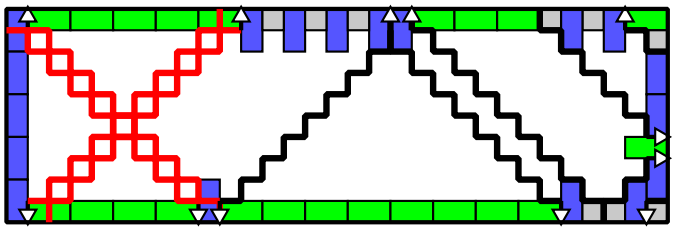
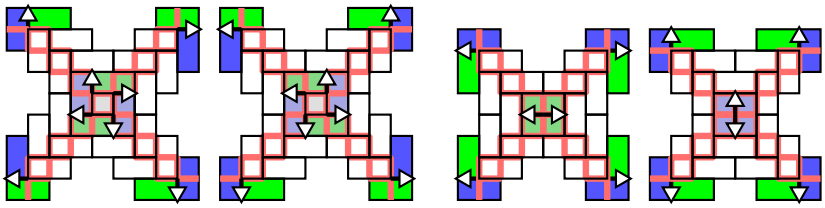


Figure 2.8: The same covering as in Figure 2.1 with only the boundary tiles showing. Ray diagrams emanating from sources on the boundary are in black and otherwise, they are drawn naïvely in red, to be matched with a candidate source from Figure 2.9.

Claim (iii) is true provided that Lemma 2.1 only needs to be applied to $O(c)$ pairs of adjacent ray diagrams. Each ray diagram bounds exactly two regions, each of which is bounded by at most three other ray diagrams. Thus, a ray diagram is adjacent to at most six others. Let $G = (V, E)$ be a graph whose vertex set corresponds to the set of ray diagrams. There is an edge between a pair of vertices whenever the corresponding ray diagrams are adjacent. The maximum degree of G can never be more than 6 (the highest achievable degree is actually



(a) Clockwise and counterclockwise vortices.

(b) Horizontal and vertical bidimers.

Figure 2.9: The four types of vortices and bidimers are recoverable from the ends of their ray diagrams, at the boundary of the grid. Extending the ray diagrams naïvely, backwards from the boundary, we form one of the two patterns in the red overlay. One occurs only for bidimers and the other for vortices. Successively placing tiles, working from the ends of the rays towards the central configuration, we also find the orientation of the source, as shown in the figure.

4, as in Figure 2.1), so the number of adjacencies, $|E|$, and hence applications of Lemma 2.1, is linear in the number of rays, $|V|$, which is at most four times the number of features, which is in $O(c)$. This proves (iii). □

Chapter 3

Enumerating tatami coverings with m monominoes

Let $T(r, c, m)$ be the number of tatami coverings of a rectangular grid with r rows, c columns, and m monominoes. Also, $T(r, c)$ will denote the sum

$$T(r, c) = \sum_{m \geqslant 0} T(r, c, m).$$

The enumeration of tatami coverings of rectangles, with respect to r, c, and m is complete for square grids, and partially complete for maximum m. This chapter contains all of these results. Values of $T(r, c, m)$ for all grids up to 14×14 are given in Tables A.1–A.18. Table A.17 is $T(r, c)$, the aggregate of these, and Tables A.18 and A.19 count coverings of the square, and coverings with maximum monominoes, respectively.

The following definition is used throughout this dissertation.

Definition 3.1 (Diagonal)**.** *Let T be a tatami covering of the $r \times c$ grid. A* diagonal*, D, of T is a contiguous sequence of like-aligned dominoes whose centers lie on a line with slope ± 1. The sequence must begin with a domino with its long edge on the boundary; the final domino touches an adjacent boundary and shares an edge with a monomino, which is also considered to be part of the diagonal (see Figures 3.2(a) and 3.1).*

A diagonal flip *of D consists of removing it from T, reflecting horizontally, rotating by $\frac{\pi}{2}$ radians, and placing it back onto the grid squares that were vacated.*

There are three things to note about the diagonal flip:

- a flipped diagonal is a diagonal;

- the operation preserves the tatami restriction; and,
- it changes the orientation of the dominoes that it contains, and maps the monomino to the other extreme of the diagonal.

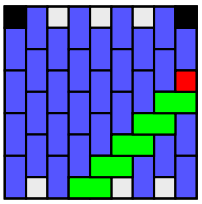


Figure 3.1: A diagonal flip in a 9×9 vertical bond.

3.1 Maximum number of monominoes

We now give necessary conditions for $T(r, c, m)$ to be non-zero.

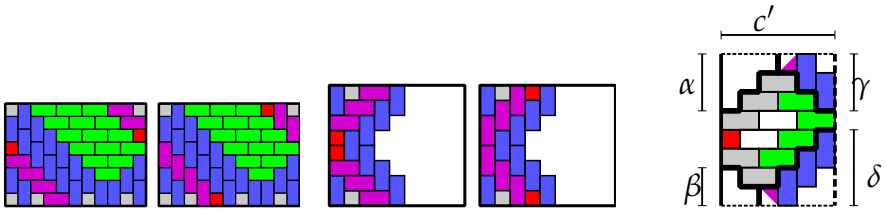
Theorem 3.2. *If $T(r, c, m) > 0$, then m has the same parity as rc and*

$$m \leqslant \max(r + 1, c + 1).$$

Proof. Let r, c and m be such that $T(r, c, m) > 0$ and let d be the number of grid squares covered by dominoes in an $r \times c$ tatami covering so that $m = rc - d$. Since d is even, m must have the same parity as rc.

We assume that $r \leqslant c$, and prove that $m \leqslant c + 1$. The proof proceeds in two steps. First, we will show that a monomino on a vertical boundary of any covering can be *mapped* to the top or bottom, without altering the position of any other monomino. Then we can restrict our attention to coverings where all monominoes appear on the top or bottom boundaries, or in the interior. Secondly, we will show that there can be at most $c + 1$ monominoes on the combined horizontal boundaries.

Let T be a tatami covering of the $r \times c$ grid with a monomino, μ, on the left boundary, touching neither the bottom nor the top boundary. The monomino μ is (a), part of a vee or a loner, or is (b), surrounded by horizontal dominoes. If μ is part of a diagonal, it can be mapped to the top or bottom boundary via a diagonal flip. In case (a) a diagonal clearly exists since it is a source and its ray will hit a horizontal boundary because $r \leqslant c$.



(a) A diagonal flip. (b) The case for vees. (c)

Figure 3.2: *(c)*, If both diagonals are blocked, then $c < r$. The covering is at least this tall and at most this wide.

If μ is surrounded by horizontal dominoes, then we argue by contradiction. Suppose neither the upward, nor the downward diagonal exists, then they must each be impeded by a distinct ray. Such rays have this horizontal region to the left so the upper one is directed SE and the lower NE and they meet the right boundary (before intersecting). Referring to Figure 3.2(c),

$$\begin{aligned}\alpha + \beta + j =& \gamma + \delta + 1 \leqslant r \\ &\leqslant c \leqslant c' = \alpha + \gamma = \beta + \delta,\end{aligned}$$

where j is some odd number. Thus $\alpha + \beta + j \leqslant \alpha + \gamma$ implying that $\beta < \gamma$. On the other hand,

$$\gamma + \delta + 1 = r \leqslant c \leqslant c' = \beta + \delta$$

implies that $\gamma < \beta$, which is a contradiction. Therefore at least one of the diagonals exists and the monomino can be mapped to a horizontal boundary via a diagonal flip.

We may now assume that there are no monominoes strictly on the vertical boundaries of the covering, and therefore all monominoes are either in the top or bottom rows or in vortices. Let v be the number of vortices. Encode the bottom and top rows of the covering by length c binary sequences Q and P, respectively. In the sequences, 1s represent monominoes and 0s represent squares covered by dominoes.

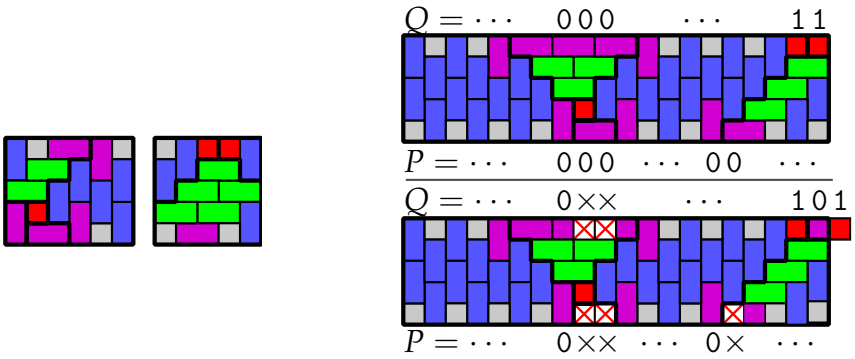
Summarizing the argument below, we first find a bijective correspondence between occurrences of 11 in Q and 00 in P, and vice versa. The 11s are replaced by 101 and 00s are replaced by 0. Second, we find v occurrences of 000, where v is the number of vortices, in each (updated) sequence. Replace each 000 with 00. If v is the number of vortices, then the total length of the updated sequences is $2c - 4v$, and total number of 1s is at most $c - 2v + 1$. Adding the monominoes in

the vortices gives the desired upper bound of $c - v + 1 \leqslant c + 1$.

A 11 in Q is a vee in the top row; the vee has a region of horizontal dominoes directly below it. This region of horizontal bond must reach the bottom row somewhere, otherwise, by an argument similar to one given previously, we would have $c < r$ (see Figure 3.3(a)). Therefore, there must be a 00 in P unique to these 1s in Q. One of these 0s is used to separate the 1s (see Figure 3.3(b)). The updated sequences contain no 11, but the total number of 1s remains unchanged.

Each vortex generates rays which reach the top and bottom boundaries, since $r \leqslant c$, and the dominoes on either side of the rays induce a 000 in P and another 000 in Q (see Figure 3.3(a)). (Although not used in this proof, note that the comments above also apply to bidimers.) Removing a 00 from each triple yields a pair of sequences whose combined length is $2c - 4v$, neither of which contains a 11 (see Figure 3.3(b)). Thus the total number of 1s is at most $\lceil |P|/2 \rceil + \lceil |Q|/2 \rceil$, which is at most $c - 2v + 1$. Adding back the v vortex monominoes, we conclude that there are at most $c - v + 1$ monominoes in total, which finishes the proof.

Note that, to acheive the bound of $c + 1$, we must have $v = 0$, and that the maximum is achieved by a vertical bond. □



(a) (b)

Figure 3.3: Each vortex and vee is associated with segments of monomino-free grid squares shown in purple. (a) Segments associated with vortices have length at least three. Those associated with vees have at least two 0s. (b) The two types of updates to sequences P and Q. The upper sequences are before the updates and the lower are after updates. The symbol $\times$ represents a deletion from the sequence.

The converse of Theorem 3.2 is false; for example, Alhazov et al. (see [1]) show that $T(9, 13, 1) = 0$. We now state a couple of consequences of Theorem 3.2.

Corollary 3.3. *The following three statements are true for tatami coverings of an $r \times c$ grid with $r \leqslant c$.*

(i) The maximum possible number of monominoes is $c + 1$ if r is even and c is odd; otherwise it is c. There is a tatami covering achieving this maximum.

(ii) A tatami covering with the maximum number of monominoes has no vortices.

(iii) A tatami covering with the maximum number of monominoes has no bidimers.

Proof. (i) That this is the correct maximum value can be inferred from Theorem 3.2. A covering consisting only of vertical bond achieves it, for example.

(ii) This was noted at the end of the proof of Theorem 3.2.

(iii) We can again use the same sort of reasoning that was used for vortices in Theorem 3.2, but there is no need to "add back" the monominoes, since a bidimer does not contain one. □

3.2 Tatami coverings of square grids

We have closed form formulae for all of $T(n, n, m)$, which are stated below.

$$T(n,n,m) = \begin{cases} m2^m + (m+1)2^{m+1} & \text{if } m < n \text{ and } 2|n^2 - m \quad (3.1) \\ n2^{n-1} & \text{if } n = m \quad (3.2) \\ 0 & \text{otherwise.} \quad (3.3) \end{cases}$$

Equation (3.3) was shown in Theorem 3.2, and the other two equations are results of the present chapter. Theorem 3.6 gives Equation (3.2) and Theorem 3.11 gives Equation (3.1).

3.2.1 Tatami coverings of the square, with maximum monominoes

In this section, we show by induction that $T(n, n, n) = n2^{n-1}$. The result is re-derived later, from Theorem 4.2, to exhibit a natural partition of coverings into n classes of size 2^{n-1}.

The following lemma and corollary provide a convenient characterisation of $n \times n$ tatami coverings with n monominoes, in terms of diagonal flips (see Definition 3.1).

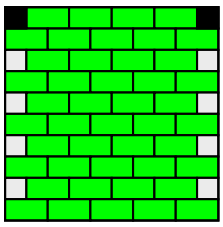


Figure 3.4: A horizontal bond for $n = 10$.

Lemma 3.4 (Erickson, Ruskey, Schurch, Woodcock, 2011, [12])**.** *For each $n \times n$ covering with n monominoes, a bond can be obtained via a finite sequence of diagonal flips in which each monomino moves at most once. The original covering is obtained by applying the same sequence of diagonal flips to this bond, in reverse order.*

Proof. There are several proofs of this, but the following idea in [12], by Jennifer Woodcock, is particularly succinct.

Let T be the T-diagram of an $n \times n$ covering with n monominoes, and let $A(T)$ be the area of a maximal connected bond which contains the centre of T, called the *central bond* (see Figure 3.5). We may assume that T has at least one feature diagram, and by Corollary 3.3 it is either a loner or a vee. Let ρ be a ray diagram that passes nearest to the centre of T. This ray diagram is the boundary between the dominoes of the central bond, and a diagonal with dominoes of the other orientation. Flipping the diagonal yields a T-diagram, T' with $A(T') > A(T)$, and its monomino becomes part of the central bond. Repeating this process we obtain a bond on the $n \times n$ grid using a finite number of diagonal flips, and no monomino is moved twice (see Figure 3.6). □

Corollary 3.5 (Erickson, Ruskey, Schurch, Woodcock, 2011, [12])**.** *Every $n \times n$ covering with n monominoes has exactly two corner monominoes and they are in adjacent corners.*

Proof. The bond contains such corner monominoes, and in the proof of Lemma 3.4 no corner monomino is moved because the containing diagonal is part of the central bond. Since a bond on the $n \times n$ grid with n monominoes has two monominoes in adjacent corners, so must every other $n \times n$ covering with n monominoes. □

Corollary 3.5 shows that the four rotations by $\pi/2$ radians of any $n \times n$ covering with n monominoes are distinct.

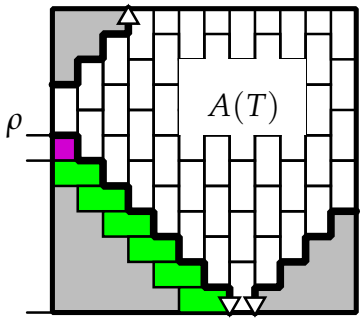


Figure 3.5: In the covering T from Lemma 3.4, the ray diagram ρ separates the central bond from a diagonal, shown with green dominoes. Flipping the diagonal adds to the central bond, which guarantees a finite number of flips and protects the diagonal's monomino from being moved again.

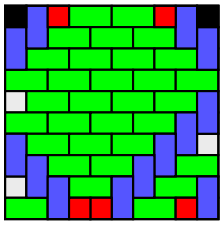


Figure 3.6: A sequence of 5 diagonal flips, shown in blue, beginning with a bond, results in this covering. Flipped monominoes are coloured red.

Theorem 3.6 (Erickson, Ruskey, Schurch, Woodcock, 2011, [12])**.** *The number of $n \times n$ coverings with n monominoes, $T(n,n,n)$, is $n2^{n-1}$.*

Proof. Let $\mathbf{T}_n$ be the tatami coverings counted by $T(n,n,n)$ which have monominoes in their top corners, and let $s(n) = T(n,n,n)/4$. We will show that

$$s(n) = 2^{n-2} + 4s(n-2), \text{ where } s(1) = \frac{1}{4} \text{ and } s(2) = 1. \tag{3.4}$$

The solution to (3.4) is $s(n) = n2^{n-3}$.

If n is even, then the bond in $\mathbf{T}_n$ is a horizontal bond, with monominoes occurring on its left and right boundaries. Coverings are obtained by flipping diagonals containing these monominoes. Each monomino besides those in the top corners, is contained in exactly two diagonals. Lemma 3.4 yields the following refinement of our vocabulary. We say a *monomino (or a diagonal that contains it) is flipped* if it is moved from its original position in the bond. The containing diagonal can be specified with a direction — up or down, when n is even (see Figure 3.7).

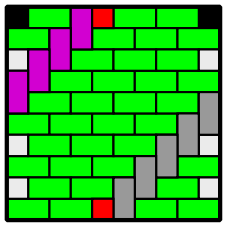


Figure 3.7: The magenta diagonal contains 5 tiles and it is flipped up. The grey diagonal contains 6 tiles and it is flipped down.

Call the lower leftmost monomino w, and the lower rightmost monomino e. If either of these is flipped up, then there remain exactly $n-3$ diagonals that can be flipped, and additionally, the flips can be made independently (see Figure 3.8). This accounts for the 2^{n-2}, in Equation (3.4). Each of the remaining four cases, where neither e nor w is flipped up, is described by a bijective correspondence between this subset of $\mathbf{T}_n$ and $\mathbf{T}_{n-2}$.

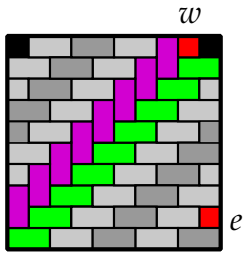


Figure 3.8: When w is flipped up (magenta), there are $n-3$ independently flippable diagonals (grey).

We associate the monominoes of an $(n-2)\times(n-2)$ covering with those of an $n\times n$ covering as follows. Draw the (unique) bond in $\mathbf{T}_{n-2}$, rotated by π radians, in the centre of the bond of $\mathbf{T}_n$. Pairs of monominoes in this drawing which share an edge are *associated*, and naturally, their upward and downward diagonals are associated as well (see Figure 3.9). Note that w and e are associated with the "fixed" corner monominoes of the smaller covering.

Note that a left-side monomino and a right-side monomino cannot both be flipped downward (or upward) in an $n\times n$ covering, if the total number of tiles in the two diagonals is greater than n (see Figure 3.10).

Let T and T' be the unique bond in $\mathbf{T}_n$ and $\mathbf{T}_{n-2}$, respectively. Let α and β be diagonals in T, associated with diagonals α' and β' in T', respectively, and note that α' has one less tile than α, and β' has one less tile than β. Therefore, α' and β'

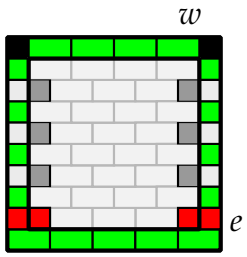


Figure 3.9: Pairs of monominoes in the respective superimposed coverings are associated if they share an edge. Their respective up and down diagonals are also associated.

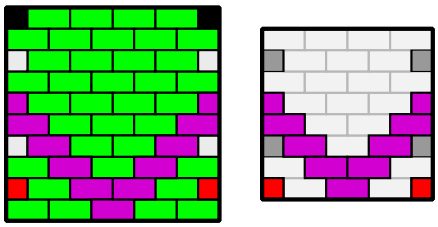


Figure 3.10: In each covering, the two magenta diagonals cannot both be flipped because they intersect.

can be flipped in T' if and only if α and β can both be flipped in T. Thus, for each configuration of flipped diagonals in T, the associated diagonals can be flipped in T', and vice versa (see Figure 3.11).

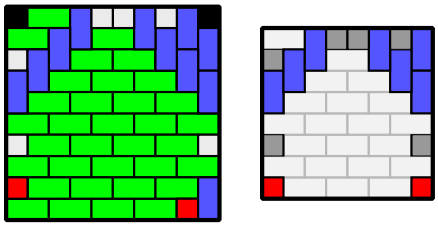


Figure 3.11: A pair of coverings in the bijection. Note that flips of the red monominoes, e and w, are irrelevant.

This bijection implies that for each of the 4 cases where neither e nor w is flipped up, there are exactly $s(n-2)$ coverings in $\mathbf{T}_n$, which accounts for the second term in (3.4).

The odd case is very similar, and is described in [12]. □

3.2.2 Non-maximal tatami coverings of the square

Let $\mathbf{T}_{n,m}$ be the set of $n \times n$ tatami coverings with m monominoes. To find $|\mathbf{T}_{n,m}|$, we use a setup similar to that of Theorem 3.6. Lemmas 3.7, 3.8, and 3.9 on the composition of such coverings seem apparent from Figures 3.12–3.14, but we prove them in general.

We show that any covering in $\mathbf{T}_{n,m}$ with $m < n$ has exactly one bidimer or vortex and we show that m uniquely determines the shortest distance from this source to the boundary. Such a feature determines all tiles in the covering except a number of diagonals that can be flipped independently. Proving the result becomes a matter of counting the number of allowable positions for the bidimer or vortex, each of which contributes a power of 2 to the total count.

For example, the 20×20 covering in Figure 3.12 has a vertical bidimer which forces the placement of the green and blue tiles, while the remaining diagonals are coloured in alternating grey and magenta. There are eight such diagonals, so there are 2^8 coverings of the 20×20 grid, with a vertical bidimer in the position shown. Each of these 2^8 coverings has exactly 10 monominoes.

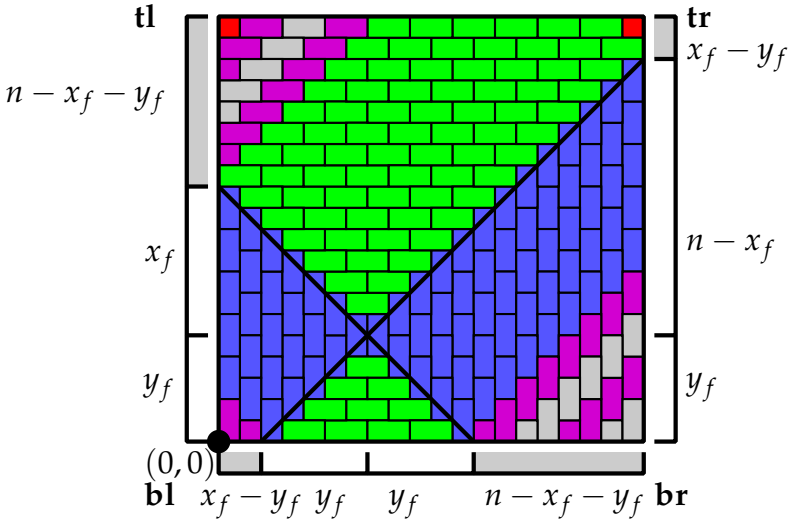


Figure 3.12: Vertical bidimer.

By Corollary 3.3, a covering in $\mathbf{T}_{n,m}$, with $m < n$, must have at least one bidimer or vortex, which we will call f. Let (x_f, y_f) be the centre of f in the cartesian plane, where the bottom left of the $n \times n$ covering is at the origin.

Define X_f as the lines $(x - x_f) + (y - y_f) = 0$ and $(x - x_f) - (y - y_f) = 0$,

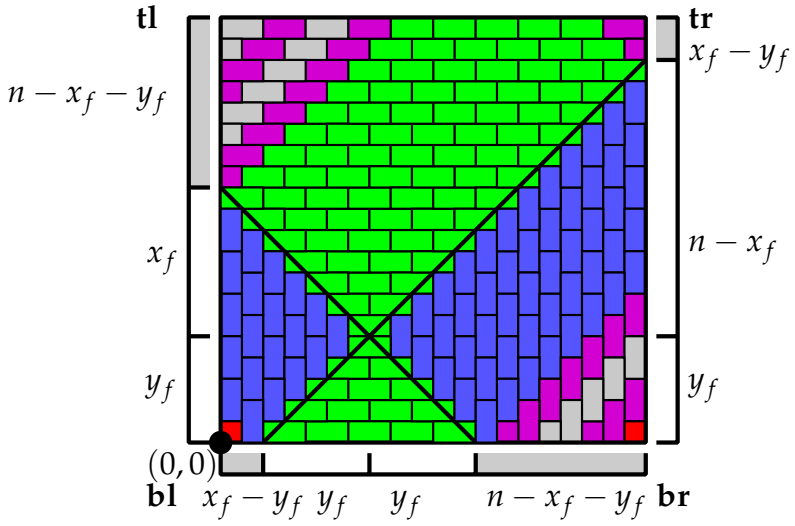


Figure 3.13: Horizontal bidimer.

bounded by the grid boundary, which form an X through (x_f, y_f) (see Figures 3.12-3.14).

Without loss of generality, we may re-orient a covering, by rotating and reflecting, so that $y_f \leqslant x_f \leqslant n/2$. The upper arms of X_f intersect the left and right boundaries of the grid, while the lower arms intersect the bottom. In this range, the *distance* from this feature to the boundary of the grid is y_f.

Lemma 3.7. *Let $T \in \mathbf{T}_{n,m}$, with $m < n$. Then T has exactly one bidimer or vortex, f, but not both, and no vees.*

Proof. If a is another bidimer or vortex in the covering, then X_a intersects X_f, which contradicts the fact that feature diagrams do not intersect. A vee has the same rays as a bidimer, by replacing the adjacent monominoes with a domino, so the only features that may appear, besides f, are loners. □

Let T_f be the covering with f as its only feature. This corresponds to the (unique) bond of $\mathbf{T}_n$, used in Theorem 3.6.

Lemma 3.8. *The covering T_f can be obtained from any covering containing the feature f via a finite sequence of diagonal flips in which each monomino moves at most once. Reversing this sequence gives the original covering.*

Proof. This is a simple modification of the proof of Lemma 3.4. □

Lemma 3.9. *All the diagonals in T_f can be flipped independently.*

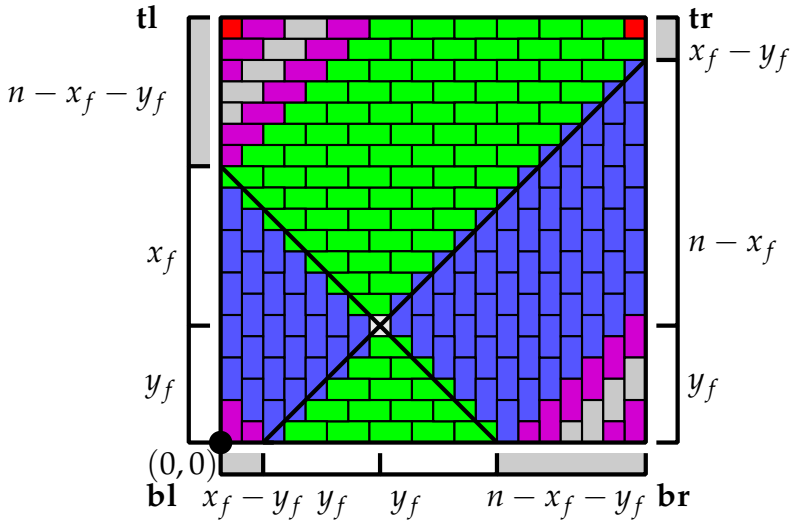


Figure 3.14: Counter clockwise vortex. Note that x_f and y_f are not integers.

Proof. We show that no two diagonals intersect, either at a monomino or a domino.

Let α and β be the monominoes in distinct diagonals, possibly with $\alpha = \beta$, which intersect. Let L_α and L_β be the longest line segments contained in the respective diagonals. Note that their slopes are in $\{1, -1\}$. Their slopes must differ, so we may assume that L_α has slope 1, and L_β has slope -1. The line segments L_α and L_β intersect (inside the grid), and therefore at least one of them instersects with X_f, which makes the corresponding diagonal unflippable. Therefore distinct diagonals may not intersect. □

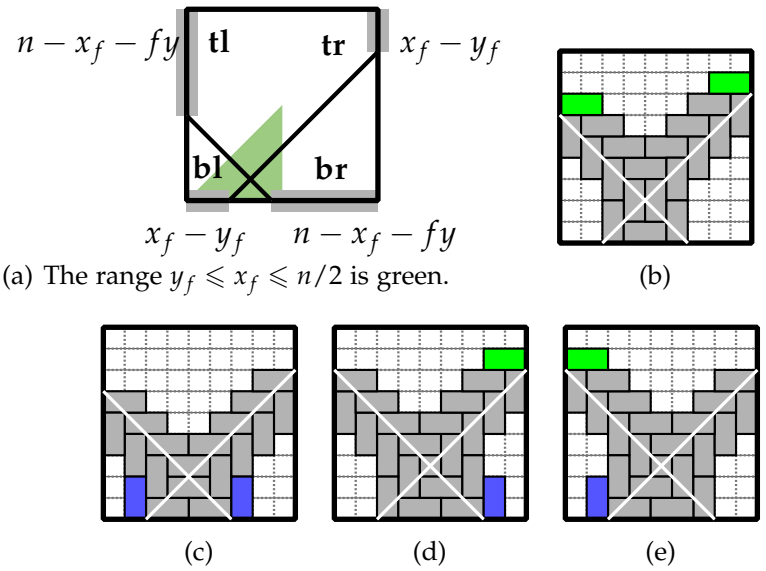
The crux of the argument in Theorem 3.11 is this:

Lemma 3.10. *If $T_f \in \mathbf{T}_{n,m}$, with $y_f \leqslant x_f \leqslant n/2$, then $m = n - 2y_f$ if f is a bidimer, and $m = n - y_f + 1$ if f is a vortex. The number of (flippable) diagonals in T_f is*

$$\begin{cases} n - 2y_f - 2, & \text{if } y_f < x_f; \\ n - 2y_f - 1, & \text{if } y_f = x_f < n/2; \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

Proof. Let **tl**, **tr**, **bl**, and **br** be the segments of the boundary of the grid delimited by X_f, as shown in grey in Figure 3.15(a). We count the number of monominoes and diagonals in T_f by measuring the lengths of **tl**, **tr**, **bl**, and **br**, minus the tiles

in the rays of f, with the different cases illustrated in Figures 3.15(b)–3.15(e).



(a) The range $y_f \leqslant x_f \leqslant n/2$ is green. (b) (c) (d) (e)

Figure 3.15: The segments **tl**, **tr**, **bl**, and **br**. When **tl** and **br** have positive length, one of their combined tiles is part of a ray of f, shown in green or blue. Similarly for **tr** and **bl** combined.

Figures 3.15(b)–3.15(e) show (in general) that if **tl** and **br** are non-zero, then together they contain exactly one grid square covered by the ray of f. The same is true of **tr** and **bl**.

Because of the above, non-zero **tl** and **br** differ in parity, so their sum is odd, and therefore they must contain exactly one corner monomino (which is not in a diagonal). Similarly for **tr** and **bl**.

We tabulate the numbers of monominoes and diagonals in **tl**, **tr**, **bl** and **br** for different (x_f, y_f) in Table 3.1, and add pairs of rows to prove each case of the theorem statement.

(x_f, y_f)	Positions	Monominoes	Diagonals
$y_f = x_f$	**bl** and **tr**	0	0
$y_f < x_f$	**bl** and **tr**	$x_f - y_f$	$x_f - y_f - 1$
$y_f = x_f = n/2$	**tl** and **br**	0	0
$y_f + x_f < n$	**tl** and **br**	$n - x_f - y_f$	$n - x_f - y_f - 1$

Table 3.1: The numbers of monominoes and diagonals in **tl**, **tr**, **bl** and **br** for different (x_f, y_f).

If $y_f < x_f$, then $m = x_f - y_f + n - x_f - y_f = n - 2y_f$ if f is a bidimer, and

$m = n - 2y_f + 1$ if f is a vortex. The number of diagonals is $n - 2y_f - 2$, as required for this case

If $y_f = x_f < n/2$, then $m = 0 + n - y_f - y_f = n - 2y_f$ if f is a bidimer, and $n - 2y_f + 1$ if f is a vortex. The number of diagonals is $n - 2y_f - 1$.

If $y_f = x_f = n/2$, then $m = 0$ if f is a bidimer, and $m = 1$ if it is a vortex. The number of diagonals is 0. □

We sum up the number of positions which give a particular m in Lemma 3.10, to count all such coverings.

Theorem 3.11. *If n and m have the same parity, and $m < n$, then $T(n, n, m) = m2^m + (m + 1)2^{m+1}$.*

Proof. We count the number of coverings T_f in $\mathbf{T}_{n,m}$, for each bidimer or vortex, f, satisfying the conditions of Lemma 3.10, and for each of these we count the coverings obtainable via a set of (independent) diagonal flips.

If f is a bidimer, then $m = n - 2k$, where k is the shortest distance from (x_f, y_f) to the boundary of the grid. These positions where f can be centred are

$$(k,k), (k,k+1), (k,k+2), \ldots (k, n-k), (k+1, n-k), \ldots,$$
$$(n-k, n-k), (n-k, n-k-1), \ldots (n-k, k), (n-k-1, k), \ldots (k+1, k),$$

of which there are $4(n - k - k)$. We apply Lemma 3.10 by making the necessary rotations and reflections, so that when $m > 0$, the four positions $(k, k), (k, n - k), (n - k, n - k)$ and $(n - k, k)$, have $m - 1$ diagonals, while the remaining $4m - 4$ positions have $m - 2$ diagonals.

The same logic applies when f is a vortex and $m > 1$, and the results are in Table 3.2.

By Lemmas 3.8 and 3.9, all of the coverings which contain f can be obtained from T_f by making a set of independent flips. Thus, the number of these is $2^{d(f)}$, where $d(f)$ is the number of diagonals in T_f.

Lemma 3.7 tells us that there is no other way to obtain a covering in $\mathbf{T}_{n,m}$, so we conclude by summing the $2^{d(f)}$s for each f such that $T_f \in \mathbf{T}_{n,m}$. Each term in the following sum comes from a row of Table 3.2, in the same respective order, and similarly, the three factors in each sum term come from the last three columns

Type	Feature	Positions	Diagonals
$x_f = y_f$	h and v bidimers	4	$m-1$
$x_f = y_f$	cc and c vortices	4	$m-2$
$x_f < y_f$	h and v bidimers	$4(m-1)$	$m-2$
$x_f < y_f$	cc and c vortices	$4(m-2)$	$m-3$

Table 3.2: Horizontal, vertical, counterclockwise and clockwise are abbreviated as h, v, cc and c, respectively. We assume $m > 0$ if f is a bidimer, and $m > 1$ if f is a vortex.

of the table.

$$\begin{aligned} T(n,m) =& 2 \cdot 4 \cdot 2^{m-1} + 2 \cdot 4 \cdot 2^{m-2} + 2 \cdot 4(m-1) \cdot 2^{m-2} + 2 \cdot 4(m-2) \cdot 2^{m-3} \\ =& 2 \cdot 2^{m+1} + 2 \cdot 2^m + (m-1)2^{m+1} + (m-2)2^m \\ =& m2^m + (m+1)2^{m+1}. \end{aligned}$$

That is, there are $m2^m$ coverings with vortices and $(m+1)2^{m+1}$ with bidimers. □

This completes the enumeration of $n \times n$ tatami coverings with m monominoes. Summing over $T(n,n,m)$ over all m yields a nice formula.

Corollary 3.12. *The number of $n \times n$ tatami coverings is $2^{n-1}(3n-4)+2$.*

Proof. By Corollary 3.3, we have that $T(n,n,m) = 0$ when $m > n$. Let $T(n,n) = \sum_{m \geqslant 0} T(n,n,m)$, so that

$$T(n,n) = n2^{n-1} + \sum_{i=1}^{\lfloor n/2 \rfloor} \left((n-2i)2^{n-2i} + (n-2i+1)2^{n-2i+1} \right),$$

and notice that the sum simplifies to

$$T(n,n) = n2^{n-1} + \sum_{i=1}^{n-1} i2^i.$$

Now we use the fact that $2^k + 2^{k+1} + \cdots + 2^{n-1} = 2^n - 1 - 2^k + 1 = 2^n - 2^k$ to

rearrange the sum.

$$\begin{aligned} T(n,n) =& n2^{n-1} + \sum_{i=1}^{n-1} i2^i \\ =& n2^{n-1} + (n-1)2^n - \sum_{i=1}^{n-1} 2^i \\ =& 2^{n-1}(3n-4) + 2 \end{aligned}$$

□

By cross referencing the result of Corollary 3.12 with the On-Line Encyclopedia of Integer Sequences, a surprising correspondence with integer compositions of n is apparent (see A027992 in [27]).

Proposition 3.13 (Erickson, Schurch, 2011, [14])**.** *The number of $n \times n$ tatami coverings is equal to the sum of the squares of all parts in all compositions of n.*

Proof. See [14]. □

3.3 Tatami coverings of proper rectangles

Compared with tatami coverings of the square, little attention has been paid to the counts for $T(r,c,m)$ when $r < c$. Similar techniques may be applied, however, by considering diagonal flips and other feature diagrams of the T-diagram. Rectangular coverings up to 14×14 have been counted by brute force and the counts are listed in Tables A.1–A.17, and A.19.

Calculating non-trivial $T(r,c,m)$, when $r < c$, is an open problem for all fixed m, with $m > 1$, however, significant progress has been made by Jennifer Woodcock on counting those with maximum monominoes. Conjecture 3.14 agrees with the counts produced by brute force, at least up to $(r,c) = (12,13)$ (see Table A.19).

Conjecture 3.14 (Woodcock, 2010, (private communication))**.** *Let $T_{max}(r,c)$ denote the number of $r \times c$ tatami coverings with the maximum possible number of monominoes.*

For $r \geqslant 3$,

$$T_{max}(r,c) = \begin{cases} 2^{r-4}(r+4)(r+2) & \text{if } c > 2r+1 \text{ and } r \text{ and } c \text{ are both even} \\ 2^{r-4}(r+3)^2 & \text{if } c > 2r+1 \text{ and } r \text{ is odd} \\ 2^{r-6}(r+2)^2 & \text{if } c > 2r+1 \text{ and } r \text{ is even and } c \text{ is odd} \\ 2^{r-4}\,(3r-c+4)\,(c-r+2) & \text{if } r \equiv c \bmod 2 \text{ and } r+1 \leqslant c \leqslant 2r+1 \\ 2^{r-4}\,(3r-c+4)\,(c-r+2) + 2^{r-4} & \text{if } r \text{ is odd and } c \text{ is even and } r+2 \leqslant c \leqslant 2r \\ 2^{r-6}(29r+17) & \text{if } r \text{ is odd and } c = r+1 \\ 2^{r-6}\,(3r-c+4)\,(c-r+2) & \text{if } r \text{ is even and } c \text{ is odd} \\ \quad +2^{r-6}(2r-2c-3) & \quad \text{and } r+1 \leqslant c \leqslant 2r+1 \\ r2^{r-1} & \text{if } r = c. \end{cases} \tag{3.5}$$

The case where $m = 0$ has the historical distinction of motivating the present research. Exercise 7.1.4.215 in "The Art of Computer Programming", by Don Knuth ([19]), asks for a generating function that counts the number of domino-only tatami coverings of fixed height. Ruskey and Woodcock, using ideas from a decomposition due to Hickerson ([17]), provide the following solution.

Theorem 3.15 (Ruskey, Woodcock, 2009, [26])**.** *Let $T(r,c,0)$ be the number of tatami coverings of the $r \times c$ grid with 0 monominoes. For fixed-height grids, these are counted by the rational generating function,*

$$\sum_{c \geqslant 0} T(r,c,0)z^c = \begin{cases} 1 & \text{for } r = 0 \\ \frac{1}{1-z^2} & \text{for } r = 1 \\ \frac{1+z^2}{1-z-z^3} & \text{for } r = 2 \\ \frac{1+z^{r-1}+z^{r+1}}{1-z^{r-1}-z^{r+1}} & \text{for } r \text{ odd}, 3 \leqslant r \leqslant c \\ \frac{(1+z)(1+z^{r-2}+z^r)}{1-z^{r-1}-z^{r+1}} & \text{for } r \text{ even}, 4 \leqslant r \leqslant c. \end{cases}$$

Proof. See [26]. □

The case where $m = 1$ is given by Alhazov, Iwamoto, and Morita, in [1], as a rational generating function.

Theorem 3.16 (Artiom Alhazov, Kenichi Morita, Chuzo Iwamoto, 2009, [1])**.** *The number of coverings of the $r \times c$ grid, with $r \leqslant c$, and exactly 1 monomino, is "generated"*

by

$$A_r(z) + \sum_{r \leqslant c} T(r,c,1)z^c = \begin{cases} \frac{z}{(1-z^2)^2} & \text{for } r = 1, \\ \frac{2z+6z^3-4z^5-4z^7}{(1-z^2-z^4)^2} & \text{for } r = 3, \\ \frac{2z+4z^{r-2}+6z^r-4z^{2r-3}-8z^{2r-1}-4z^{2r+1}}{(1-z^{r-1}-z^{r+1})^2}, & \text{for odd } r \geqslant 5, \end{cases}$$

where $A_r(z)$ is a polynomial of degree at most $r-1$.

Recall that $T(r,c) = \sum_{m \in \mathbb{Z}} T(r,c,m)$. As it turns out, the generating functions for $T(r,c)$ are also rational functions, for fixed r. This result, due to Mark Schurch, is restated here, but we give a new proof using the transfer matrix method (see [28]). For each fixed r, an application of the transfer matrix method produces the generating function $T_r(\lambda)$, defined in Theorem 3.17.

Theorem 3.17 (Erickson, Ruskey, Schurch, Woodcock, 2011, [12])**.** *The generating function*

$$T_r(\lambda) = \sum_{c \geqslant 0} T(r,c)\lambda^c,$$

is a rational function, where $T(r,c)$ is defined above.

Proof. See [28], Section 4.7, pg. 241 on the transfer matrix method, for undefined terms.

Let $D = (V,E)$ be a digraph. The vertex set, V, comprises all coverings of the $r \times 1$ grid, by vertical dominoes, ▮, monominoes, □, and the left square of horizontal dominoes, ▬ (see Figure 3.16). The number of coverings, $|V|$, is equal to $p(r)$, where $p(n) = p(n-2) + 2p(n-1)$, with initial conditions $\{p(1) = 2, p(2) = 5\}$ (see A000129 in [27]).

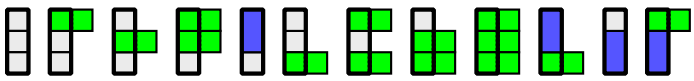


Figure 3.16: The $r \times 1$ coverings by vertical dominoes, and two types of "monomino", shown for the case of $r = 3$.

A tatami covering of the $r \times c$ grid can be regarded as a walk in D, of length c. There is an arc (i,j) in D whenever vertex j can follow vertex i in an $r \times c$ tatami covering. There are two rules that govern this.

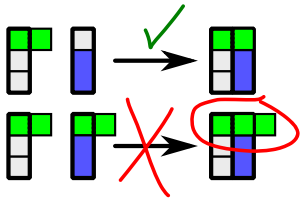


Figure 3.17: A horizontal domino can only be extended by a monomino.

1. If a square of vertex i is covered by ▭, the corresponding square of vertex j must be covered by □ (see Figure 3.17).

2. If two adjacent squares of vertex i are covered by distinct tiles, and neither of them are ▭, then the corresponding squares of j are covered by a ▯.

Let A be the adjacency matrix of D. Theorem 4.7.2 of [28] says that the number of length-c walks in D that begin at vertex i and end at vertex j is the coefficient of λ^c in

$$F_{ij}(D,\lambda) = \frac{(-1)^{i+j}\det(I-\lambda A : j,i)}{\det(I-\lambda A)}. \tag{3.6}$$

We find that $F_{ij}(D,\lambda) = (I-\lambda A)^{-1}_{ij}$, which is a rational function. □

Further to the above proof, the last column in a covering cannot contain ▭. For example, with $r = 3$ the last column may only be vertex $1, 5$, or 11. The rational function we want is

$$T_r(\lambda) = 1 + \lambda \left(\sum_{\substack{1 \leqslant i \leqslant p(r) \\ j \in V_e}} F_{ij}(D,\lambda) \right),$$

where V_e is the subset of vertices that can be the last column of an $r \times c$ covering. The result is shifted by λ because we want to count columns of the $r \times c$ grid, rather than grid-lines.

The fractal nature of the adjacency matrices, for various r, is unsurprising, it is worth comparing its density with that of its non-tatami counterpart (compare Figure 3.18 with Figure 3.19).

Remark that the degree of the numerator in Equation (3.6) satisfies

$$\deg((-1)^{i+j}\det(I-\lambda A)) < p(r),$$

Figure 3.18: The 2378×2378 adjacency matrix for $r = 9$, with the tatami constraint. Compare its density, approximately 0.002, with that of Figure 3.19.

since the dimensions of A are $p(r) \times p(r)$. Therefore, the coefficients of

$$(-1)^{i+j} \det(I - \lambda A : j, i)$$

can be determined from the first $p(r)$ coefficients of the series expansion. This is faster than finding the inverse mentioned in the proof of Theorem 3.17, if the initial values are computed efficiently. Note that they can be calculated (inefficiently) with $(A^n)_{ij}$, since this is the number of length-n walks from i to j, by Theorem 4.7.1 of [28]. See Algorithm 1 for this method, and see `http://alejandroerickson.com/tatami` for an implementation in the Maple programming language.

Algorithm 1 Calculate $T_r(\lambda)$ without using $(I - \lambda A)^{-1}$.

Require:

$g(\lambda) \leftarrow \det(I - \lambda A)$

$endWalks \leftarrow \{v \in V(D) : v \text{ has no '}$ ▬ $\text{'} \}$

$T(\lambda) \leftarrow |endWalks| + \sum_{c=1}^{p(r)} \left(\sum_{\substack{1 \leqslant i \leqslant p(r) \\ j \in endWalks}} (A^c)_{ij} \right) \lambda^c$ ▷ Initial conditions.

$f(\lambda) \leftarrow T(\lambda) g(\lambda) \mod \lambda^{p(r)}$

$T(\lambda) \leftarrow 1 + \lambda \frac{f(\lambda)}{g(\lambda)}$ ▷ Shift by 1 and add the $r \times 0$ covering.

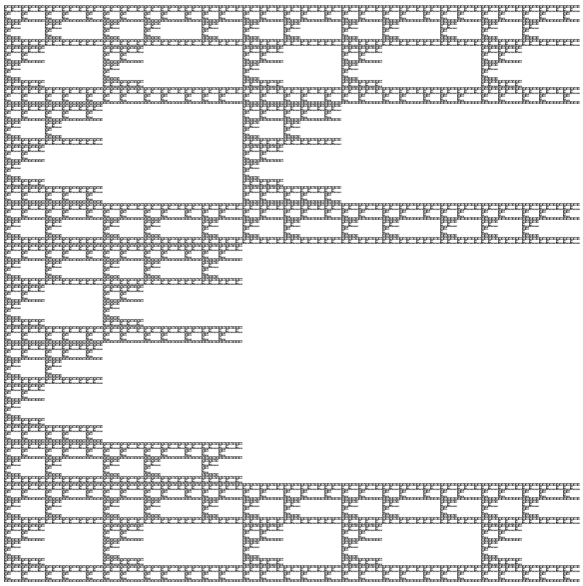


Figure 3.19: The 2378×2378 adjacency matrix for $r = 9$, without the tatami constraint. Its density, approximately 0.108, is much higher than that of Figure 3.18.

Let $L(\lambda), P(\lambda)$, and $Q(\lambda)$ be polynomials such that $T_r(\lambda) = L(\lambda) + P(\lambda)/Q(\lambda)$, where $Q(0) = 1$ and $\deg(P(\lambda)) < \deg(Q(\lambda))$ (the parameter r is understood). Table 3.3 contains the coefficients of $Q(\lambda)$ up to $r = 11$, and Table 3.4 gives $L(\lambda)$ and $P(\lambda)$ up to $r = 10$. The first 14 cofficients of each $T_r(\lambda)$ are in Table A.17, for $1 \leqslant r \leqslant 14$.

Salient patterns in these coefficients are summarized in Conjectures 3.18 and 3.19. Note that Conjecture 3.18 implies $Q(\lambda)$ is a self-reciprocal polynomial for $r \equiv 2 \pmod 4$. Interestingly, the corresponding generating functions for $m = 0, 1$, from Theorems 3.16 and 3.15, do not have a similar self-reciprocal property.

Conjecture 3.18. *Let $L(\lambda)$, $P(\lambda)$, and $Q(\lambda)$ be as defined above, where $P(\lambda)$ and $Q(\lambda)$ are relatively prime polynomials,* $\deg(Q(\lambda)) = n$, *and $r \geqslant 1$. Then,*

$$Q(\lambda) = \begin{cases} -\lambda^n Q\left(\dfrac{1}{\lambda}\right), & \text{if } r \equiv 0 \pmod 4, \\ -\lambda^n Q\left(-\dfrac{1}{\lambda}\right), & \text{if } r \equiv 1 \pmod 4, \\ \lambda^n Q\left(\dfrac{1}{\lambda}\right), & \text{if } r \equiv 2 \pmod 4, \\ \lambda^n Q\left(-\dfrac{1}{\lambda}\right), & \text{if } r \equiv 3 \pmod 4. \end{cases}$$

A mod 4 pattern also seems to occur in the degrees of the denominators of $T_r(\lambda)$. The rigid structure we encounter in tatami tilings prompts us to infer this pattern upon all values as well.

Conjecture 3.19. *Let $Q(\lambda)$ be as defined above. Then,*

$$\deg(Q(\lambda)) = \begin{cases} 8m^2+2m+1, & \text{if } r \equiv 0 \pmod 4, \\ 8m^2+4m+2, & \text{if } r \equiv 1 \pmod 4, \\ 8m^2+10m+4, & \text{if } r \equiv 2 \pmod 4, \\ 8m^2+8m+6, & \text{if } r \equiv 3 \pmod 4. \end{cases}$$

We conclude this section by noting that we have not been able to devise a uniform presentation of the generating function of $T_r(\lambda)$ similar to what was done in [26].

It would also be interesting to consider the generating function $T_r(x, y, \lambda)$, in which the coefficient of $x^h y^v \lambda^m$ is the number of coverings with h vertical dominoes, v horizontal dominoes, and m monominoes.

Coefficients of $Q(\lambda)$ are ordered from left to right by ascending degree, and then folded like these arrows: ↪ for $r \leqslant 3$, ↩ for $r = 4, 5, 6, 7, 9$, and ↻ for $r = 8, 10, 11$.

r	q	$Q(\lambda)$	r	q	$Q(\lambda)$
1	2	1,-1,-1	5	14	1,-1,-1, 1,-3, 1,-5, 2 -1,-1, 1, 1, 3, 1, 5
2	4	1, -2, 0, -2, 1	6	22	1,-1,-1, 1,-1,-2, 2,-10, 9,-1, 4, 6 1,-1,-1, 1,-1,-2, 2,-10, 9,-1, 4
3	6	1, -1, -2, 0, -2, 1, 1	7	22	1,-1,-3, 3, 4,-4,-9, 7, 6,-5, 2, 0 1, 1,-3,-3, 4, 4,-9,-7, 6, 5, 2
4	11	1,-1,-1,-1, 1,-7 -1, 1, 1, 1,-1, 7	8	37	1,-1,-1, 1,-1, 1,-1,-3, 3,-13, 12 -1, 1, 1,-1, 1,-1, 1, 3,-3, 13,-12 -34, 2,-6,-20, 6,-12,-0,-0 34,-2, 6, 20,-6, 12,-0,-0
9	42	1,-1,-1, 1,-1, 1,-1, 1,-5, 3,-11, 8, 6,-4, 14,-8, 20,-2, 28,-2, 24 -1,-1, 1, 1, 1, 1, 1, 1, 5, 3, 11, 8,-6,-4,-14,-8,-20,-2,-28,-2,-24,-10			
10	56	1,-1,-1, 1,-1, 1,-1, 1,-1,-4, 4,-16, 15, 1,-1 1,-1,-1, 1,-1, 1,-1, 1,-1,-4, 4,-16, 15, 1,-1 -120, 68,-78,-18, 18,-66, 66,-2, 7, 41,-23, 33,-17, 17 68,-78,-18, 18,-66, 66,-2, 7, 41,-23, 33,-17, 17			
11	54	1,-1,-5, 5, 13,-13,-27, 27, 48,-48,-83, 81, 125,-120,-160 1, 1,-5,-5, 13, 13,-27,-27, 48, 48,-83,-81, 125, 120,-160 -34, 83, 89,-156,-165, 199, 210,-202,-206, 185, 193,-154 -34,-83, 89, 156,-165,-199, 210, 202,-206,-185, 193, 154			

Table 3.3: Coefficients of denominators, $Q(\lambda)$, where $q = \deg(Q(\lambda))$. The ordering reflects the patterns in Conjecture 3.18.

r	l	p	$L(\lambda)$	$P(\lambda)$
1	–	0	0	1
2	–	3	0	1,0,2,-1
3	2	5	-3,1,1	4,-2,2,6,-10,2
4	3	10	-13,3,3,2	14,-12,10,0,10,-104,114,-80,34,12,-2
5	4	13	-28,10,17,10,3	29,-31,24,60,-97,61,-196,83,31,-84,96,-1,13,-8
6	5	21	-90,27,33,32,18,10	
$P(\lambda)$			91,-105,46,146,-172,-114,166,-1066,1099,-827,403,409,90,14,930, -1060,682,-318,-71,141,-160,28	
7	6	21	-169,71,139,76,54,42,17	
$P(\lambda)$			170,-220,-242,772,-18,-1110,-508,2050,-848,-1662,1850,38,-468, 160,1332,-650,-996,730,218,-396,-126,86	
8	7	36	-505,176,251,178,71,138,98,48	
$P(\lambda)$			506,-648,206,752,-1041,720,-1116,-1096,1614,-8068,8612,-5759,540, 6784,-6729,13628,-3028,2004,19791,-22386,14426,-5029,-11124,8876, -11000,2763,-1960,-5916,6379,-4200,2254,120,-426,542,-754,1009, -222	
9	8	41	-897,425,956,408,128,126,250,224,88	
$P(\lambda)$			898,-1268,402,1708,-1982,1146,-1740,970,-5188,4686,-13780,11660, -3544,-7718,16922,-15074,23100,-12852,29344,-10828,33924,-12798, -7010,15094,-28120,12296,-18636,-924,-14040,-5490,-9394,2922, 2578,-1576,3950,-176,74,1068,534,1352,748,-550	
10	9	55	-2593,999,1736,946,-40,18,186,674,504,224	
$P(\lambda)$			2594,-3504,910,3704,-5546,3559,-5834,4186,-6074,-8098,12070, -52250,57132,-35426,2000,53526,-66398,125591,-117208,163754, -44712,34452,202264,-226674,189166,-87718,-187340,287264,-466224, 376408,-381724,59088,-7768,-238846,220182,-113850,18958,129296, -120950,139892,-82782,75555,-19764,24540,28030,-32766,23466, -14478,1712,-366,-208,1400,-2760,4075,-5480,1302	

Table 3.4: Coefficients of $L(\lambda)$ and $P(\lambda)$ in ascending order of degree, where $l = \deg(L(\lambda))$ and $p = \deg(P(\lambda))$. For $r \geqslant 5$, the coefficients of $P(\lambda)$ are displayed in the next row.

Chapter 4

Square grids, maximum monominoes, v vertical dominoes

Recall that $\mathbf{T}_n$ is the set of monomino-domino tatami coverings of the $n \times n$ grid with the maximum number, n, of monominoes, oriented so that they have a monomino in each of the top left and top right corners. On the basis of some computer investigations, Don Knuth discovered that the generating polynomial for small tatami coverings of $\mathbf{T}_n$, with respect to the number of vertical dominoes they contain, is a product of cyclotomic polynomials and a mainly mysterious, irreducible polynomial (private communication, December 2010). Knuth's discovery and our own observations motivated Conjecture 4 in [12], which is presented here as Equation (4.2). In this chapter we generalise and prove Knuth's cyclotomic factors, and determine some important properties of the mysterious polynomial.

Let $H(n,k)$ be the number of coverings in $\mathbf{T}_n$ with exactly k horizontal dominoes, and let $V(n,k)$ be the number with exactly k vertical dominoes. Let $S_n(z) = \prod_{i=1}^{n}(1+z^i)$. We prove that the polynomial

$$\mathcal{H}_n(z) := 2\sum_{i=1}^{\lfloor \frac{n-1}{2} \rfloor} S_{n-i-2}(z)S_{i-1}(z)z^{n-i-1} + \left(S_{\lfloor \frac{n-2}{2} \rfloor}(z)\right)^2, \tag{4.1}$$

is equal to $\sum_{k\geqslant 0} H(n,k)z^k$ for odd n and is equal to $\sum_{k\geqslant 0} V(n,k)z^k$ for even n. The key step is to show the relationship between diagonal flips, defined in Chapter 3, and the number of subsets of $\{1,2,\ldots,n\}$ whose elements sum to a given k.

Knuth's observation generalises to

$$\mathit{VH}_n(z) = P_n(z) \prod_{j \geqslant 1} S_{\left\lfloor \frac{n-2}{2^j} \right\rfloor}(z), \tag{4.2}$$

where $P_n(z)$ is the "mysterious" polynomial. We prove Equation (4.2) in Theorem 4.6.

The remaining factors of Equation (4.2) are of the form $S_k(z)$, where k is a binary right shift of $n-2$, and the complete factorisation of these is known in general. The *ith cyclotomic polynomial*, $\Phi_i(z)$, is defined as $\prod_{\omega \in \Omega}(z-\omega)$, where Ω is the set of ith primitive roots of unity. Lemma 5 in [12] states that

$$S_k(z) = \prod_{j=1}^{k} \left(\phi_{2j}(z)\right)^{\left\lfloor \frac{k+j}{2^j} \right\rfloor}, \tag{4.3}$$

and cyclotomic polynomials are known to be irreducible. Thus $\mathit{VH}_n(z)$ can apparently be factored completely as

$$\mathit{VH}_n(z) = P_n(z) \prod_{j \geqslant 1} \Phi_{2j}(z)^{\left\lfloor \frac{n-2}{2^j} \right\rfloor}. \tag{4.4}$$

We have verified the irreducibility of $P_n(z)$ for $1 < n < 200$ (the degree of $P_{199}(z)$ is 13022 and its largest coefficient has 55 digits), and thus we hope that Equation (4.4) is the complete factorisation of $\mathit{VH}_n(z)$ for all $n \geqslant 2$.

The class of polynomials, $P_n(z)$, has some compelling properties, some of which are proven, others which are empirical. For example, we observe in Conjecture 4.9 that the alternating sums of $P_n(z)$ are the coefficients of the ordinary generating function

$$\sum_{n \geqslant 2} P_n(-1) z^{n-2} = \frac{(1+z)(1-2z)}{(1-2z^2)\sqrt{1-4z^2}},$$

for $1 < n < 200$. If the conjecture is true, then $P_{2(n+1)}(-1) = \binom{2n}{n}$. Furthermore $P_n(-1)$ is equal to the sum of the absolute values of the coefficients of $P_n(z)$, only for $n \geqslant 20$. This second fact is surprising, considering the way $P_n(z)$ is derived – why $n \geqslant 20$?

The complex roots of $P_n(z)$ appear to cluster neatly around the unit circle, and form convergent sequences as $n \longrightarrow \infty$ (see Figures 4.5–4.6).

Theoretical progress on $P_n(z)$ comprises Theorem 4.7 and Theorem 4.8. The former states that $\deg(P_n(z)) = \sum_{k=1}^{n-2} Od(k)$, where $Od(n)$ is the largest odd divisor of n. We prove in Theorem 4.8 that for all $n \geqslant 2$, the sum of the coefficients of $P_n(z)$ is equal to $n2^{\nu(n-2)-1}$, where $\nu(n)$ is the number of 1-bits in the binary representation of n.

Once again, we employ the diagonal flip (see Definition 3.1). The added observation that a diagonal flip changes the orientation of some dominoes, enables us to further exploit it. The crux of the argument uses the partition of $\mathbf{T}_n$ from Theorem 2 of [14] which reveals diagonal flips each with $1, 2, \ldots, k$ dominoes, respectively, that can be flipped independently. We use this to express $VH_n(z)$ in terms of $S_k(z)$, the generating polynomial for the number of subsets of $\{1, 2, \ldots, k\}$ whose elements sum to i.

4.1 Representing a covering as a string

We describe binary and ternary string representations for $n \times n$ coverings with n monominoes. Recall that each monomino, besides the two corner monominoes, is in exactly two diagonals in the bond, and in a given covering a monomino is flipped in one of these diagonals, or it is unflipped. A ternary symbol for each monomino indicates which of the three possible states it assumes. Each covering is described by a unique string of these ternary symbols, called *trits*, represented in the same order as the following indexed labelling (see caption at Figure 4.1(c)).

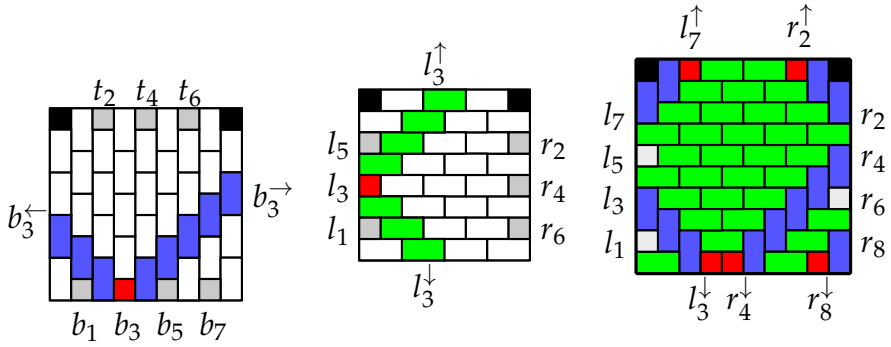
Monominoes and their diagonals are labelled as shown in Figure 4.1, such that the index, i, of a monomino is equal to the length of one of its diagonals, and $n - i - 1$ is the length of the other. This relationship between diagonal length and index is helpful in proving Lemma 4.1.

The ternary string representing the 10×10 covering in Figure 4.1(c) is $s = (0, 1, -1, 0, 0, 1, -1)$, where $s_i = 1$ if the i^{th} monomino is flipped upward, $s_i = -1$ if it is flipped downward, and $s_i = 0$ if it is unflipped.

The ternary symbol is theoretically significant because it unites pairs of diagonals associated with the same monomino, but computer hardware prefers bit-strings. Therefore, we define the binary analogue where each trit, s_i, is replaced

by two bits, p. They are related by

$$s_i = -1 \iff p_i = 10, \tag{4.5}$$
$$s_i = 0 \iff p_i = 00; \text{ and,} \tag{4.6}$$
$$s_i = 1 \iff p_i = 01. \tag{4.7}$$



(a) (b) (c)

Figure 4.1: Labelling for $\mathbf{T}_n$. *(a)* For odd n, monominoes are labelled t_i and b_i. The distances from t_i and b_i to the left boundary are both i. *(b)* For even n, monominoes are labelled l_i and r_i. The distances from l_i to the bottom boundary, and from r_i to the top boundary, are both i. *(c)* The covering, $(0, 1, -1, 0, 0, 1, -1)$.

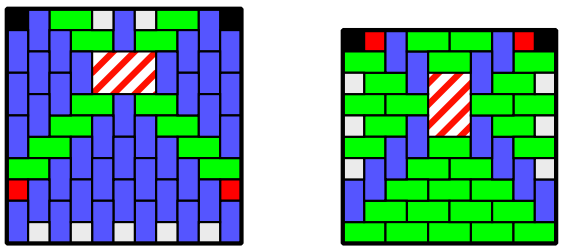
We use $l_i^{\uparrow}, l_i^{\downarrow}, r_i^{\uparrow}, r_i^{\downarrow}, t_i^{\rightarrow}, t_i^{\leftarrow}, b_i^{\rightarrow}, b_i^{\leftarrow}$ to denote the diagonals that the monominoes l_i, r_i, t_i, b_i can be flipped on. Naturally, l_i and r_i can only be (diagonally) flipped up or down, whilst t_i and b_i can only be flipped left or right.

Let $d_n(a)$ be the number of dominoes in the diagonal a, also called the length or size of the diagonal. It is a function of the index and direction of a:

$$d_n(a) = \begin{cases} i, & \text{if } a \in \{l_i^{\downarrow}, r_i^{\uparrow}, t_i^{\leftarrow}, b_i^{\leftarrow}\}; \\ n - i - 1, & \text{if } a \in \{l_i^{\uparrow}, r_i^{\downarrow}, t_i^{\rightarrow}, b_i^{\rightarrow}\}. \end{cases}$$

Flipped diagonals which intersect are called *conflicting*, and can occur as one of two types (see Figure 4.2).

Type 1 A pair of diagonals with monominoes originating on the same boundary are flipped toward one another (e.g. $(t_i^{\rightarrow}, t_j^{\leftarrow})$ for some $i < j$).



(a) (b)

Figure 4.2: Example of, *(a)*, a Type 1 conflict, and, *(b)*, a Type 2 conflict.

Type 2 A pair of diagonals with monominoes originating on opposite boundaries are flipped in the same direction (e.g. $(l_i^{\uparrow}, r_j^{\uparrow})$) and their combined length is at least n (see Table 4.1).

Pair	Type 2 $\iff$
$l_i^{\downarrow}, r_j^{\downarrow}$	$j \leqslant i-1$
$l_i^{\uparrow}, r_j^{\uparrow}$	$i \leqslant j-1$
$t_i^{\leftarrow}, b_j^{\leftarrow}$	$n \leqslant j+i$
$t_i^{\rightarrow}, b_j^{\rightarrow}$	$i+j \leqslant n-2$

Table 4.1: Conditions for Type 2 conflicts.

Lastly, if a is a diagonal containing a given monomino, let $\bar{a}$ be the monomino's other diagonal.

4.1.1 A partition of $\mathbf{T}_n$

Let $\mathbf{T}_n(a) \subseteq \mathbf{T}_n$, where a is a diagonal such that $d_n(a) \geqslant d_n(\bar{a})$, be defined as the collection of coverings in $\mathbf{T}_n$ in which a is the *longest flipped diagonal*; for each flipped diagonal b, distinct from a, we have $d_n(b) < d_n(a)$.

Let $\mathbf{T}_n(\varnothing)$ be the set of coverings in which no monomino is flipped on its longest diagonal. Note the distinction between a monomino flipped on its longest diagonal, and the longest flipped diagonal in the whole covering.

The sets $\mathbf{T}_n(\varnothing)$ and $\mathbf{T}_n(a)$, for each diagonal a defined above, are a partition of $\mathbf{T}_n$.

4.2 Enumerating coverings in $\mathbf{T}_n$ with k vertical dominoes

Let $S(s,k)$ be the number of subsets of $\{1,2,\ldots,s\}$ whose sum is k. The number of coverings with k vertical (or horizontal) dominoes is expressible in terms of this function by making independent flips of diagonals whose lengths are some subset $\{1,2,\ldots,s\}$. We identify these sets of diagonals in the proof of Lemma 4.1.

Lemma 4.1. *Let $V(n,k)$ and $H(n,k)$ be the number of coverings in $\mathbf{T}_n$ with exactly k vertical and horizontal dominoes, respectively. If n is even, then $V(n,k)$ is equal to*

$$\begin{aligned} V\!H(n,k) :=& 2\sum_{i=1}^{\lfloor\frac{n-1}{2}\rfloor}\left(\sum_{\substack{k_1+k_2=\\ k-(n-i-1)}} S(n-i-2,k_1)S(i-1,k_2)\right) && \text{(4.8a)}\\ &+\sum_{k_1+k_2=k} S\left(\left\lfloor\frac{n-2}{2}\right\rfloor,k_1\right)S\left(\left\lfloor\frac{n-2}{2}\right\rfloor,k_2\right). && \text{(4.8b)}\end{aligned}$$

When n is odd, $V\!H(n,k)$ is equal to $H(n,k)$.

Proof. Each outer sum term of (4.8a) adds the coverings for $\mathbf{T}_n(a)$, for some diagonal a, and the term (4.8b) counts those in $\mathbf{T}_n(\varnothing)$.

Case n even: The bond covering in $\mathbf{T}_n$ consists only of horizontal dominoes, and flipping the diagonal a contributes $d_n(a)$ vertical dominoes. Diagonals $l_i^{\uparrow}$ and $r_i^{\uparrow}$ have even length, for all i, while $l_i^{\downarrow}$ and $r_i^{\downarrow}$ have odd length. We use this fact to find sets of diagonals which have lengths $1,2,\ldots,s$, for some $s\in\mathbb{N}$, by combining allowable diagonals in opposite corners, for each $\mathbf{T}_n(a)$. Table 4.2 shows the lengths of the longest allowable diagonals in each corner for each $\mathbf{T}_n(a)$, and from this we can find the required sets of diagonals. For example, the allowable diagonals in $\mathbf{T}_n(l_i^{\uparrow})$ are shown in Figure 4.3(a) (for $(n,i)=(18,5)$) and their respective lengths are

$$\begin{array}{ll} l_1^{\downarrow}, l_3^{\downarrow}, \ldots, l_{i-2}^{\downarrow} & 1,3,\ldots,i-2,\\ l_{i+2}^{\uparrow}, l_{i+4}^{\uparrow}, \ldots, l_{n-3}^{\uparrow} & n-i-3, n-i-5,\ldots,2,\\ r_{i+1}^{\downarrow}, r_{i+3}^{\downarrow}, \ldots, r_{n-2}^{\downarrow} & n-i-2, n-i-4,\ldots,1,\\ r_2^{\uparrow}, r_4^{\uparrow}, \ldots, r_{i-1}^{\uparrow} & 2,4,\ldots,i-1. \end{array}$$

We have $d_n(l_i^{\uparrow}) = n-i-1$, so we are interested in the number of combinations of the above independently flippable diagonals with exactly $k-(n-i-1)$ vertical

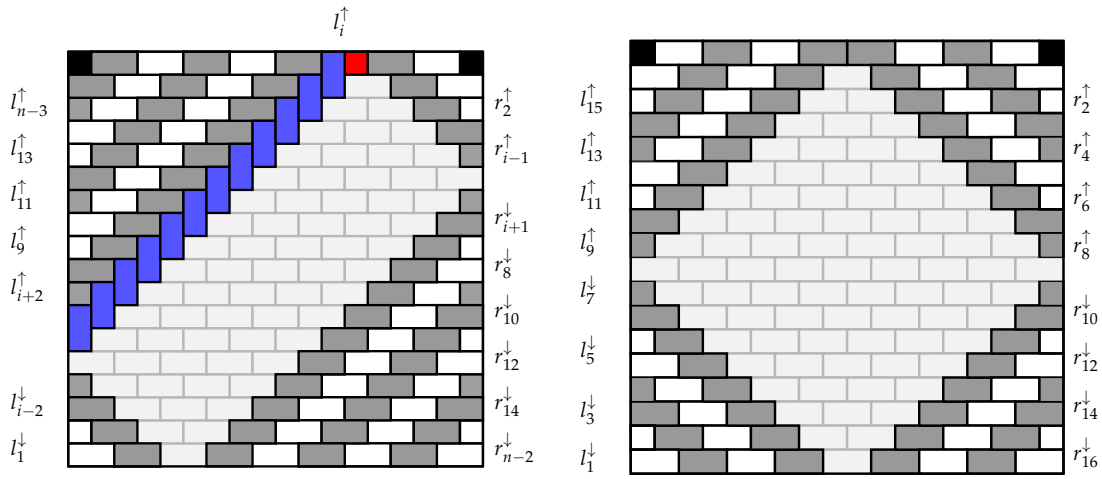


Figure 4.3: Allowable diagonals shown in alternating grey and white, *(a)*, for $\mathbf{T}_n(l_i)$, where $(n,i) = (18,5)$, and *(b)*, for $\mathbf{T}_{18}(\varnothing)$.

dominoes. That number is

$$\sum_{\substack{k_1+k_2= \\ k-(n-i-1)}} S(n-i-2,k_1)S(i-1,k_2).$$

The indices of the diagonals $l_i^{\uparrow}$ for which $d_n(l_i^{\uparrow}) \geqslant d_n(l_i^{\downarrow})$ and $r_i^{\downarrow}$ for which $d_n(r_i^{\downarrow}) \geqslant d_n(r_i^{\uparrow})$, range from 1 to $\left\lfloor \frac{n-1}{2} \right\rfloor$, as required for (4.8a).

Now suppose $a = \varnothing$. If i is the largest index such that $d_n(l_i^{\downarrow}) < d_n(l_i^{\uparrow})$ and j is the largest index such that $d_n(r_j^{\uparrow}) < d_n(r_j^{\downarrow})$, then $\max(i,j) = \lfloor \frac{n-2}{2} \rfloor$ and $|i-j| = 1$. The allowable diagonals in $\mathbf{T}_n(\varnothing)$ and their respective sizes are shown in the table below (see Figure 4.3(b)).

$l_1^{\downarrow}, l_3^{\downarrow}, \ldots, l_i^{\downarrow}$	$1,3,\ldots,i$
$l_{i+2}^{\uparrow}, l_{i+4}^{\uparrow}, \ldots, l_{n-3}^{\uparrow}$	$n-i-3, n-i-5, \ldots, 2$
$r_2^{\uparrow}, r_4^{\uparrow}, \ldots, r_j^{\uparrow}$	$2,4,\ldots,j,$
$r_{j+2}^{\downarrow}, r_{j+4}^{\downarrow}, \ldots, r_{n-2}^{\downarrow}$	$n-j-3, n-j-5, \ldots, 1.$

Choosing subsets of the independently flippable diagonals with k vertical dominoes contributes the term

$$\sum_{k_1+k_2=k} S\left(\left\lfloor \frac{n-2}{2} \right\rfloor, k_1\right) S\left(n - \left(\left\lfloor \frac{n-2}{2} \right\rfloor - 1\right) - 3, k_2\right),$$

$\mathbf{T}_n(a)$	Index and size of largest diagonal in this corner			
n even	$l_j^{\downarrow}$ (j odd)	$l_j^{\uparrow}$ (j odd)	$r_j^{\downarrow}$ (j even)	$r_j^{\uparrow}$ (j even)
$\mathbf{T}_n(l_i^{\uparrow})$	$j < i$	$j > i$	$j > i$	Type 2
index j:	$i-2$	$i+2$	$i+1$	$i-1$
size:	$i-2$	$n-i-3$	$n-i-2$	$i-1$
$\mathbf{T}_n(r_i^{\downarrow})$	Type 2	$i < j$	$j > i$	$j < i$
index j:	$i-1$	$i+1$	$i+2$	$i-2$
size:	$i-1$	$n-i-2$	$n-i-3$	$i-2$
$\mathbf{T}_n(l_i^{\downarrow})$		Symmetric with $\mathbf{T}_n(r_i^{\downarrow})$.		
$\mathbf{T}_n(r_i^{\uparrow})$		Symmetric with $\mathbf{T}_n(l_i^{\uparrow})$.		
n odd	$t_j^{\leftarrow}$ (j even)	$t_j^{\rightarrow}$ (j even)	$b_j^{\leftarrow}$ (j odd)	$b_j^{\rightarrow}$ (j odd)
$\mathbf{T}_n(t_i^{\rightarrow})$	$j < i$	$j > i$	$j < n-i-1$	Type 2
index j:	$i-2$	$i+2$	$n-i-2$	$n-i$
size:	$i-2$	$n-i-3$	$n-i-2$	$i-1$
$\mathbf{T}_n(b_i^{\rightarrow})$	$j < n-i-1$	Type 2	$j < i$	$j > i$
index j:	$n-i-2$	$n-i$	$i-2$	$i+2$
size:	$n-i-2$	$i-1$	$i-2$	$n-i-3$
$\mathbf{T}_n(t_i^{\leftarrow})$		Symmetric with $\mathbf{T}_n(t_i^{\rightarrow})$.		
$\mathbf{T}_n(b_i^{\leftarrow})$		Symmetric with $\mathbf{T}_n(b_i^{\rightarrow})$.		

Table 4.2: The longest allowable diagonals in each of four corners for each $\mathbf{T}_n(a)$. Entries are calculated using the parity of i and j, the avoidance of conflicts, and the requirement that a be the longest diagonal in $\mathbf{T}_n(a)$. Recall that conflict Type 2 occurs between diagonals a and b iff $d_n(a) + d_n(b) \geqslant n$.

and since $n - (\lfloor (n-2)/2 \rfloor - 1) - 3 = \lfloor (n-2)/2 \rfloor$, this is equal to (4.8b) for even n.

Case n odd: The bond covering is a vertical bond with $\lfloor (n-2)/2 \rfloor$ monominoes at the top (besides the two that are fixed) and $\lceil (n-2)/2 \rceil$ non-fixed monominoes along the bottom boundary. When diagonal a is flipped, $d_n(a)$ horizontal dominoes are added to the covering, instead of vertical dominoes. Hence we argue for $H(n,k)$ rather than $V(n,k)$.

Now $t_j^{\leftarrow}$ and $t_j^{\rightarrow}$ have even length, and $b_j^{\leftarrow}$ and $b_j^{\rightarrow}$ have odd length (see Table 4.2). For example, the allowable diagonals in $\mathbf{T}_n(t_i^{\rightarrow})$ are shown in Figure 4.4(a) (for $(n,i) = (17,6)$), and their respective lengths are

$$\begin{array}{ll} t_1^{\leftarrow}, t_3^{\leftarrow}, \ldots, t_{i-2}^{\leftarrow} & 1, 3, \ldots, i-2, \\ t_{i+2}^{\rightarrow}, t_{i+4}^{\rightarrow}, \ldots, t_{n-3}^{\rightarrow} & n-i-3, n-i-5, \ldots, 2, \\ b_1^{\leftarrow}, b_3^{\leftarrow}, \ldots, b_{n-i-2}^{\leftarrow} & 1, 3, \ldots, n-i-2, \\ b_{n-i}^{\rightarrow}, b_{n-i+2}^{\rightarrow}, \ldots, b_{n-2}^{\rightarrow} & i-1, i-3, \ldots, 1. \end{array}$$

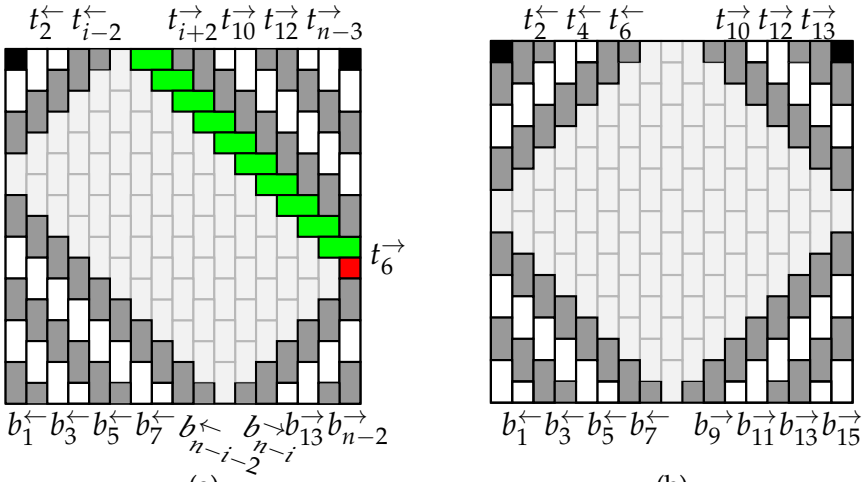


Figure 4.4: Allowable diagonals shown in alternating grey and white, *(a)*, for $\mathbf{T}_n(t_i)$, where $(n,i) = (17,6)$, and *(b)*, for $\mathbf{T}_{17}(\varnothing)$.

Once again $d_n(t_i^{\rightarrow}) = n-i-1$, so we are interested in the number of combinations of the above independently flippable diagonals with exactly $k-(n-i-1)$ horizontal dominoes. As before, that number is

$$\sum_{\substack{k_1+k_2=\\ k-(n-i-1)}} S(n-i-2,k_1)S(i-1,k_2).$$

Now suppose $a = \varnothing$, then if i is the largest index such that $d_n(t_i^{\rightarrow}) < d_n(t_i^{\leftarrow})$ and j is the largest index such that $d_n(b_j^{\rightarrow}) < d_n(b_j^{\leftarrow})$ then $\max(i,j) = \lfloor \frac{n-2}{2} \rfloor$ and $|i-j| = 1$. The allowable leftward diagonals in $\mathbf{T}_n(\varnothing)$ and their respective sizes are given in the table below.

$$\begin{array}{ll} t_2^{\leftarrow}, t_4^{\leftarrow}, \ldots, t_j^{\leftarrow} & 2,4,\ldots,j, \\ b_1^{\leftarrow}, b_3^{\leftarrow}, \ldots, b_i^{\leftarrow} & 1,3,\ldots,i \end{array}$$

and by horizontal symmetry, the rightward diagonals have the same lengths. We conclude that the coverings with k horizontal dominoes of $\mathbf{T}_n(\varnothing)$ is also generated by (4.8b) when n is odd. □

The terms $\mathcal{VH}(n,k)z^k$ can be summed over k to obtain the generating polynomial $T(n,z)$ (same as $\mathcal{VH}_n(z)$), mentioned in Conjecture 4 of [12].

Theorem 4.2. *Let* $\mathcal{VH}_n(z) = \sum_{k\geqslant 0} \mathcal{VH}(n,k)z^k$. *We have*

$$\mathcal{VH}_n(z) := 2\sum_{i=1}^{\lfloor\frac{n-1}{2}\rfloor} S_{n-i-2}(z)S_{i-1}(z)z^{n-i-1} + \left(S_{\lfloor\frac{n-2}{2}\rfloor}(z)\right)^2, \tag{4.9}$$

where $S_n(z) = \sum_{k\in\mathbb{Z}} S(n,k)z^k$. *This "generates" $V(n,k)$ for even n, and $H(n,k)$ for odd n.*

Proof. The details in this proof are used to prove Theorem 3.6 again. We obtain $\mathcal{VH}_n(z)$ by simplifying the sum $\sum_{k\in\mathbb{Z}} \mathcal{VH}(n,k)z^k$ to

$$\sum_{k\in\mathbb{Z}} \left(2\sum_{i=1}^{\lfloor\frac{n-1}{2}\rfloor} \left(\sum_{\substack{k_1+k_2=\\ k-(n-i-1)}} S(n-i-2,k_1)S(i-1,k_2) \right) z^k \right)$$
$$+ \sum_{k\in\mathbb{Z}} \left(\sum_{k_1+k_2=k} S\left(\left\lfloor\frac{n-2}{2}\right\rfloor,k_1\right) S\left(\left\lfloor\frac{n-2}{2}\right\rfloor,k_2\right)\right) z^k.$$

A small adjustment gives

$$2\sum_{i=1}^{\lfloor\frac{n-1}{2}\rfloor} \left(z^{n-i-1} \sum_{k\in\mathbb{Z}} \left(\sum_{k_1+k_2=k} S(n-i-2,k_1)z^{k_1}S(i-1,k_2)z^{k_2} \right)\right) \tag{4.10}$$
$$+ \sum_{k\in\mathbb{Z}} \left(\sum_{k_1+k_2=k} S\left(\left\lfloor\frac{n-2}{2}\right\rfloor,k_1\right) z^{k_1} S\left(\left\lfloor\frac{n-2}{2}\right\rfloor,k_2\right) z^{k_2}\right), \tag{4.11}$$

Recall that for each k and i, the term $S(n-i-2,k_1)S(i-1,k_2)$ counts some of the coverings in $\mathbf{T}_n(a)$, for some diagonal, a. Here, $d(a) = n-i-1$, and all of the coverings that are counted have exactly k vertical (or horizontal, if n is odd) dominoes. If i is fixed, the sum of these over all k is $|\mathbf{T}_n(a)|$. We easily obtain Equation (4.9) from Equations (4.10) and (4.11). □

Theorem 3.6 follows from the proof of Theorem 4.2 in a way that provides a natural partition of $n2^{n-1}$ into equivalence classes, which solves an implied question by Don Knuth, noted in [13].

Proof of Theorem 3.6. The comments in the above proof reveal that $S_{n-i-2}(1)S_{i-1}(1) = |\mathbf{T}_n(a)|$, for some diagonal a with $d(a) = n-i-1$. Thus we have $2\left\lfloor\frac{n-1}{2}\right\rfloor$ sets of coverings from $|\mathbf{T}_n(a)| = 2^{n-3}$, and one more from the last

term, which gives $|\mathbf{T}_n(\varnothing)| = 2^{2\left\lfloor \frac{n-2}{2} \right\rfloor}$. Even and odd cases appear again when these floors are evaluated.

If n is odd there are $n-1$ classes of type $\mathbf{T}_n(a)$, and the last one is obtained from $|\mathbf{T}_n(\varnothing)| = 2^{n-3}$. This is the partition we desire, since we are only considering coverings of a single orientation.

If n is even there are only $n-2$ classes of type $\mathbf{T}_n(a)$, but $|\mathbf{T}_n(\varnothing)| = 2^{n-2}$. We arbitrarily divide $|\mathbf{T}_n(\varnothing)|$ into halves to obtain the last two classes of size 2^{n-3} that are required. □

The degree of $\mathcal{VH}_n(z)$ is $\frac{n^2-n}{2} - (n-1)$, because this is the largest number of vertical dominoes possible in a covering of $\mathbf{T}_n$, for even n (and horizontal dominoes for odd n). For example, the covering with all l_i flipped up and all r_i flipped down contains exactly $n-1$ horizontal dominoes.

The coefficients of $\mathcal{VH}_n(z)$ are listed in Table 4.3 up to $n = 10$, and the following conjecture is true at least up to $n = 20$. If $Q(z)$ is a polynomial, then write $\langle z^k \rangle Q(z)$ to denote the coefficient of z^k.

Conjecture 4.3.

(a) For $k \leqslant n-2$, we have $\langle z^k \rangle \mathcal{VH}_n(z) = \langle z^k \rangle \prod_{m \geqslant 0} (1+z^m)^2$, the number of partitions of k into distinct parts with two types of each part (see A022567 in [27]).

(b) For $0 \leqslant k < n-3$, we have

$$\langle z^{\deg(\mathcal{VH}_n(z))-k} \rangle \mathcal{VH}_n(z) = 2\langle z^k \rangle \prod_{m \geqslant 0} (1+z^m),$$

twice the number of partitions of k into distinct parts (see A000009 in [27]).

Rotating a covering of $\mathbf{T}_n$ by $\pi/2$ radians interchanges vertical and horizontal dominoes, and this transformation can be applied to the generating polynomial $\mathcal{VH}_n(z)$ to obtain the polynomial $\mathcal{VH}_n(z^{-1})z^{n(n-1)/2}$. Thus we can easily derive the bivariate generating polynomial $R_n(x,y)$, whose coefficient of $x^v y^h$ is the number of tatami coverings with exactly v vertical dominoes and h horizontal dominoes.

Our remarks along with some small algebraic manipulations prove the following corollary.

Corollary 4.4. *Let $R_n(x,y)$ be as defined above. We have*

$$R_n(x,y) = 2\mathcal{VH}_n(xy^{-1})y^{\frac{n^2-n}{2}} + 2\mathcal{VH}_n(x^{-1}y)x^{\frac{n^2-n}{2}}. \tag{4.12}$$

$n\backslash z^k$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	1															
3	1	2														
4	1	2	3	2												
5	1	2	3	6	4	2	2									
6	1	2	3	6	9	8	7	6	2	2	2					
7	1	2	3	6	9	14	15	14	14	10	8	6	4	2	2	2
8	1	2	3	6	9	14	22	24	25	28	25	22	19	14	10	10
9	1	2	3	6	9	14	22	32	37	42	49	48	49	46	38	34
10	1	2	3	6	9	14	22	32	46	56	66	78	84	90	92	88
$n\backslash z^k$	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
8	8	4	4	2	2	2										
9	30	24	20	16	12	12	10	6	4	4	2	2	2			
10	81	76	69	58	51	44	38	34	28	22	20	16	14	12	8	6
$n\backslash z^k$	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
10	4	4	2	2	2											

Table 4.3: Table of coefficients of $\mathcal{VH}_n(z)$ for $2 \leqslant n \leqslant 10$. The (n,k)th entry represents the number of coverings of $\mathbf{T}_n$ with k vertical dominoes when n is even, and k horizontal dominoes when n is odd. See Table A.20 for larger values of n.

We list some basic properties of $R_n(x,y)$.

- The degree of $R_n(x,1)$ as well as the degree of every term in $R_n(x,y)$ is $\frac{n^2-2}{2}$;
- coefficients of $R_n(x,1)$ are given in Table A.21, for small n,
- the polynomial $R_n(x,y)$ can be recovered from $R_n(x,1)$, and the latter is the generating polynomial for the set of all $n \times n$ coverings with n monominoes with exactly v vertical dominoes (or h horizontal dominoes);
- the polynomial $R_n(x,1)$ is self reciprocal because of interchangeability of vertical and horizontal dominoes; and finally,
- the polynomial $R_n(x,1)$ has similar properties to those listed for $\mathcal{VH}_n(z)$ in Conjecture 4.3, in the sense that for some increasing integer function f, we have $\langle x^k \rangle R_n(x,1) = \langle x^k \rangle R_{n+1}(x,1)$, whenever $k < f(n+1)$.

If there is an even number of dominoes, which is the case when $(n^2-n)/4$ is an integer, then $\langle x^k y^k \rangle R_n(x,y) = 4\langle z^k \rangle \mathcal{VH}_n(z)$, where $k = (n^2-n)/4$. Rotating the covering maps k vertical dominoes to k horizontal dominoes, and vice versa. The coverings counted by these coefficients are called *balanced tatami coverings*, appropriately named by Knuth (private communication), because the number of vertical and horizontal dominoes are equal. Here is $\langle z^k \rangle \mathcal{VH}_n(z)$ for $2 \leqslant n \leqslant 56$: 0, 0, 2, 2, 0, 0, 10, 20, 0, 0, 114, 210, 0, 0, 1322, 2460, 0, 0, 16428, 31122, 0, 0, 214660, 410378, 0, 0, 2897424, 5575682, 0, 0, 40046134, 77445152, 0, 0, 563527294, 1093987598, 0, 0, 8042361426, 15660579168, 0, 0, 116083167058, 226608224226, 0, 0, 1691193906828, 3308255447206, 0, 0, 24830916046462, 48658330768786, 0, 0, 366990100477712, (see

A182107 in [27]). Note that this is perhaps better viewed as four sequences, one for each $0 \leqslant j < 4$ such that $n \pmod 4 = j$.

4.3 A mysterious factor of $\mathit{VH}_n(z)$

In this section we prove that the generating polynomial $\mathit{VH}_n(z)$ has (very nearly) the factorisation conjectured in [12]. We use the following lemma.

Lemma 4.5. *For all $x \geqslant 0$,*

$$\lfloor x \rfloor = \sum_{k \geqslant 1} \left\lfloor \frac{x}{2^k} + \frac{1}{2} \right\rfloor. \tag{4.13}$$

Proof. Let $n = \lfloor x \rfloor$ and apply strong induction on n. Clearly Equation (4.13) holds for the base case, when $n = 0$. Suppose it holds for $0, 1, \ldots, n-1$, then

$$\begin{aligned}\sum_{k \geqslant 1} \left\lfloor \frac{x}{2^k} + \frac{1}{2} \right\rfloor &= \left\lfloor \frac{x}{2} + \frac{1}{2} \right\rfloor + \sum_{k \geqslant 1} \left\lfloor \frac{\frac{x}{2}}{2^k} + \frac{1}{2} \right\rfloor \\ &= \left\lfloor \frac{x}{2} + \frac{1}{2} \right\rfloor + \left\lfloor \frac{x}{2} \right\rfloor = \left\lfloor \frac{\lfloor x \rfloor}{2} + \frac{1}{2} \right\rfloor + \left\lfloor \frac{\lfloor x \rfloor}{2} \right\rfloor = \lfloor x \rfloor.\end{aligned}$$

Two applications of Equation (3.11) in [15] yield the penultimate equation and the final equation follows by considering the parity of $\lfloor x \rfloor$, or by using (3.26) in [15] with $m = 2$ and $x' = x/2$. □

Theorem 4.6. *The generating polynomial $\mathit{VH}_n(z)$ has the factorisation*

$$\mathit{VH}_n(z) = P_n(z) D_n(z)$$

where $P_n(z)$ is a polynomial and

$$D_n(z) = \prod_{j \geqslant 1} S_{\left\lfloor \frac{n-2}{2^j} \right\rfloor}(z). \tag{4.14}$$

Proof. We prove that $D_n(z)$ divides $\mathit{VH}_n(z)$ by using the factorisation of $S_n(z)$ into cyclotomic polynomials ([12], Lemma 5),

$$S_n(z) = \prod_{j \geqslant 1} \Phi_{2j}(z)^{\left\lfloor \frac{n+j}{2j} \right\rfloor}, \tag{4.15}$$

and showing that the power of $\Phi_i(z)$ is greater in each term of $VH_n(z)$ than it is in $D_n(z)$.

The power of $\Phi_{2j}(z)$ in $D_n(z)$ is obtained by substituting Equation (4.15) into Equation (4.14):

$$\begin{aligned} D_n(z) = \prod_{i\geqslant 1} S_{\left\lfloor \frac{n-2}{2^i} \right\rfloor}(z) &= \prod_{i\geqslant 1, j\geqslant 1} \Phi_{2j}(z)^{\left\lfloor \frac{\left\lfloor \frac{n-2}{2^i} \right\rfloor + j}{2j} \right\rfloor} \\ &= \prod_{j\geqslant 1} \Phi_{2j}(z)^{\sum_{i\geqslant 1} \left\lfloor \frac{\left\lfloor \frac{n-2}{2^i} \right\rfloor + j}{2j} \right\rfloor}. \end{aligned}$$

We simplify $D_n(z)$ to

$$D_n(z) = \prod_{j\geqslant 1} \Phi_{2j}(z)^{\left\lfloor \frac{n-2}{2j} \right\rfloor} \tag{4.16}$$

by applying Lemma 4.5 and with Equation (3.11) in [15].

Expanding the second term of $VH_n(z)$ gives

$$\left(S_{\frac{n-2}{2}}(z)\right)^2 = \prod_{j\geqslant 1} \Phi_{2j}(z)^{2\left\lfloor \frac{\frac{n-2}{2}+j}{2j} \right\rfloor},$$

which is divisible by $D_n(z)$ since

$$\left\lfloor \frac{n-2}{2j} \right\rfloor \leqslant 2 \left\lfloor \frac{\frac{n-2}{2}+j}{2j} \right\rfloor$$

for all $j \geqslant 1$ and positive even integers n.

The other terms in $V_n(z)$ are of the form

$$S_{n-k-2}(z) S_{k-1}(z) z^d = \left(\prod_{j>0} \Phi_{2j}(z)^{\left\lfloor \frac{(n-k-2)+j}{2j} \right\rfloor} \right) \left(\prod_{j>0} \Phi_{2j}(z)^{\left\lfloor \frac{(k-1)+j}{2j} \right\rfloor} \right) z^d$$

for each $1 \leqslant k \leqslant \left\lfloor \frac{n-1}{2} \right\rfloor$ where d is the appropriate power of z. These terms are

all divisible by $D_n(z)$ if the exponents in Equation (4.16) satisfy

$$\left\lfloor \frac{n-2}{2j} \right\rfloor \leqslant \left\lfloor \frac{k-1}{2j} + \frac{1}{2} \right\rfloor + \left\lfloor \frac{n-k-2}{2j} + \frac{1}{2} \right\rfloor. \tag{4.17}$$

Let r_1 and r_2 be integers such that $0 \leqslant r_i < 2j$ and $\frac{k-1}{2j} = \left\lfloor \frac{k-1}{2j} \right\rfloor + \frac{r_1}{2j}$ and $\frac{n-2}{2j} = \left\lfloor \frac{n-2}{2j} \right\rfloor + \frac{r_2}{2j}$. We eliminate occurrences of $\left\lfloor \frac{k-1}{2j} \right\rfloor$ and $\left\lfloor \frac{n-2}{2j} \right\rfloor$ from Inequality (4.17) since they are integers and can be removed from floors, and rewrite the inequality as

$$0 \leqslant \left\lfloor \frac{r_1}{2j} + \frac{1}{2} \right\rfloor + \left\lfloor \frac{r_2 - r_1 - 1}{2j} + \frac{1}{2} \right\rfloor. \tag{4.18}$$

It is straightforward to show that if the second term is -1, then the first term is equal to 1.

Therefore, $D_n(z)$ divides each and every term of $\mathcal{VH}_n(z)$. □

$n\backslash z^k$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	1	2														
4	1	1	2													
5	1	1	2	4	0	2										
6	1	0	1	2	2	-2	2									
7	1	0	1	2	2	4	-2	4	0	2	-2	2				
8	1	0	1	1	2	3	4	-2	2	0	4	-2	2	-2	2	
9	1	0	1	1	2	3	4	6	-2	6	0	8	-2	4	-4	6
10	1	-1	1	0	1	1	1	2	2	-6	6	-2	6	-6	4	-4
11	1	-1	1	0	1	1	1	2	2	4	-8	10	-4	10	-8	8
$n\backslash z^k$	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
9	-2	4	-2	2	-2	2										
10	6	-6	6	-4	4	-4	2									
11	-8	10	-10	12	-8	10	-12	10	-6	6	-6	6	-4	4	-4	2

Table 4.4: Table of coefficients of $P_n(z)$ for $3 \leqslant n \leqslant 11$. See Table A.22 for larger values of n.

Our computer investigations show that $P_n(z)$ is irreducible for $1 < n < 200$, and we know the complete factorisation of $S_k(z)$, for each positive integer k. We suspect, therefore, that the complete factorisation is

$$\mathcal{VH}_n(z) = P_n(z) \prod_{j \geqslant 1} \Phi_{2j}(z)^{\left\lfloor \frac{n-2}{2j} \right\rfloor}. \tag{4.19}$$

The factor $P_n(z)$ is somewhat more mysterious than $D_n(z)$; e.g., we have no formula to express it besides $\mathcal{VH}_n(z)/D_n(z)$. Take $P_{11}(z)$ for example, which is equal to $1 - 1z^1 + 1z^2 + 0z^3 + 1z^4 + 1z^5 + 1z^6 + 2z^7 + 2z^8 + 4z^9 - 8z^{10} + 10z^{11} - 4z^{12} + 10z^{13} - 8z^{14} + 8z^{15} - 8z^{16} + 10z^{17} - 10z^{18} + 12z^{19} - 8z^{20} + 10z^{21} - 12z^{22} + 10z^{23} - 6z^{24} + 6z^{25} - 6z^{26} + 6z^{27} - 4z^{28} + 4z^{29} - 4z^{30} + 2z^{31}$. The coefficients are almost all non-zero, a great many of them are even, they alternate in sign for a

long stretch, the central coefficients are larger than the ones at the tails, and the polynomial is irreducible.

The degree of $P_{11}(z)$ is $\deg(P_{11}(z)) = \deg(WH_{11}(z)) - \deg(D_{11}(z))$, both of which are easily calculated since $\deg(WH_n(z)) = \frac{n^2-n}{2} - (n-1)$, and $\deg(S_n(z)) = \binom{n+1}{2}$ gives a sum for the degree of $D_n(z)$ (see proof of Theorem 4.7, below). In general $\deg(P_n(z))$ is equal to the sum of the sequence of largest odd divisors of the numbers $1, 2, \ldots, n-2$, which is a sequence with some nice properties (see A135013 in [27]).

Theorem 4.7. *For each $n \geqslant 2$,*

$$\deg(P_n(z)) = \sum_{k=1}^{n-2} Od(k),$$

where $Od(k)$ is the largest odd divisor of k.

Proof. The degree of $D_n(z)$ is the sum of the degrees of its factors, given in Equation (4.14), so we can write

$$\deg(P_n(z)) = \deg(WH_n(z)) - \deg(D_n(z)) \tag{4.20}$$

$$= \binom{n-1}{2} - \sum_{k\geqslant 1} \binom{\left\lfloor \frac{n-2}{2^k} \right\rfloor + 1}{2} \tag{4.21}$$

since $\deg(S_n(z)) = \binom{n+1}{2}$.

The proof that $\sum_{k=1}^{n} Od(k) = \deg(P_{n+2}(z))$ is by induction, and the base case, where $n = 0$, is easily verified. Let $p_n = \deg(P_n(z))$, for $n \geqslant 2$, to abbreviate the notation. It remains for us to show that $p_{n+3} - p_{n+2} = Od(n+1)$.

Let $n'01^\alpha$ be the binary representation of n, so that $(n+1)_2 = n'10^\alpha$, and let $[\![A]\!] = 1$ if the statement A is true, and $[\![A]\!] = 0$ otherwise. Observe that

$$\left\lfloor \frac{n+1}{2^k} \right\rfloor = [\![k \leqslant \alpha]\!] + \left\lfloor \frac{n}{2^k} \right\rfloor \tag{4.22}$$

which we use to simplify

$$\sum_{k\geqslant 1} \left(\binom{\left\lfloor \frac{n+1}{2^k} \right\rfloor + 1}{2} - \binom{\left\lfloor \frac{n}{2^k} \right\rfloor + 1}{2} \right)$$

and write

$$p_{n+3} - p_{n+2} = (n+1) - \sum_{k=1}^{\alpha} \left(\left\lfloor \frac{n}{2^k} \right\rfloor + 1 \right).$$

Using Equation (4.22) and the fact that $(n+1)/2^k$ is an integer for $1 \leqslant k \leqslant \alpha$, we write

$$p_{n+3} - p_{n+2} = (n+1) - \sum_{k=1}^{\alpha} \left(\left\lfloor \frac{n+1}{2^k} + \frac{1}{2} \right\rfloor \right),$$

and then express this as the remaining sum terms in Equation (4.13)

$$\begin{aligned} p_{n+3} - p_{n+2} &= \sum_{k \geqslant \alpha+1} \left(\left\lfloor \frac{n+1}{2^k} + \frac{1}{2} \right\rfloor \right) \\ &= \sum_{k-\alpha \geqslant 1} \left(\left\lfloor \frac{\frac{n+1}{2^\alpha}}{2^{k-\alpha}} + \frac{1}{2} \right\rfloor \right). \end{aligned}$$

Applying Equation (4.13) again, we have $p_{n+3} - p_{n+2} = (n+1)/2^\alpha$, which is equal to $Od(n+1)$, as required. □

In addition to finding $\deg(P_{11}(z))$, we can evaluate at $z = 1$ with $P_{11}(1) = \mathcal{VH}_{11}(1)/D_{11}(1) = 22$, a ratio which is also easy to calculate in general because $\mathcal{VH}_n(1)$ and $D_n(1)$ have well understood combinatorial interpretations. It also leads to an interesting sequence, whose derivation for all n is given below.

Theorem 4.8. *The sum of the coefficients of $P_n(z)$ is equal to $n2^{\nu(n-2)-1}$, where $\nu(n)$ is the number of* 1*s in the binary representation of n*

Proof. The sum of the coefficients of $P_n(z)$ is equal to $P_n(1)$, which is expressible as $\mathcal{VH}_n(1)/D_n(1)$. The numerator evaluates to $n2^{n-3}$, since this is the number of coverings in $\mathbf{T}_n$, and the denominator is evaluated as described below.

It is well known that $\Phi_k(1) = p$ if k is a non-zero power of a prime p and $\Phi_k(1) = 1$ if k is divisible by two distinct primes (see [20], p.74). We can evaluate $D_n(1)$ using Equation (4.16),

$$D_n(1) = \prod_{i \geqslant 1} \Phi_{2i}(1)^{\left\lfloor \frac{n-2}{2i} \right\rfloor} = 2^{\sum_{i \geqslant 1} \left\lfloor \frac{n-2}{2^i} \right\rfloor},$$

by ignoring the factors for which $2i$ is not a power of 2. Apply Equation (4.24)

in [15] to obtain $D_n(1) = 2^{n-2-\nu(n-2)}$. Thus

$$P_n(1) = n2^{n-3-(n-2)+\nu(n-2)} = n2^{\nu(n-2)-1}.$$

□

We have verified that $P_n(z)$ is irreducible over the integers for $1 < n < 200$, but we do not understand its structure well enough to prove it for all n. We state below some of the observable structure which has also been verified for $1 < n < 200$, as Conjecture 4.9, and we plot some complex roots for odd n up to 67 in Figure 4.5.

Conjecture 4.9.

(a) If $k \geqslant 1$ and $n \pmod{2^k} = 2$, then $\langle z^i \rangle P_n(z) = \langle z^i \rangle P_{n+j}(z)$ for $i \leqslant \frac{n-2}{2^{k-1}}$ and $j \leqslant 2^k$.

(b) When n is odd, $P_n(z)$ has exactly one real root α_n, with $-1 < \alpha_n \leqslant -0.5$, and $\{\alpha_n\}_{n\ odd}$ is a monotonically decreasing sequence.

(c) When n is even, $P_n(z)$ has no real root.

(d) The polynomial $P_n(z)$ is irreducible over the integers for $n \geqslant 2$.

(e) The alternating sums of coefficients are given by the generating function

$$\sum_{n \geqslant 2} P_n(-1) z^{n-2} = \frac{(1+z)(1-2z)}{(1-2z^2)\sqrt{1-4z^2}}. \tag{4.23}$$

(f) For even n, the sum of the absolute values of coefficients of $P_n(z)$ is equal to $P_n(-1)$ when $n \geqslant 20$.

The right hand side of Equation (4.23) is the sum of two generating functions, with odd and even powered terms, respectively. The sequence of coefficients of the odd power terms is $-\sum_{i=0}^{k} 2^{k-i}\binom{2i}{i}$, for $k \geqslant 0$ (see A082590 in [27]), and that of the even power terms is $\binom{2k}{2}$, for $k \geqslant 1$ (see A000984 in [27]). The first few numbers $P_n(-1)$, starting with $n = 2$, are: 1, −1, 2, −4, 6, −14, 20, −48, 70, −166, 252, −584, 924, −2092, 3432, −7616, 12870, −28102, 48620, −104824, 184756, −394404, 705432, −1494240, 2704156, −5692636, 10400600.

Conjecture 4.9(f) compares the above sequence with the sum of the absolute values of the coefficients of $P_n(z)$. The first few of these are listed, also starting

with $n = 2$: 1, 3, 4, 10, 10, 22, 28, 64, 76, 180, 260, 606, 932, 2124, 3440, 7666, 12872, 28178, 48620, 104946, 184756, 394638, 705432, 1494600, 2704156, 5693376, 10400600.

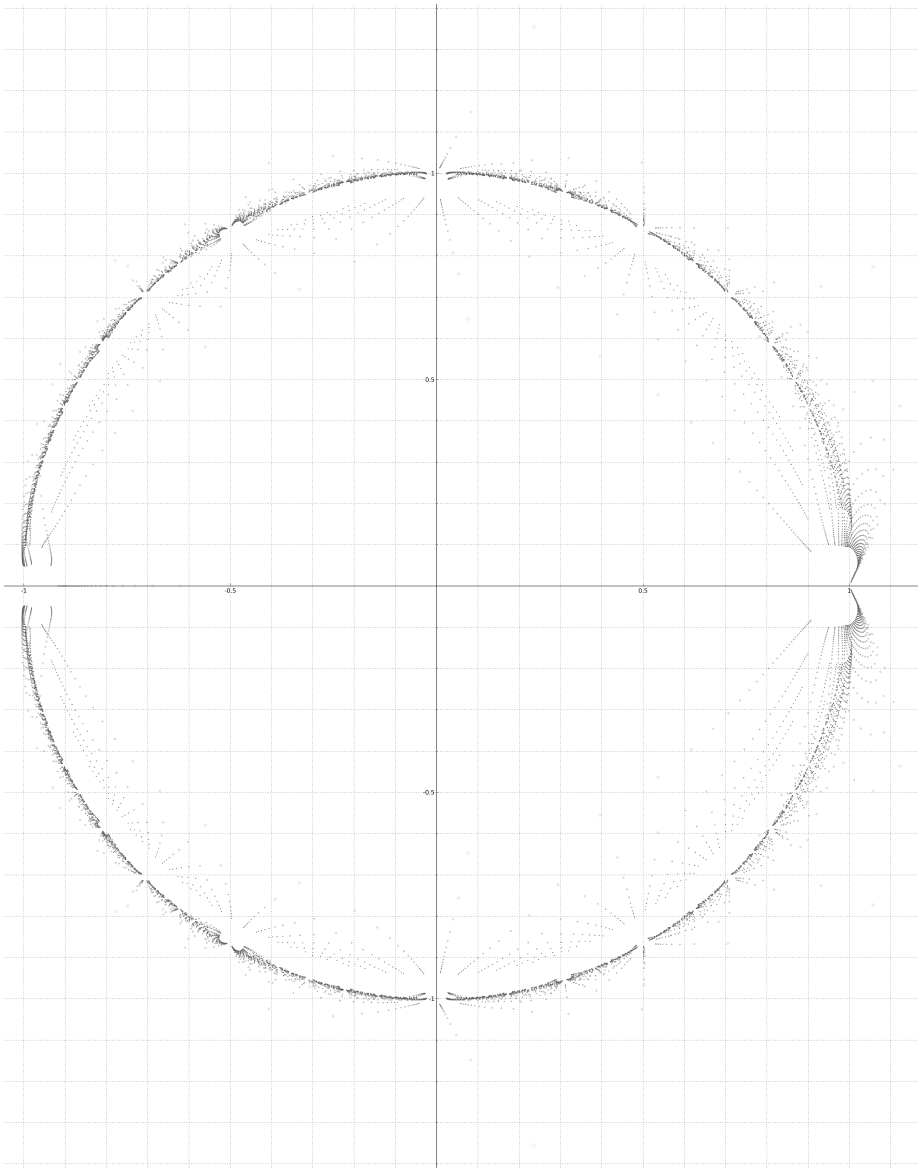


Figure 4.5: The complex zeros of $P_n(z)$ for odd n, where $3 \leqslant n \leqslant 67$. Darker and smaller points are used for larger n.

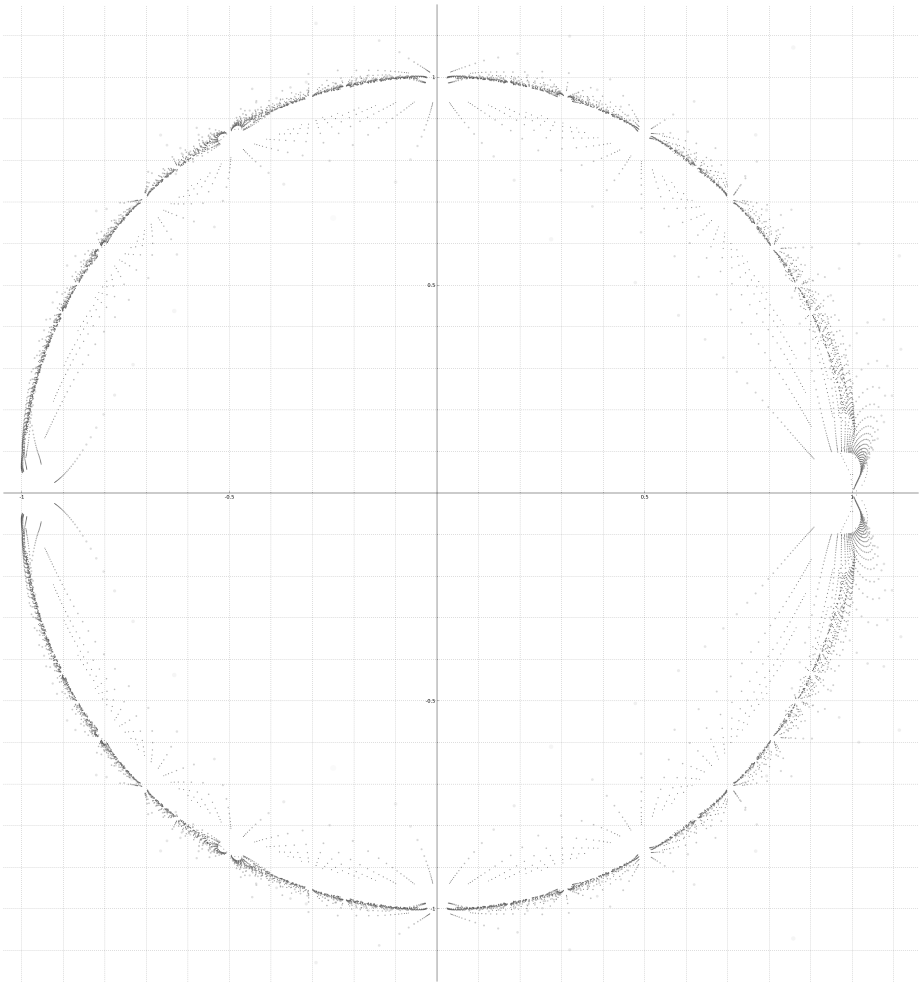


Figure 4.6: The complex zeros of $P_n(z)$ for even n, where $4 \leqslant n \leqslant 68$. Darker and smaller points are used for larger n.

Chapter 5

Combinatorial generation of tatami coverings

Our efforts to enumerate tatami coverings reveal algorithms for generating them. We show how to list all of the coverings in $\mathbf{T}_n$, as well as those with k vertical dominoes, by using the partition in Section 4.1.1. We also generate equivalence classes of coverings of two-way infinitely wide, horizontal strip, in Section 5.3. All of these algorithms run in constant amortised time per covering.

5.1 Coverings of the $n \times n$ grid with n monominoes

In this section we describe two combinatorial algorithms for generating all elements of $\mathbf{T}_n$. Each algorithm is a natural extension of our proof of Lemma 4.1, and we use the notation described in Section 4.1.

Firstly, consider the alternate proof of Theorem 3.6, given on page 46 (the original proof leads to another, recursive algorithm). Each class, $\mathbf{T}_n(a)$, of the partition contains k independently flippable diagonals (see Section 4.1.1 and Figures 4.3-4.4), and therefore generating the coverings of $\mathbf{T}_n(a)$ is equivalent to generating all subsets of k, for which there exist various constant amortised time (CAT) algorithms. Table 4.2 contains constant-time calculations which provide the required set of flippable diagonals. Recall, for example, the flippable diagonals for $\mathbf{T}_{18}(l_5^{\uparrow})$, which are given on page 42. We conclude that there is a CAT algorithm which generates all $n \times n$ tatami coverings with n monominoes.

5.1.1 Gray code

Consider a Gray code whose operation is the diagonal flip. We provide a Gray code for $\mathbf{T}_n$.

Using the ternary representation given in Section 4.1, flipping a diagonal is equivalent to incrementing or decrementing a symbol. Let $A, B \in \mathbf{T}_n$, with ternary representations $A = a_1 a_2 \dots a_{n-2}$ and $B = b_1 b_2 \dots b_{n-2}$, and define the *distance*, H, between A and B as

$$H(A, B) = \sum_{i=1}^{n-2} |a_i - b_i|.$$

This is consistent with the Hamming distance between binary representations of A and B (see Section 4.1). Successive coverings, A and B, in our Gray code, satisfy $H(A, B) = 1$.

Recall that each class $\mathbf{T}_n(a)$ contains a set, A, of independently flippable diagonals and therefore generating the coverings of $\mathbf{T}_n(a)$ is equivalent to generating all subsets of A. For the sake of argument we use the binary reflected Gray code, defined in terms of listing all subsets of $\{a_1, a_2, \dots, a_n\}$.

Definition 5.1 (Binary Reflected Gray Code (BRGC) [16])**.** *Let G_n be the BRGC listing of the subsets of $\{a_1, a_2, \dots, a_n\}$, where the a_i are distinct, but otherwise arbitrary. Then G_n satisfies $G_1 = (\varnothing, \{a_1\})$, and $G_n = (g_1, g_2, \dots, g_k, \{a_n\} \cup g_k, \{a_n\} \cup g_{k-1}, \dots, \{a_n\} \cup g_1)$, where $(g_1, g_2, \dots, g_k) = G_{n-1}$.*

Let $T \in \mathbf{T}_n$ and let $\{x_1, x_2, \dots, x_k\}$ be the set of flipped diagonals in T. For convenience write $T = \{x_1, x_2, \dots, x_k\}$. Let A be the set, $\{a_1, a_2, \dots, a_{|A|}\}$, of flippable diagonals in $\mathbf{T}_n(a)$, where a is a diagonal or $a = \varnothing$. Let $\{\varnothing\} \cup g_i = g_i$, and let $a \cup G_{|A|} = (\{a\} \cup g_1, \{a\} \cup g_2, \dots, \{a\} \cup g_{2^{|A|}})$. The list $a \cup G_{|A|}$ is a Gray code for $\mathbf{T}_n(a)$.

A Gray code for $\mathbf{T}_n$ is obtained by concatenating the set of Gray codes, $a \cup G_{|A|}$, for each $\mathbf{T}_n(a)$. For even n, this list of equivalence classes begins with

$$\mathbf{T}_n(l_1^{\uparrow}), \mathbf{T}_n(r_2^{\downarrow}), \mathbf{T}_n(l_3^{\uparrow}), \mathbf{T}_n(r_4^{\downarrow}), \dots, \mathbf{T}_n(a^{(1)}), \tag{5.1}$$

where $a^{(1)} \in \{l_i^{\uparrow}, r_j^{\downarrow}\}$, and all of the classes of the form $\mathbf{T}_n(l_i^{\uparrow})$ and $\mathbf{T}_n(r_j^{\downarrow})$ are listed. The next class is $\mathbf{T}_n(\varnothing)$, followed by

$$\mathbf{T}_n(r_{n-2}^{\uparrow}), \mathbf{T}_n(l_{n-3}^{\downarrow}), \mathbf{T}_n(r_{n-4}^{\uparrow}), \mathbf{T}_n(l_{n-5}^{\downarrow}), \dots, \mathbf{T}_n(a^{(2)}), \tag{5.2}$$

where $a^{(2)} \in \{r_i^{\uparrow}, l_j^{\downarrow}\}$, and all of the classes of the form $\mathbf{T}_n(r_i^{\uparrow})$ and $\mathbf{T}_n(l_j^{\downarrow})$ are listed.

Notice that $r_{i+1}^{\downarrow}$ is flippable in $\mathbf{T}_n(l_i^{\uparrow})$, and $l_{i+1}^{\uparrow}$ is flippable in $\mathbf{T}_n(r_i^{\downarrow})$. Let A be the set, $\{a_1, a_2, \ldots, a_{|A|}\}$, of flippable diagonals in $\mathbf{T}_n(a)$, letting $a_{|A|} = r_{i+1}^{\downarrow}$ if $a = l_i^{\uparrow}$, and $a_{|A|} = l_{i+1}^{\uparrow}$ if $a = r_i^{\downarrow}$, so that the last covering listed in $\mathbf{T}_n(a)$ is $\{a, a_{|A|}\}$. This is one flip away from $\{a_{|A|}\}$, which is the first element listed in the next equivalence class. If $a = a^{(1)}$ it is $\mathbf{T}_n(\varnothing)$, and otherwise it is $\mathbf{T}_n(a_{|A|})$.

The respective equivalence classes in (5.1) and (5.2) are symmetric, so an analogous argument holds for the latter. The class that separates them, $\mathbf{T}_n(\varnothing)$, remains to be discussed.

If $a = a^{(1)}$, then $\mathbf{T}_n(a_{|A|})$ is undefined, but instead $\{a_{|A|}\} \in \mathbf{T}_n(\varnothing)$. Let $B = \{b_1, b_2, \ldots, b_{|B|}\}$ be the flippable diagonals for $\mathbf{T}_n(\varnothing)$, and set $b_{|B|} = a_{|A|}$. List $G_{|B|}$ in reverse order so that the last covering is $\{\}$. Now (5.2) can be listed, and all of the interfaces between equivalence classes satisfy the Gray code requirement that they differ by exactly one diagonal.

If n is odd, then the equivalence classes are ordered

$$\begin{aligned}
&\mathbf{T}_n(b_1^{\rightarrow}), \mathbf{T}_n(t_{n-3}^{\leftarrow}), \mathbf{T}_n(b_3^{\rightarrow}), \mathbf{T}_n(t_{n-5}^{\leftarrow}), \ldots, \mathbf{T}_n(a^{(1)}),\\
&\mathbf{T}_n(\varnothing),\\
&\mathbf{T}_n(b_{n-2}^{\leftarrow}), \mathbf{T}_n(t_2^{\rightarrow}), \mathbf{T}_n(b_{n-4}^{\leftarrow}), \mathbf{T}_n(t_4^{\rightarrow}), \ldots, \mathbf{T}_n(a^{(2)}),
\end{aligned}$$

with definitions for $a^{(1)}$ and $a^{(2)}$ similar to those above (see Figure 4.4). The argument for even n is easily adapted for odd n.

Theorem 5.2. *There exists a Gray code for listing the elements of $\mathbf{T}_n$ such that successive coverings differ by exactly one diagonal flip. The list can be created in constant amortised time per covering.*

Proof. The preceding description applies to the binary representation for coverings of $\mathbf{T}_n$, so that each class $\mathbf{T}_n(a)$ can be generated with Algorithm 1 of [3] in constant amortised time. There are $n-1$ or n calls to Algorithm 1, and for each one, the input is a set of $n-3$ or $n-2$ flippable diagonals. Each of these diagonals is determined in constant time by using Table 4.2. This requires $\mathcal{O}(n^2)$ preprocessing operations, but $|\mathbf{T}_n| = n2^{n-3}$, so our algorithm is CAT. □

Remark 5.3. *Successive classes $\mathbf{T}_n(a)$ and $\mathbf{T}_n(b)$ have a symmetric difference of exactly two flippable diagonals, provided that $a \neq \varnothing$*

Two problems remain open in this discussion. Is there a Gray code for $\mathbf{T}_n$ whose first and last elements also differ by 1 diagonal flip? A positive answer would provide a cyclic Gray code for all $n \times n$ tatami coverings with n monominoes.

5.2 Coverings in $\mathbf{T}_n$ with v vertical dominoes.

In this section, we present the CAT procedure `genVH`(n,k), which generates the coverings counted by $VH(n,k)$ (example output is shown in Figure 5.1, and also at `http://alejandroerickson.com/tatami/`). Let $\mathbf{S}(n,k)$ denote the set of subsets of $\{1,2,\ldots,n\}$ whose elements sum to k; thus $|\mathbf{S}(n,k)| = S(n,k)$. The procedure follows naturally from the sums in Equation (4.8a-4.8b), since each term $S(a,i)S(b,j)$ counts some set of '$\diagup$' and '$\diagdown$'-oriented diagonals. The sets $\mathbf{S}(a,i) \times \mathbf{S}(b,j)$ are generated in constant amortised time (CAT) by a modification of `C4` from [30] (see Listing 5.1). Our modified algorithm, `modC`, is invoked for each sum term of Equation (4.8a-4.8b). Procedure `modC` is CAT for the same reasons that `C4` is CAT.

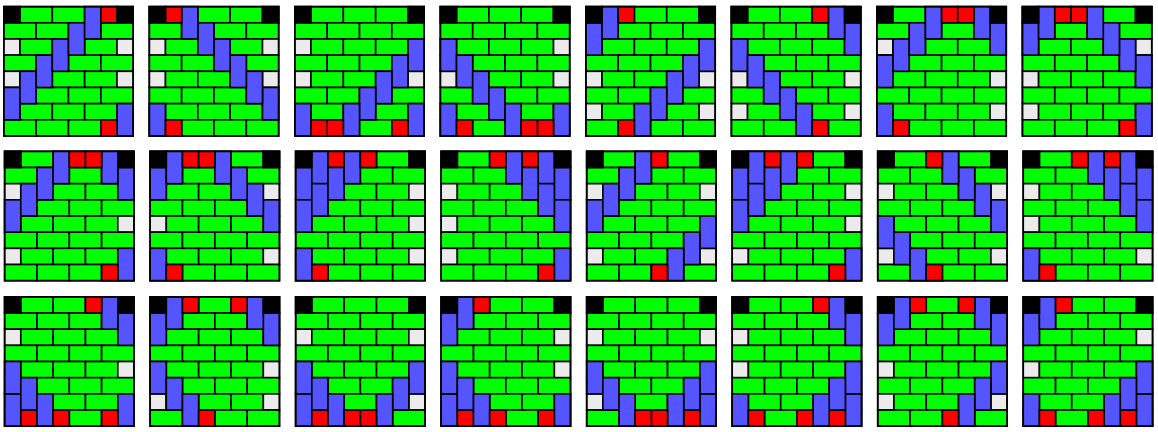


Figure 5.1: The coverings of $\mathbf{T}_8$ with exactly 7 vertical dominoes. This is the output of `genVH`$(8,7)$ printed in the order the coverings are generated (as one would naturally read text).

There is one subtlety involved in exploiting the CATness of `C4`. Invoking `C4`(a,i) requires $\Omega(a)$ preprocessing steps if its input list is recreated for each call, but `C4`(a,i) may not produce so many combinations for small a and large i. The result is that we may make many calls to `modC` that require too much preprocessing, but this is dealt with, as follows: a top level call to `C4`(i,j) in [30] takes the list $[i+1,2,3,\ldots,i+1]$, which requires $i+1$ steps to create, however, `C4`(i,j) also

Listing 5.1: Python code for a modified version of `C4` from [30] to compute $\mathbf{S}(a,i) \times \mathbf{S}(b,j)$. Global variables `aiSet` and `bjSet` are the lists representing $\mathbf{S}(a,i)$ and $\mathbf{S}(b,i)$, respectively

```
def modC(a,i,b,j,comp,isFirst):
    global aiSet,bjSet
    if( a == 0):
        if(isFirst):
            modC(b,j,0,0,False,False)
        else:
            Output(aiSet,bjSet)
    else:
        if(isFirst):
            L = aiSet
        else:
            L = bjSet
        if( i > a*(a+1)/2 ):
            i = a*(a+1)/2 - i; comp = not comp
        if( i<a ):
            if(comp):
                L[a] = L[0]; L[0] = i+1
                modC( i, i, b, j, comp, isFirst)
                L[0] = L[a]; L[a] = a+1
            else:
                modC( i, i, b, j, comp, isFirst)
        else:
            L[a] = L[0]; L[0] = a
            if(comp):
                modC( a-1, i, b, j, comp, isFirst)
                L[0] = L[a]; L[a] = a+1
                modC( a-1, i-a, b, j, comp, isFirst)
            else:
                modC( a-1, i-a, b, j, comp, isFirst)
                L[0] = L[a]; L[a] = a+1
                modC( a-1, i, b, j, comp, isFirst)
```

concludes with the same list (see Listing 5.1). Let A and B be the largest integers for which `modC` is called to compute $\mathbf{S}(a,i) \times \mathbf{S}(b,i)$. We set `aiSet`$= [1,2,3,\ldots,A]$ and `biSet`$= [1,2,3,\ldots,B]$, and by setting `aiSet`$[0] = a+1$ and `biSet`$[0] = b+1$, we initialise for each call to `modC` with exactly two operations.

Theorem 5.4. *The coverings in $\mathbf{T}_n$ with exactly k vertical dominoes can be exhaustively generated in constant amortised time.*

Proof. The outer procedure does a constant amount of preprocessing steps per call to `modC`. This subroutine is CAT, so the outer procedure is also CAT. □

5.3 Finite tatami coverings of the infinite strip

A *strip* of height r is a two-way infinitely wide integer grid of constant height r. The T-diagrams defined for rectangular grids also apply to the strip because the strip has no inside corners. The only difference is that there are no vertical boundaries. A *finite monomino-domino tatami strip covering* is a monomino-domino tatami covering of the strip with a finite number of T-diagram feature diagrams. We refer to these as strip coverings, as we do not consider any other type.

Two T-diagram feature diagrams are *isomorphic* if the respective sets of line segments they comprise are horizontal translations of each other. Two strip coverings are isomorphic if their respective feature diagrams, listed from left to right, are isomorphic.

Strip coverings, up to isomorphism, encapsulate some of the combinatorial properties of rectangle coverings without so many of the geometric details that arise when packing feature diagrams into a rectangle. In fact, we only consider Lemma 2.1(H1). On the other hand, the T-diagram of a strip covering can be bounded by two vertical lines, thereby converting it to a rectangular T-diagram. In this section we enumerate and generate coverings with k-features, of the height r strip.

Theorem 5.5. *If $R(r,n)$ is the number of non-isomorphic strip coverings with exactly n features, then it satisfies the system of homogeneous linear recurrence relations,*

$$V_r(n) = 4(r-1)V_r(n-1) + 2H_r(n-1), \text{ where } V_r(0) = 1, V_r(1) = 4r-2; \quad (5.3)$$

$$H_r(n) = 2V_r(n-1), \text{ where } H_r(0) = 1; \quad (5.4)$$

$$R(r,n) = V_r(n) + H_r(n). \quad (5.5)$$

Proof. Recall that a T-diagram partitions the strip into regions, covered by vertical or horizontal bond. Let $V_r(n)$ and $H_r(n)$ be the number of non-isomorphic strip coverings whose leftmost regions are vertical and horizontal bond, respectively. The number of non-isomorphic features on the height r strip are as follows:

Bidimers: There are $r-1$ vertical, and $r-1$ possible horizontal bidimers.

Vortices: There are $r-2$ clockwise, and $r-2$ possible counterclockwise vortices.

Vees: There is 1 vee on the top boundary and 1 vee on the bottom.

Loners: There are four loners, $\nearrow$, $\nwarrow$, $\searrow$, and $\swarrow$. The first two occur on the bottom boundary, and the latter on top boundary.

All of the bidimers, vortices and vees have vertical bond to their left and right. The $\searrow$ and $\nearrow$ loners have horizontal and vertical bond to their left and right, respectively, while the $\swarrow$ and $\nwarrow$ loners have vertical and horizontal bond to their left and right, respectively.

The bond coverings of the strip are either a horizontal or vertical bond. These are counted by the initial conditions $H_r(0) = V_r(0) = 1$. If the leftmost region of the covering is horizontal bond, then the leftmost feature must be a $\searrow$ or $\nearrow$ loner. The region to the left of the remaining features is a vertical bond, so $H_r(n) = 2V_r(n-1)$.

The total number of features with vertical bond on their left side is $(r-1) + (r-1) + (r-2) + (r-2) + 1 + 1 + 1 + 1$, so this gives $V_r(1) = 4r-2$. Exactly two of these features, namely $\nwarrow$ and $\swarrow$ loners, have horizontal bond on their right, so $V_r(n) = 4(r-1)V_r(n-1) + 2H_r(n-1)$.

Thus $R(r,n) = V_r(n) + H_r(n)$, as required. □

The following Maple commands give the generating function in Corollary 5.6:

```
eqn1 := V(n) = 4*(r-1)*V(n-1) + 2*H(n-1);
init1 := V(0) = 1, V(1) = 4*r-2;
eqn2 := H(n) = 2*V(n-1);
init2 := H(0) = 1, H(1)=2;
eqn3 := R(n) = V(n) + H(n);
init3 := R(0) = 2, R(1) = 4*r;
soln1 := rsolve({eqn1,init1,eqn2,init2,eqn3,init3},
                {R(n),V(n),H(n)},'genfunc'(z));
```

Corollary 5.6. *The generating function*

$$R_r(z) = \sum_{k \geqslant 0} R(r,k) z^k, \tag{5.6}$$

satisfies

$$R_r(z) = \frac{-2(2r-4)z + 2}{-4z^2 - (4r-4)z + 1}. \tag{5.7}$$

The first few values for $3 \leqslant r \leqslant 5$ are given below.

$r \backslash n$	0	1	2	3	4	5	6	7
3	2	12	104	880	7456	63168	535168	4534016
4	2	16	200	2464	30368	374272	4612736	56849920
5	2	20	328	5328	86560	1406272	22846592	371170560

An application of the Rational Expansion Theorem from [15] shows that the growth rate is approximately $(4r)^n$ (see page 340 of [15]). Let $R(z) = P(z)/Q(z)$, with $P(z) = -2(2r-4)z + 2$, and $Q(z) = -4z^2 - (4r-4)z + 1$. Applying the quadratic formula, we have $Q(z) = (1 - \rho_1 z)(1 - \rho_2 z)$, where $\psi = \sqrt{(r-1)^2 + 1}$,

$$\rho_1 = \frac{-2}{r - 1 + \psi} \text{ and } \rho_2 = \frac{-2}{r - 1 - \psi}.$$

The Rational Expansion Theorem says that $\langle z^n \rangle R(z) = a_1 \rho_1^n + a_2 \rho_2^n$, where the a_is are (known) constants. Our goal is to find the growth rate of $\langle z^n \rangle R(z)$, for large r, which is dominated by ρ_2, since $|\rho_1| < 1$ for $r \geqslant 2$. We show that $\rho_2 \approx 4r$; equivalently, we show that

$$\lim_{r \longrightarrow \infty} r(\psi - (r-1)) = 1/2. \tag{5.8}$$

Let $1/t = r$, so that

$$\begin{aligned} &\lim_{r \longrightarrow \infty} r(\psi - (r-1)) \\ &= \lim_{t \longrightarrow 0} \frac{\sqrt{(1/t - 1)^2 + 1} - 1/t + 1}{t} \\ &= \lim_{t \longrightarrow 0} \frac{\sqrt{(t-1)^2 + t^2} - 1 + t}{t^2}, \end{aligned}$$

and notice that this is the indeterminate form $0/0$. Two applications of L'Hopital's

rule yield Equation (5.8). The values of $r(\psi - (r-1))$ are

$$0.828427124, 0.708203931, 0.64911064, 0.61552813, 0.59411708, 0.57933771,$$

for $2 \leqslant r \leqslant 7$. Therefore for larger r, $\langle z^n \rangle R(z) = a_1\rho_1^n + a_2\rho_2^n \approx a_2(4r)^n$.

Corollary 5.7. *There exists a CAT algorithm for generating non-isomorphic, height r strip coverings.*

Proof. There are $4r$ possible non-isomorphic features in height r strip coverings, each of which can be expressed uniquely as an element of $\{0, 1, \ldots, 4r-1\}$. The recurrence relations in the proof of Theorem 5.5 describe a tree whose internal nodes are at least of degree 2, and whose leaves all represent output. The recursive algorithm which naturally arises from Theorem 5.5, iterates through the features that can be added, given the bond of the leftmost region. After adding each feature to the covering, using its unique symbol, the algorithm recurses. There is a constant number of operations per call and a constant number of calls per leaf. Therefore the algorithm is CAT, since there are more leaves than internal nodes. □

Comparing these results on non-isomorphic strip coverings to Theorem 3.17, we have a closed form formula for the generating function $R_r(z)$, while our methods for computing $T_r(z)$ become cost prohibitive at around $r = 15$. In addition, the methods of this section yield a straightforward CAT algorithm, whereas the same cannot be said of Theorem 3.17.

Monomino-domino fixed-height coverings are a natural extension of domino fixed-height coverings, proposed by Knuth in [19], and discussed in [26]. It is worth mentioning, therefore, that Theorem 5.5 might offer an improvement to Theorem 3.17, by considering things such as non-isomorphic strip coverings whose T-diagrams are bounded on the left and right, and strip coverings with minimal distance between features. The desired respective positions for adjacent pairs of features can be tabulated, for example, in a $4r \times 4r$ matrix.

Enumerating isomorphic strip coverings which fit in a bounded portion of the strip, perhaps is equivalent to counting a type of integer partition. That is, the total amount of space between features is a constant, while the placement of a feature can be shifted horizontally by an even number of grid squares, if it is unhindered by a neighbouring feature or a vertical boundary.

Finally, some work must be done to determine what can happen at the vertical boundaries.

Chapter 6

Domino Tatami Covering is NP-complete

In this chapter we consider domino-only tatami coverings of *rectilinear regions*, which we define as a finite subset of the integer grid. We say a rectilinear region, R, *is covered by dominoes* if it is covered exactly by non-overlapping dominoes. We describe a polynomial reduction from the NP-complete problem planar 3SAT to Domino Tatami Covering (DTC, see Definition 6.1). The gadgets used in the reduction were discovered with the help of a SAT-solver.

As a consequence it is therefore NP-complete to decide whether there is a perfect matching of a graph that meets every 4-cycle, even if the graph is restricted to be an induced subgraph, D, of the grid-graph. This abstraction provides a way of comparing DTC to its non-tatami counterpart. Since D is bipartite, deciding whether or not it admits a perfect matching is equivalent to a maximum flow problem, for which there are various polynomial algorithms.

Definition 6.1 (Domino Tatami Covering (DTC))**.**

INSTANCE: *A rectilinear region, R, represented as n grid squares.*

QUESTION: *Can R be covered exactly by non-overlapping dominoes such that no four of them meet at any one point?*

Theorem 6.2. *Domino tatami covering is NP-complete*

There are some previous complexity results about tilings and domino coverings. Historically, perhaps the first concerned colour-constrained coverings, such as those of Wang tiles. It is well known, for example, that covering the $k \times k$ grid

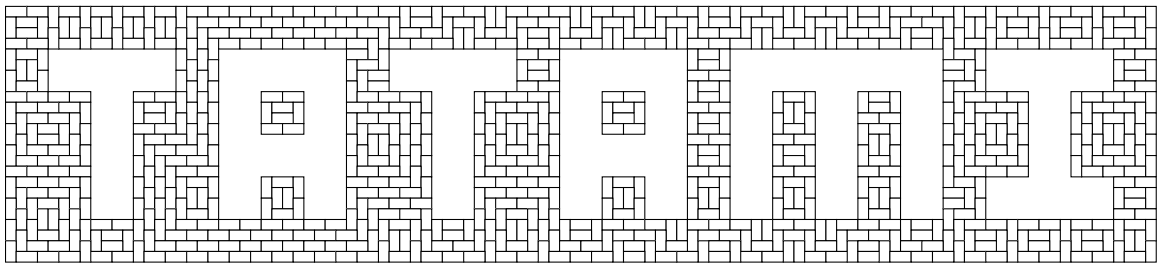


Figure 6.1: A domino tatami covering of a rectilinear region, produced by a SAT-solver.

with Wang tiles is NP-complete (see [21]). On the other hand tatami does not appear to be a special case of these, nor of similar colour restrictions on dominoes (e.g. [2,31]).

A more closely related mathematical context is found, instead, among the graph matching problems discussed by Churchley, Huang, and Zhu, in [4]. In their paper, an *H-transverse matching* of a graph G, is a matching M, such that $G - M$ has no subgraph H. In a tatami covering of the rectilinear grid, G is a finite induced subgraph of the infinite grid-graph, H is a 4-cycle, and we require a perfect matching of the edges. In fact, if the matching is not required to be perfect, the problem is polynomial.

There is no comprehensive structure theorem for tatami coverings of rectilinear grids, but evidently much of the structure is still there, as is illustrated in Figure 6.1. In contrast with other tatami-related results, however, we make no attempt to characterise this structure. Instead, our reduction relies on the interactions between coverings of a few specific regions that are discovered using a SAT-solver.

SAT-solvers have been applied to a broad range of industrial and mathematical problems in the last decade. Our reduction from planar 3SAT uses Minisat (see [7]) to help automate gadget generation, as was also done by Ruepp and Holzer (see [25]). It is easy to see that instances of other locally restricted covering problems can be expressed as satisfiability formulae, which suggests that SAT-solvers may provide a methodological applicability in hardness reductions involving those problems.

6.1 Preliminaries

Let ϕ be a CNF formula, with variables U, and clauses C. The formula is *planar* if there exists a planar graph $G(\phi)$ with vertex set $U \cup C$ and edges $\{u, c\} \in E$, where one of the literals u or $\bar{u}$ is in the clause c. When the clauses contain at most three literals, ϕ is an instance of planar 3SAT (P3SAT), which is NP-complete (see [22]).

We construct an instance of DTC which emulates a given instance, ϕ, of P3SAT, by replacing the vertices and edges of $G(\phi)$ with a rectilinear region, $R(\phi)$, that can be tatami-covered with dominoes if and only if ϕ is satisfiable. Let $n = |U \cup C|$. In Section 6.3 we show that $R(\phi)$ can be created in $\mathcal{O}(n)$ time, and that it fits in an $\mathcal{O}(n) \times \mathcal{O}(n)$ grid, by using Rosenstiehl and Tarjan's algorithm (see [24]).

6.2 Gadgets

In this section we describe wire, NOT gates, and AND gates, which form the required gadgets. The functionality of our gadgets depends on the coverings of a certain 8×8 sub-grid.

Lemma 6.3. *Let R be a rectilinear grid, with an 8×8 sub-grid, S. If a domino crosses the boundary of S in a domino tatami covering of R, then at least one corner of S is also covered by a domino that crosses its boundary.*

Proof. Suppose R is covered by dominoes, and consider those dominoes which cover S. Such a cover may not be exact, in the sense that a domino may cross the boundary of S. If we consider all such dominoes to be monominoes within S, we obtain a monomino-domino covering of S. This covering inherits the tatami restriction from the covering of R, so it is one of the 8×8 monomino-domino coverings enumerated in [14] (and/or [12]).

The proof of Lemma 3.10, paragraph 3, states that there is a monomino in at least one corner of S if $0 < m < n$; Corollary 3.5 states that there is a monomino in at least one corner of S if $m = n$. This monomino corresponds to a domino which crosses the boundary in a corner of S, as required. □

The rectilinear region $R(\phi)$ incorporates a network of 8×8 squares, whose centres reside on a $16\mathbb{Z} \times 16\mathbb{Z}$ grid, and whose corners form part of the boundary of $R(\phi)$. Lemma 6.3 implies that no domino may cross their boundaries, and thus

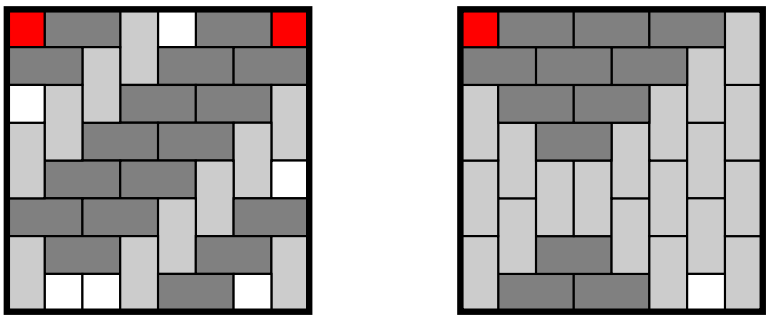


Figure 6.2: All monomino-domino tatami coverings of the square have at least one monomino in their corners (see [12,14]). The squares in $R(\phi)$ have isolate corners, so these must be covered in exactly one of the two ways given by Exercise 7.1.4.215 in [19], shown in the left and right-hand cross-hatched squares in Figure 6.3(a).

each one must be covered in one of the two ways shown in Figure 6.3(a). (For proofs see [26] and Exercise 215, Section 7.1.4 in [19]).

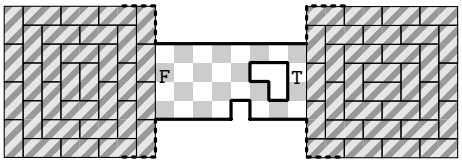
The coverings of these squares are related to each other by connecting regions. The part of an 8×8 square which borders on a connector may be covered either by two tiles, denoted by `F` to signify "false", or three tiles, denoted by `T` to signify "true" (see Figure 6.3(a)). Note that the covering of a square is not `T` or `F` by itself, because connectors below and beside it would meet the square at differing interfaces.

A connector, which imposes a relationship between the coverings of a set of 8×8 squares, is verified by showing that it can be covered if and only if the relationship is satisfied. The connectors we describe were generated with `SAT`-solvers, but they are simple enough that we can verify them by hand, as is done below.

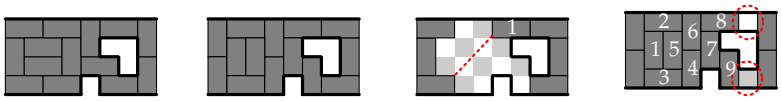
`NOT` gate. The `NOT` gate interfaces with two 8×8 squares (see Figure 6.3(a)), and can be covered if and only if these squares are covered with differing configurations.

Wire gadget. Wire transmits `T` or `F` through a sequence of squares (see Figure 6.4(a)). A turn may incorporate a `NOT` gate in order to maintain the same configuration (see Figure 6.4(b)).

AND gate. The `AND` gate interfaces with two 8×8 input squares, and one output square (see Figure 6.5). It can be covered with dominoes if and only if the output

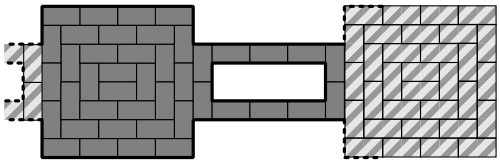


(a) NOT gate with F and T interfaces. This also shows the two possible domino coverings of the 8×8 square.

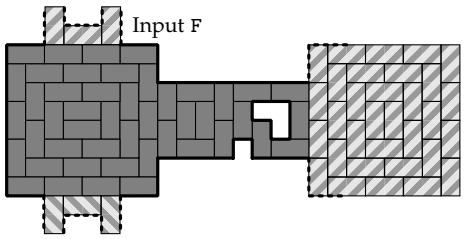


(b) F⟶T. (c) T⟶F. (d) F⟶F. (e) T⟶T.

Figure 6.3: NOT gate can be covered if and only if the input differs from the output. Numbered tiles indicate the (non-unique) ordering in which their placement is forced. Red dotted lines indicate how domino coverings are impeded in *(d)* and *(e)*.



(a) Unit of wire, carrying T.



(b) Wire branch and turn, carrying F.

Figure 6.4: Wire gadget.

value is the AND of the inputs (see Figs. 6.6 and 6.7).

Variable gadget. We use a vertical segment of wire. The variable gadget is set to T or F by choosing the appropriate covering of one of its 8×8 squares. Its value (or its negation) is propagated to clause gadgets via horizontal wire gadgets, representing edges.

Clause gadget. The clause gadget is a circuit for $\neg(\bar{a} \wedge (\bar{b} \wedge \bar{c}))$, or the equivalent with fewer inputs, ending in a configuration that can be covered if and only if the

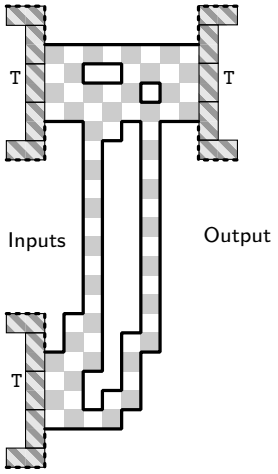


Figure 6.5: AND gate with input (T,T).

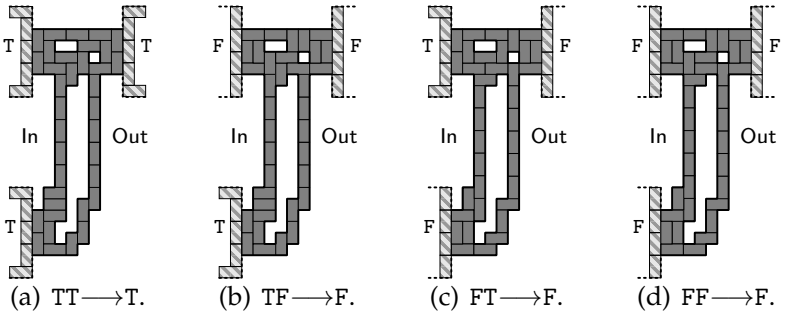


Figure 6.6: AND gate coverings.

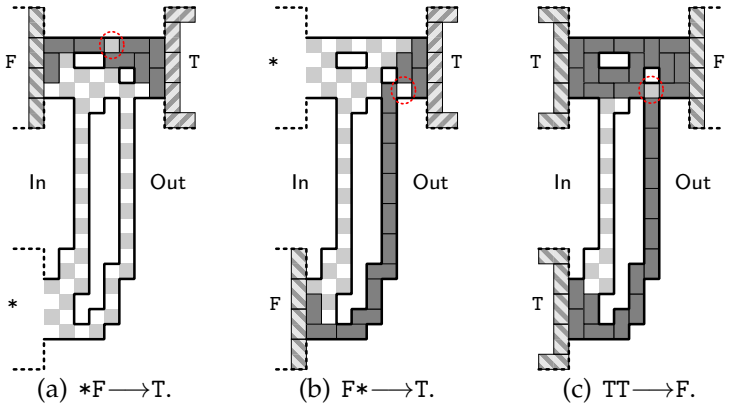
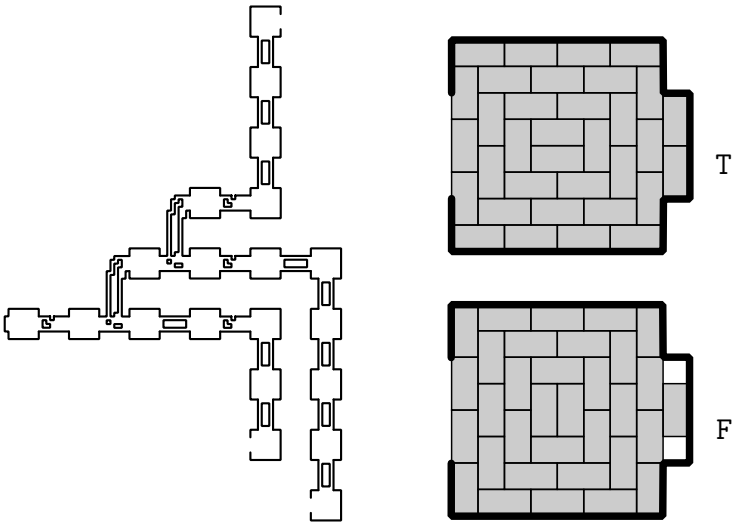


Figure 6.7: Impossible AND gate coverings, where * denotes F or T.

output signal of the circuit is T. To satisfy the layout requirements, the inputs to the clause are vertically translated by wire (see Figure 6.8).



(a) (b) End of clause.

Figure 6.8: A three input clause gadget from the circuit $\neg(\bar{a} \wedge (\bar{b} \wedge \bar{c}))$. Vertical wire translates horizontal inputs without changing the signal. The end of the clause is coverable if and only if its signal is T.

6.3 Layout

Let $G(\phi)$ be a planar embedding of the Boolean 3CNF formula ϕ, using Rosenstiehl and Tarjan's (see [24]) algorithm, so that each vertex is represented by a vertical line segment, and each edge is represented by a horizontal line segment. All parts lie on integer grid lines, inside of a $\mathcal{O}(n) \times \mathcal{O}(n)$ grid, where $n = |U \cup C|$, and the embedding is found in $\mathcal{O}(n)$ time.

There exists a constant K, which is the same for any planar 3CNF formula, such that $G(\phi)$ can be scaled to fit on the $nK \times nK$ grid, and its parts replaced by the gadgets described above. Each gadget has a constant number of grid squares, which ensures that $R(\phi)$ has $\mathcal{O}(n^2)$ grid squares altogether.

The variable gadget is connected to edges by branches. The layout of $G(\phi)$ prevents conflicts between edges meeting the variable gadget on the same side, while two edges can meet the left and right sides of the variable gadget without interfering with each other. The inputs of the clause gadget are symmetric, so there are no conflicts when connecting these to horizontal edges (see Figure 6.8(a)).

Example. The planar Boolean formula from Figure 1 in [22] gives the DTC instance in Figure 6.9.

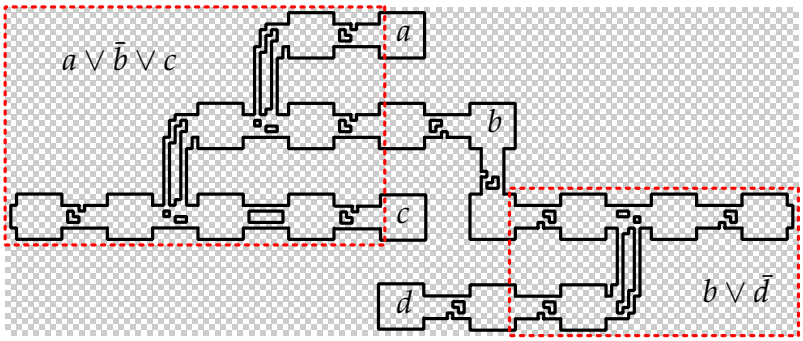


Figure 6.9: An instance of DTC for the formula $(a \vee \bar{b} \vee c) \wedge (b \vee \bar{d})$.

6.4 SAT-solver

The search for logical gates required fast testing of small DTC instances. We reduced DTC to SAT in order to use the SAT-solver, Minisat (see [7]), and efficiently test candidate regions connecting 8×8 squares while satisfying the conditions of the desired gate. The DTC solver was also allowed to make certain decisions about the region, rather than simply testing regions generated by another program. See Section A.2 for the computer script described below.

Our search algorithm requires the following inputs:

- an $r \times c$ rectangle of grid squares, partitioned into pairwise disjoint sets K, X, A, C; and,
- a set of partial (good) coverings, G, and partial (bad) coverings, B, of C.

The output, R, is the region $A' \cup K$, where $A' \subseteq A$, which satisfies the following constraints.

(g) There exist coverings of R which form partial tatami domino coverings with each element of G.

(b) There exists no covering of R which forms a partial tatami domino covering with an element of B.

The outer loop of the search algorithm calls the SAT-solver to find a region that satisfies all elements of G, and avoids a list of forbidden regions, which is initially empty. Upon finding such a region, the inner loop checks whether the region satisfies any element of B. The search succeeds when *(g)* and *(b)* are both satisfied, and fails if the outer loop's SAT instance has no satisfying assignment.

The search space grows very quickly for several reasons, not least of which is the fact that 2^{160} regions are possible within the 20×8 rectangle occupied by our AND gate (if corners are allowed to meet one another). In addition, the list of forbidden regions, L, becomes too large for the SAT-solver to handle efficiently.

We used two heuristics on the inputs to obtain a feasible search. The first was searching for a smaller AND gate, which we modified to fit the placement of the 8×8 squares. The second was choosing forbidden squares, X, and required squares, K, to reduce the number of trivially useless regions that are tested.

6.4.1 DTC as a Boolean formula

The SAT instances used above are modifications of a formula which is satisfiable if and only if a given region has a domino tatami covering.

Let R be the region we want to cover, and consider the graph whose vertices are the grid squares of R, and whose edges connect vertices of adjacent grid squares. Let H be the set of horizontal edges and let V be the set of vertical edges. The variables of the SAT instance are $H \cup V$, and those variables set to true in a satisfying assignment are the dominoes in the covering. The clauses are as follows, where $h, h' \in H$ and $v, v' \in V$.

1. Ensure a matching: For each pair of incident horizontal edges (h, h'), require the clause $\bar{h} \vee \bar{h}'$, and similarly for (v, v'), (h, v).

2. Ensure the matching is perfect: For each set of edges $\{h, h', v, v'\}$, which are incident to a vertex, require the clause $h \vee h' \vee v \vee v'$.

3. Enforce the tatami restriction: For each 4-cycle, $hvh'v'$, require the clause $h \vee h' \vee v \vee v'$.

6.5 Lozenge Tatami Covering

There are other locally constrained covering problems that are easily represented as Boolean formulae. Some of these are obviously polynomial, such as monomino-domino tatami covering, but others may be NP-complete. SAT-solvers can sometimes be used in such problems to create elaborate gadgets, which may help find a hardness reduction.

An example problem, whose computational complexity is open, is Lozenge-only Tatami Covering (see Definition 1.1). This problem is the decision about

whether or not a finite sub-grid of the triangular lattice can be covered with lozenges, such that no 5 lozenges meet at any point. A structure similar to that of tatami coverings occurs for this constraint (see Figure 6.10).

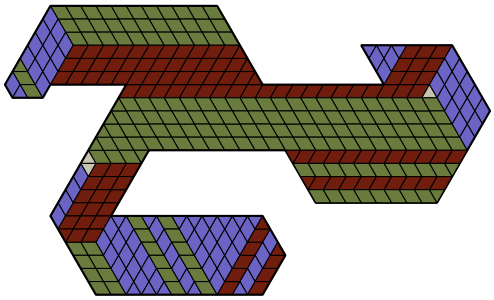


Figure 6.10: A triangle-lozenge tatami covering.

Our main question about DTC is the complexity of the case where the region is simply connected (no holes). It seems likely that the problem is still NP-complete, but a completely new approach will be required (see Section 7.1).

Secondarily, we are interested in H-transverse perfect matchings for H and G other than C_4 and grid-graphs. Are there other H-transverse perfect matchings of interest which induce a tatami-like global structure in the containing graph?

Another variant, mildly advocated by Don Knuth (personal communication), concerns inner corners of the coverings, such as occurs at the upper left in the letter T in Figure 6.1. If corners such as these, where a $+$ occurs, are forbidden but corners such as the upper right one in the I are allowed (a $\perp$ shape or one of its rotations), then the nature of tatami coverings changes. The complexity of such coverings is unknown.

Chapter 7

Open problems

7.1 Structure and complexity

The tatami structure opens up many questions, and we have answered a representative subset of these. Open problems related to the tatami restriction are discussed here. The first one concerns the global structure of general rectilinear regions.

Problem 7.1. *Given a rectilinear region, R, what are the minimal partial coverings of R which determine a unique monomino-domino tatami covering of R?*

Clearly, the four features in Figures 2.2(a)–2.4 force the propagation of rays, just as they do in rectangular coverings, but some care needs to be taken at inside corners. For example, a ray may begin at an inside corner. Given a characterisation of such minimal partial coverings, we would like to establish the analogue of Lemma 2.2; is every covering of the region, R, uniquely determined by the tiles on its boundary?.

Another consideration is the variant mentioned in Section 6.5. Rather than forbidding four tiles from meeting, we might forbid the edges of tiles to form a $+$-shape together; let this be called the *$+$-tatami* restriction. Tatami and $+$-tatami are the same for coverings of the rectangle, but the alternative definition forbids three tiles to meet at an inside corner. In this case a ray cannot meet an inside corner, which causes the tatami structure to break down somewhat. On the other hand, the results of Chapter 6 do not make explicit use of this. The `SAT`-solver searches could easily be modified to avoid $+$-shapes in the covering. Given any tatami problem on the rectilinear grid, we might ask the same problem with the $+$-tatami variant, so we choose `DTC` to represent this class of problems.

Problem 7.2. *Is the $+$-tatami variant of* DTC *NP-complete?*

Another variant of DTC is its restriction to simply connected regions. There appears to be no way to simulate planar 3SAT in a region with no holes. We might start by imagining an n-tentacled cephalopod-shaped region and having to find coverings of the tentacles which do not conflict in the mantle. Can the shape of the mantle be chosen to encode some NP-complete problem whose solution lies in the coverings of the tentacles? Whatever the case may be, this strengthening of DTC deserves a number.

Problem 7.3. *Is* DTC *NP-complete, even when the region R is simply connected?*

Constructive solutions to certain instances of DTC appear to be determined by the locations of bidimers in the region. These $\times$-shaped features motivate a loosely related continuous problem, proposed by Frank Ruskey. Consider the following configuration of a pair of orthogonal line segments, intersecting at a point, (x, y), strictly inside an arbitrary rectilinear region, R: a *water strider* at the point (x, y) is a pair of orthogonal line segments, which intersect at (x, y) (see Figure 7.1). Their lengths are maximal, without touching the boundary of R, and their slopes are ± 1.

The fact that water striders are open sets, and (x, y) cannot be on the boundary is only critical to the analogy with bidimers, but this may or may not be more interesting than the natural alternatives.

Definition 7.4 (Water Strider Problem, 2011, [12])**.**

INSTANCE: *A rectilinear region, R, with n segments, and vertices in $\mathbb{R}^2$.*

QUESTION: *Is there a configuration of at most k water striders, such that no two water striders intersect, and no more water striders can be added?*

A modification of the problem provides a stricter adherence to the analogue with DTC, and it is given below. A set of non-intersecting water striders in R partitions R into sub-regions. Each water strider bounds exactly four sub-regions, which are naturally above, below, and to the sides of its centre. In the modified Water Strider Problem, sub-regions to the left of each water strider may not have horizontal boundaries and the regions above and below it may not have vertical boundaries (of non-zero length).

Returning to the realm of the discrete, another natural problem on rectilinear grids, proposed by Matt DeVos (personal communication), is as follows.

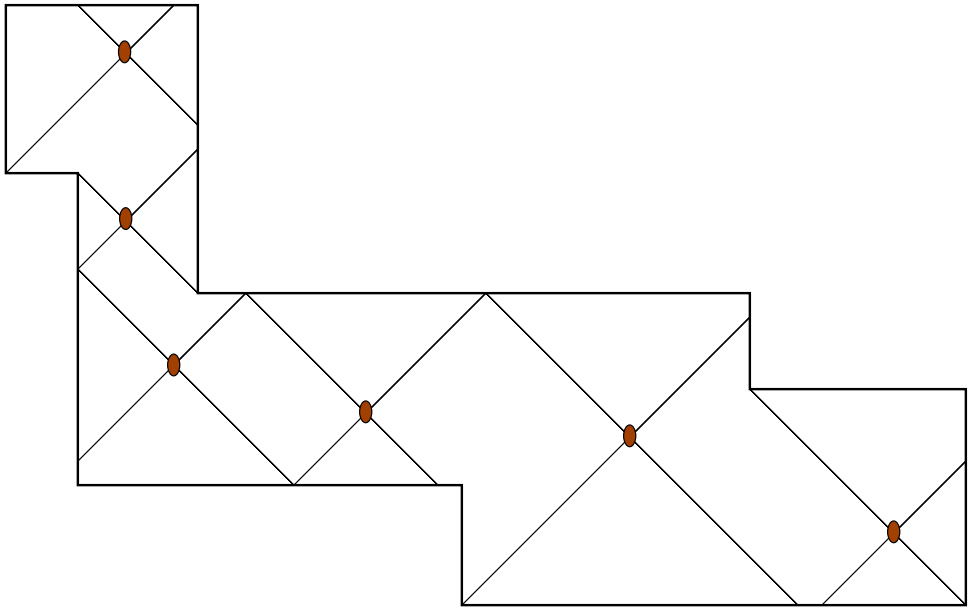


Figure 7.1: A maximal arrangement of water striders, derived from Figure 7.2

Definition 7.5 (Partial Tatami Covering, PTC)**.**

INSTANCE: *A rectilinear region R, made up of n segments, and a partial covering of R, with k tiles.*

QUESTION: *Is this partial covering part of a tatami covering of R?*

The point of interest, of course, is the computational complexity of this, not to mention its $+$-tatami variant.

Problem 7.6.

(a) Is PTC NP-complete?

(b) Is PTC polynomial when $n = 4$?

The following decision problem on rectangular grids, proposed by Martìn Matamala (private communication), is named *Tomoku*, after the puzzle game in [8] (see Figure 7.3). Without the tatami condition this decision problem is NP-complete (Theorem 4, [6]).

Definition 7.7 (Tomoku)**.**

INSTANCE: *An $r \times c$ rectangle, and for each row and each column, three integers indicating the number of grid squares covered by vertical dominoes, horizontal dominoes, and monominoes, respectively.*

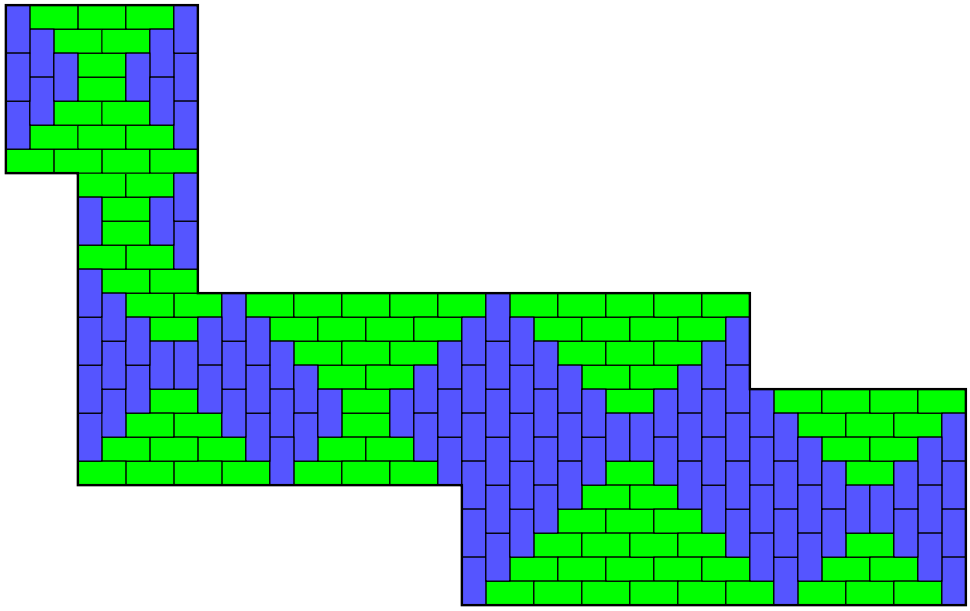


Figure 7.2: A solved instance of DTC.

QUESTION: *Does there exist a covering of the $r \times c$ rectangle with these row and column projections?*

Problem 7.8. *Is Tomoku NP-complete?*

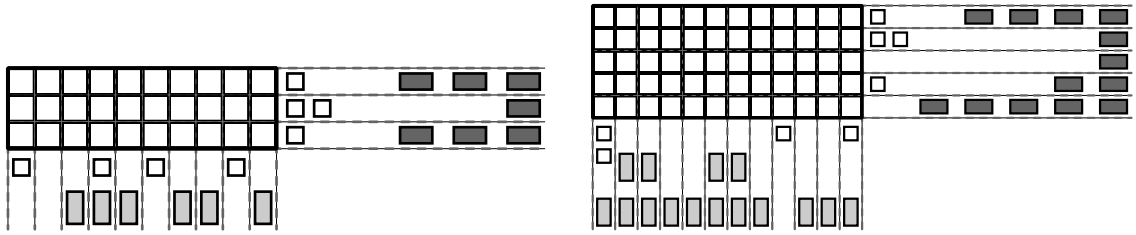
It is known that there are pairs of coverings with the same row and column projection. For example, an $n \times n$ covering with a central clockwise vortex gives the same row and column projections as a counterclockwise vortex.

Two instances are represented graphically in Figure 7.3, reprinted from [8]. Here we draw the tiles contained in each row and column, which is why monominoes appear twice. Tatami coverings do exist for these row and column projections, and finding them is an entertaining diversion.

The solution can be drawn with a pencil, but backtracking, and hence erasing is common. A mechanical realisation of the tatami restriction which improves game play is described in Section A.3.

7.2 Enumeration

The starting point for enumerating tatami coverings is characterising $T(r, c, m)$ for all parameters.



(a) (b)

Figure 7.3: Instances of Tomoku, reprinted from [8]. The 5×12 puzzle is quite challenging.

Problem 7.9. *Find $T(r, c, m)$ when $m > 1$, and m is not maximum, and m has the same parity as rc.*

Conjecture 3 from [12] seems to simplify Problem 7.9, so it is included below.

Conjecture 7.10 (Erickson, Ruskey, Schurch, Woodcock, 2011, [12])**.** *For all $d \geqslant 0$ and $m \geqslant 1$ there is an n_0 such that, for all $n \geqslant n_0$,*

$$T(n, n+d, m) = T(n_0, n_0+d, m),$$

whenever $n(n+d)$ has the same parity as m.

Experimentally, it appears that the smallest n_0 is $m + d + 4$, if $d \geqslant 1$.

Conjecture 3.14 should be mentioned here, though we have little doubt that it gives us $T(r, c, m)$ when m is maximum, and $r \leqslant c$.

Chapter 4 contains our most compelling enumeration problems, which concern the mysterious polynomial, $P(z)$. Of the ideas in Conjecture 4.9, we are most interested in proving the irreducibility of $P(z)$. Second to that is Conjecture 4.9(f); why does this pattern begin at $n \geqslant 20$? The broader question, however, is why does $\mathcal{VH}_n(z)$ factor this way at all?

Problem 7.11. *Is there a natural geometric interpretation for the factorisation of $\mathcal{VH}_n(z)$? Is there a closed form formula for $P(z)$?*

Chapter 4 touches briefly on balanced coverings of the $n \times n$ grid with n monominoes. Observe that a tatami covering of the $r \times c$ rectangle is a partition of the rectangle into $\mathcal{O}(\max(r, c))$ regions of horizontal and vertical bond (and isolated grid squares), by Lemma 2.2, and these regions are bounded either by ray diagrams, or the rectangle's boundary, which gives them a certain class of shape. Thus, a balanced tatami covering defines two collections of such shapes

whose total areas are the same (omitting monominoes). Balanced coverings were first investigated by Don Knuth (private communication), perhaps for this or other reasons, but virtually nothing has been published on them.

Problem 7.12. *How many balanced tatami coverings are there of the $r \times c$ rectangle, with m monominoes.*

7.3 Combinatorial Algorithms

Algorithms which use the diagonal flip are easily conceived. For example, Section 3.2.2 provides for making independent diagonal flips to produce the coverings of the $n \times n$ rectangle, with a given bidimer or vortex. Coverings of rectangles with maximum monominoes are also characterised by their diagonal flips, by Lemma 3.3, and algorithms for generating various cases are in the details of Conjecture 3.14.

Foremost, however, is to generate all coverings of a given rectangle, perhaps with an analogue to our algorithm for strip coverings, in Section 5.3.

Problem 7.13. *Generate all tatami coverings of the $r \times c$ grid in constant amortised time.*

Strip coverings ignore the precise horizontal location of a feature, and consider only the order in which the features appear, from left to right. In contrast, coverings of the rectangle are bounded, and their features have precise locations. Furthermore, the vertical boundaries introduce a minor complication, because (*a*) vees and loners may originate on vertical boundaries, and (*b*) a pair of rays which terminate at a vertical boundary may conflict if the covering is extended to a strip covering.

7.4 Triangular Tatami Coverings

Among the conceivable generalisations of tatami coverings, there is one which exhibits a global structure, akin to the one discussed in this dissertation (see Figure 7.8). A *triangle (tile)* in the triangular lattice is an equilateral triangle with unit side-length, and a *lozenge (tile)* is composed of two triangles, joined along one edge. A tile is *arranged on the lattice* if its vertices are at lattice points.

Definition 7.14 (Triangle-lozenge 5-tatami covering)**.** *A* triangle-lozenge 5-tatami covering *is an arrangement of triangles and lozenges, on the triangular lattice, in which no* 5 *tiles meet at any point. We shorten this to* 5*-tatami covering.*

Enumerating 5-tatami coverings of hexagonal regions extends research on boxed plane partitions (see [5,23]), just as monomino-domino tatami coverings of rectangular regions extend research on domino coverings. Our tatami problems on general rectilinear regions, on the other hand, can be asked for appropriately restricted finite regions of the triangular lattice; i.e. they may or may not be simply connected, or the outside corners of the region may or may not touch each other, etc. We describe the structure of triangle-lozenge 5-tatami coverings below.

We proceed as we did for monomino-domino tatami coverings, and reuse some of the vocabulary. Whenever two lozenges of distinct orientations share an edge, they propagate this same pattern, called a *ray* until it reaches the boundary (see Figure 7.4). A case analysis reveals that a ray may begin at exactly one of four types of features shown in Figures 7.4–7.7: a *tridimer*, *vortex*, *loner*, or *vee*.

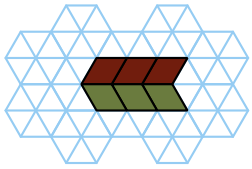


Figure 7.4: A ray propagates itself until it reaches a boundary. Six orientations are possible.

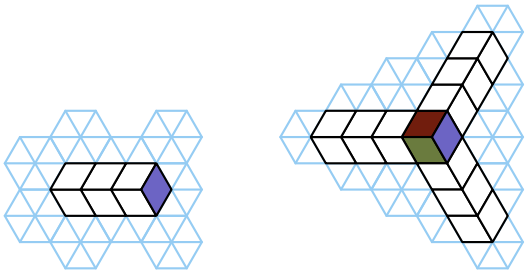


Figure 7.5: When a ray begins with a lozenge (of a different orientation), the resulting feature is a tridimer.

Grids with no inside corners appear to be readily enumerable, and otherwise, we have the analogue to rectilinear regions and all of the related questions. Some questions need to be modified. For example the question of balanced coverings takes on a three-way symmetry, since there are three orientations for lozenges.

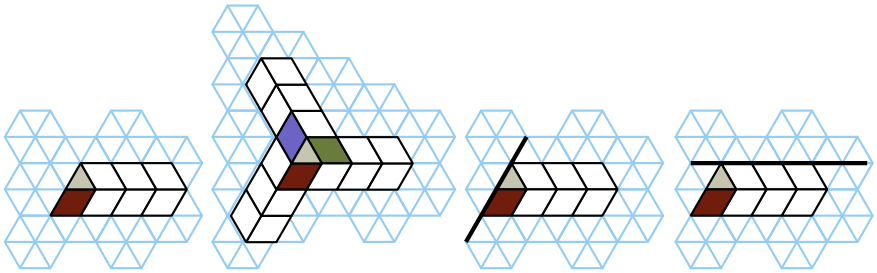


Figure 7.6: If the ray begins with a triangle and lozenge, then it is either a vortex, or one of two types of loner; a hiloner or a loloner.

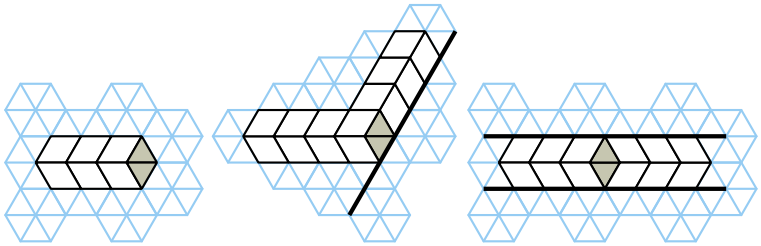


Figure 7.7: When there are two triangles at the beginning of the ray, we have a vee, or two loloners.

An $a \times b \times c$ *box* is an equi-angular hexagon whose vertices are lattice points, with side lengths a, b, c, a, b, c. Let $T(a, b, c, m)$ be the 5-tatami coverings of the $a \times b \times c$ box with exactly m triangles.

Problem 7.15. *What is* $T(a, b, c, m)$*?*

When $m = 0$, this counts the plane partitions of the $a \times b \times c$ box which satisfy the 5-tatami constraint. We are motivated to find some elegant "slice" of MacMahon's formula ([23, 29]),

$$\prod_{i=1}^{a}\prod_{j=1}^{b}\prod_{k=1}^{c}\frac{i+j+k-1}{i+j+k-2},$$

which counts box plane partitions in general.

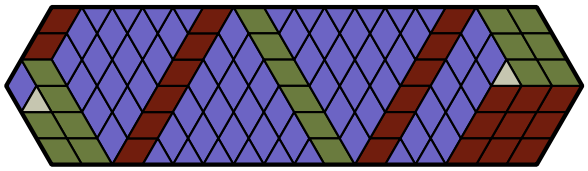


Figure 7.8: This $16 \times 3 \times 3$ tiling has every possible feature, up to rotation and reflection, except a vee. The tridimer can be replaced by a vee by cutting its central red lozenge into two triangles.

Chapter 8

Final Remarks

Curiosity has been our primary motivation to explore the tatami restriction so deeply, and we have good reason to be curious about it. Not only has the idea been around since at least the 17th century, it is a natural restriction of a well studied problem — monomino-domino coverings — with a starkly simple description: no four tiles meet. It is remarkable that the tatami structure was not studied earlier, from a computer science and mathematics point of view.

Our characterisation of the tatami restriction reveals that monomino-domino tatami coverings are a rich and mathematically harmonious topic. The structure is at first surprising, yet predictable, and visually appealing.

Summarising, tatami coverings of an $r \times c$ rectangle are determined by $\mathcal{O}(\max(r, c))$ sources. Of these, there are four types, up to reflection and rotation, and each consists of at most five tiles. Up to four rays emanate from each source, marking all of the boundaries between horizontal and vertical bond, forming a partition of the rectangle. Tatami coverings of rectilinear regions inherit much of the same structure, though some additional thought needs to be given to what can happen at inside corners. The most telling diagrams are Figures 2.1– 2.4, which show the features and coverings of rectangles, and Figure 1.1, which shows a covering of a non-rectangular region.

We use the structure for various enumeration results on coverings of rectangles, which fall into several categories. The first concerns coverings with a certain number of monominoes; we enumerate coverings with a maximum or minimum number of monominoes, with a precise number of monominoes, and with any number of monominoes. The second concerns coverings with a certain number of vertical dominoes.

The latter of these is perhaps the most interesting result after the structure

itself, and is restated below. Recall that $V(n,k)$ and $H(n,k)$ are the number of coverings in $\mathbf{T}_n$ with k vertical and horizontal dominoes, respectively, and that $S_n(z) = \prod_{i=1}^{n}(1+z^i)$. Theorem 4.2 states that $\mathcal{VH}_n(z) = \sum_{k\geqslant 0} V(n,k)z^k$ and $\mathcal{VH}_n(z) = \sum_{k\geqslant 0} H(n,k)z^k$ for even and odd n, respectively.

Theorem 4.6 (Erickson, Ruskey, 2013, [11])**.** *The generating polynomial $\mathcal{VH}_n(z)$ has the factorisation*

$$\mathcal{VH}_n(z) = P_n(z)D_n(z)$$

where $P_n(z)$ is a polynomial and

$$D_n(z) = \prod_{j\geqslant 1} S_{\left\lfloor \frac{n-2}{2^j} \right\rfloor}(z).$$

This result was first observed for small n by Don Knuth. Our general solution makes use of several results on tatami coverings, such as Theorem 3.2, on the maximum number of monominoes, and the diagonal flip characterisation. The recursive nature of the factorisation in Theorem 4.6 compels us to find a connection with the recursive nature of the coverings themselves, as well as with the family of polynomials, $P_n(z)$ (see Section 7.2).

In answering some of the most natural questions about tatami coverings, we showed that the topic has considerable mathematical richness. We developed a tool set, namely the structure, for solving problems in this area, and these techniques can be applied to several of the open problems mentioned in Chapter 7.

It is the present author's earnest hope that other computer scientists and mathematicians will become interested in tatami coverings, be it for the purest pursuit of knowledge, or some undiscovered application. They will not be disappointed.

Bibliography

[1] Artiom Alhazov, Kenichi Morita, and Chuzo Iwamoto. A note on tatami tilings. In *Proceedings of the 2009 LA Winter Symposium (Mathematical Foundation of Algorithms and Computer Science)*, volume 1691, pages 1 – 7, 2010.

[2] Therese Biedl. The complexity of domino tiling. In *Proceedings of the 17th Canadian Conference on Computational Geometry (CCCG)*, pages 187–190, 2005.

[3] James R. Bitner, Gideon Ehrlich, and Edward M. Reingold. Efficient generation of the binary reflected gray code and its applications. *Communications of the ACM*, 19:517–521, September 1976.

[4] Ross Churchley, Jing Huang, and Xuding Zhu. Complexity of cycle transverse matching problems. In *Combinatorial Algorithms (IWOCA)*, number 7056 in LNCS, pages 135–143. Springer Berlin / Heidelberg, January 2011.

[5] Henry Cohn, Michael Larsen, and James Propp. The shape of a typical boxed plane partition. *New York J. Math*, 4:137—166, 1998.

[6] Christoph Dürr, Flavio Guiñez, and Martín Matamala. Reconstructing 3-colored grids from horizontal and vertical projections is np-hard. In *Algorithms - ESA 2009*, volume 5757 of *LNCS*, pages 776–787. Springer Berlin / Heidelberg, 2009.

[7] Niklas Eén and Niklas Sörensson. An extensible sat-solver. In *Theory and Applications of Satisfiability Testing*, volume 2919 of *LNCS*, pages 333–336. Springer Berlin / Heidelberg, 2004.

[8] Alejandro Erickson. Tomoku! 80 challenging puzzles. Self Published, Victoria, Canada, Jan. 2012.

[9] Alejandro Erickson. TatamiMaker: A combinatorially rich mechanical game board. In *Bridges*, June 2013. 8 pages. `http://arxiv.org/abs/1301.5969`.

[10] Alejandro Erickson and Frank Ruskey. Domino Tatami Covering is NP-complete. In *The International Workshop on Combinatorial Algorithms (IWOCA)*, July 2013. 10 pages, to appear in Lecture Notes in Computer Science (LNCS). `http://arxiv.org/abs/1305.6669`.

[11] Alejandro Erickson and Frank Ruskey. Enumerating maximal tatami mat coverings of square grids with v vertical dominoes. Submitted to a journal. `http://arxiv.org/abs/1304.0070`, 2013.

[12] Alejandro Erickson, Frank Ruskey, Mark Schurch, and Jennifer Woodcock. Monomer-dimer tatami tilings of rectangular regions. *The Electronic Journal of Combinatorics*, 18(1):24, 2011.

[13] Alejandro Erickson and Mark Schurch. Enumerating tatami mat arrangements of square grids. In *International Workshop on Combinatorial Algorithms (IWOCA)*, volume 7056 of *LNCS*, pages 223–235. Springer Berlin / Heidelberg, January 2011.

[14] Alejandro Erickson and Mark Schurch. Monomer-dimer tatami tilings of square regions. *Journal of Discrete Algorithms*, 16(0):258–269, October 2012.

[15] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Professional, 2 edition, March 1994.

[16] Frank Gray. Pulse code communication, March 17 1953. U.S. Patent 2,632,058.

[17] Dean Hickerson. Filling rectangular rooms with tatami mats. `http://oeis.org/A068920/a068920.txt`, March 2002.

[18] Watanabe Hiroshi. *Entry for tatami, Kodansha encyclopedia of Japan*, volume 7. Kodansha International/USA Ltd., New York, NY, 1st edition, 1983.

[19] Donald E. Knuth. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*. Addison-Wesley Professional, 1st edition, January 2011.

[20] Serge Lang. *Algebraic number theory*. Addison-Wesley, 1970.

[21] Harry R. Lewis. Complexity of solvable cases of the decision problem for the predicate calculus. In *19th Annual Symposium on Foundations of Computer Science, 1978*, pages 35–47, October 1978.

[22] David Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329, 1982.

[23] Percy Alexander MacMahon. *Combinatory analysis, vols. 1 and 2*. Cambridge University Press, Cambridge, 1915, 1916.

[24] Pierre Rosenstiehl and Robert E. Tarjan. Rectilinear planar layouts and bipolar orientations of planar graphs. *Discrete & Computational Geometry*, 1(1):343–353, December 1986.

[25] Oliver Ruepp and Markus Holzer. The computational complexity of the Kakuro puzzle, revisited. In *Fun with Algorithms*, volume 6099 of *LNCS*, pages 319–330. Springer Berlin / Heidelberg, 2010.

[26] Frank Ruskey and Jennifer Woodcock. Counting fixed-height tatami tilings. *The Electronic Journal of Combinatorics*, 16:20, October 2009.

[27] Neil J. A. Sloane. Online encyclopedia of integer sequences. Published electronically at `http://oeis.org`, The On-Line Encyclopedia of Integer Sequences.

[28] Richard Stanley. *Enumerative combinatorics*, volume 1. Cambridge University Press, Cambridge, June 1997.

[29] Richard Stanley. *Enumerative Combinatorics*, volume 2. Cambridge University Press, Cambridge, February 2001.

[30] Dominique Roelants van Baronaigien and Frank Ruskey. Efficient generation of subsets with a given sum. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 14:87–96, 1993.

[31] Chris Worman and Mark D. Watson. Tiling layouts with dominoes. In *Proceedings of the 16th Canadian Conference on Computational Geometry (CCCG)*, pages 86–90, 2004.

Appendix A

Appendix

A.1 Tables

Tables A.1–A.19 were computed with a backtracking program that is ignorant of the tatami structure proven in Chapter 2. Periods (.) are used at parameters with no coverings due to Theorem 3.2.

$r\backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	1	2												
3	.	3	.											
4	1	4	4	2										
5	.	6	.	3	.									
6	1	9	6	3	2	2								
7	.	13	.	3	.	2	.							
8	1	19	10	5	2	1	2	2						
9	.	28	.	5	.	1	.	2	.					
10	1	41	16	6	4	2	0	1	2	2				
11	.	60	.	8	.	3	.	0	.	2	.			
12	1	88	26	8	4	4	2	0	0	1	2	2		
13	.	129	.	11	.	3	.	1	.	0	.	2	.	
14	1	189	42	13	6	3	4	2	0	0	0	1	2	2

Table A.1: Number of tatami coverings of the $r \times c$ grid with 0 monominoes, and $r \leqslant c$.

$r\backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1													
2	.	.												
3	2	.	10											
4	.	.	.	.										
5	3	.	18	.	10									
6	.	.	.	.	.	.								
7	4	.	38	.	8	.	10							
8	.	.	.	.	.	.	.	.						
9	5	.	72	.	18	.	4	.	10					
10	.	.	.	.	.	.	.	.	.	.				
11	6	.	136	.	24	.	4	.	4	.	10			
12	.	.	.	.	.	.	.	.	.	.	.	.		
13	7	.	250	.	32	.	18	.	0	.	4	.	10	
14	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Table A.2: Number of tatami coverings of the $r \times c$ grid with 1 monomino, and $r \leqslant c$.

$r\backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	1	4												
3	.	9	.											
4	3	18	27	32										
5	.	35	.	52	.									
6	6	64	75	62	60	32								
7	.	112	.	99	.	58	.							
8	10	192	177	152	102	46	50	32						
9	.	323	.	163	.	78	.	50	.					
10	15	534	393	258	184	100	36	36	50	32				
11	.	872	.	343	.	115	.	34	.	50	.			
12	21	1410	829	408	246	182	92	34	18	36	50	32		
13	.	2260	.	632	.	139	.	68	.	18	.	50	.	
14	28	3596	1691	746	414	212	174	92	18	16	18	36	50	32

Table A.3: Number of tatami coverings of the $r \times c$ grid with 2 monominoes, and $r \leqslant c$.

$r \backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	.	.												
3	1	.	12											
4	.	.	.	.										
5	4	.	56	.	88									
6	.	.	.	.	.	.								
7	10	.	198	.	160	.	88							
8	.	.	.	.	.	.	.	.						
9	20	.	570	.	340	.	122	.	88					
10	.	.	.	.	.	.	.	.	.	.				
11	35	.	1478	.	550	.	214	.	98	.	88			
12	.	.	.	.	.	.	.	.	.	.	.	.		
13	56	.	3554	.	936	.	372	.	80	.	98	.	88	
14	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Table A.4: Number of tatami coverings of the $r \times c$ grid with 3 monominoes, and $r \leqslant c$.

$r \backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	.	.												
3	.	1	.											
4	1	7	13	32										
5	.	26	.	64	.									
6	5	73	97	133	182	224								
7	.	179	.	269	.	359	.							
8	15	403	433	437	500	381	398	224						
9	.	850	.	730	.	766	.	398	.					
10	35	1707	1517	1243	1056	901	752	293	350	224				
11	.	3303	.	1823	.	1189	.	745	.	350	.			
12	70	6203	4571	2949	2050	1681	1420	729	440	237	350	224		
13	.	11366	.	4577	.	1574	.	1190	.	456	.	350	.	
14	126	20407	12479	6287	4054	2839	1960	1420	760	425	344	237	350	224

Table A.5: Number of tatami coverings of the $r \times c$ grid with 4 monominoes, and $r \leqslant c$.

$r\backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	.	.												
3	.	.	.											
4	.	.	.	.										
5	1	.	16	.	80									
6	.	.	.	.	.	.								
7	6	.	152	.	320	.	544							
8	.	.	.	.	.	.	.	.						
9	21	.	836	.	952	.	896	.	544					
10	.	.	.	.	.	.	.	.	.	.				
11	56	.	3472	.	2216	.	1992	.	696	.	544			
12	.	.	.	.	.	.	.	.	.	.	.	.		
13	126	.	12070	.	5044	.	3160	.	1632	.	568	.	544	
14	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Table A.6: Number of tatami coverings of the $r \times c$ grid with 5 monominoes, and $r \leqslant c$.

$r \backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	.	.												
3	.	.	.											
4	.	.	.	.										
5	.	1	.	7	.									
6	1	10	18	40	81	192								
7	.	52	.	110	.	368	.							
8	7	194	214	280	449	752	1024	1280						
9	.	597	.	669	.	1343	.	2040	.					
10	28	1624	1442	1318	1535	1944	2496	2064	2224	1280				
11	.	4046	.	2670	.	3238	.	4136	.	2256	.			
12	84	9428	7052	5240	4367	4288	5024	4280	3832	1616	2000	1280		
13	.	20847	.	9011	.	6421	.	6671	.	4136	.	2000	.	
14	210	44194	28158	16184	11539	9942	8808	7416	7832	3528	2360	1328	2000	1280

Table A.7: Number of tatami coverings of the $r \times c$ grid with 6 monominoes, and $r \leqslant c$.

$r\backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	.	.												
3	.	.	.											
4	.	.	.	.										
5	.	.	.	.	.									
6	.	.	.	.	.	.								
7	1	.	18	.	96	.	448							
8	.	.	.	.	.	.	.	.						
9	8	.	282	.	600	.	1728	.	2944					
10	.	.	.	.	.	.	.	.	.	.				
11	36	.	2292	.	2338	.	4272	.	4672	.	2944			
12	.	.	.	.	.	.	.	.	.	.	.	.		
13	120	.	13076	.	7728	.	8440	.	9440	.	3680	.	2944	
14	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Table A.8: Number of tatami coverings of the $r \times c$ grid with 7 monominoes, and $r \leqslant c$.

$r\backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	.	.												
3	.	.	.											
4	.	.	.	.										
5	.	.	.	.	.									
6	.	.	.	.	.	.								
7	.	1	.	9	.	40	.							
8	1	13	18	48	112	224	440	1024						
9	.	87	.	138	.	584	.	1920	.					
10	9	408	354	433	712	1368	2336	3904	5280	6656				
11	.	1532	.	1239	.	2497	.	6848	.	10576	.			
12	45	4951	3410	2849	3256	4072	6160	9408	12352	10432	11424	6656		
13	.	14361	.	6655	.	7602	.	14352	.	21216	.	11680	.	
14	165	38369	22480	14697	12360	11984	13696	16328	21952	20736	19008	8256	10400	6656

Table A.9: Number of tatami coverings of the $r \times c$ grid with 8 monominoes, and $r \leqslant c$.

$r\backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	.	.												
3	.	.	.											
4	.	.	.	.										
5	.	.	.	.	.									
6	.	.	.	.	.	.								
7	.	.	.	.	.	.	.							
8	.	.	.	.	.	.	.	.						
9	1	.	18	.	120	.	512	.	2304					
10	.	.	.	.	.	.	.	.	.	.				
11	10	.	426	.	824	.	3072	.	8704	.	14848			
12	.	.	.	.	.	.	.	.	.	.	.	.		
13	55	.	4808	.	4454	.	8424	.	20544	.	23040	.	14848	
14	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Table A.10: Number of tatami coverings of the $r \times c$ grid with 9 monominoes, and $r \leqslant c$.

$r \backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	.	.												
3	.	.	.											
4	.	.	.	.										
5	.	.	.	.	.									
6	.	.	.	.	.	.								
7	.	.	.	.	.	.	.							
8	.	.	.	.	.	.	.	.						
9	.	1	.	9	.	56	.	208	.					
10	1	16	18	48	128	288	608	1152	2224	5120				
11	.	131	.	154	.	784	.	2912	.	9472	.			
12	11	742	498	600	936	1928	3680	6816	11504	19200	25856	32768		
13	.	3308	.	1927	.	3665	.	12128	.	33296	.	51968	.	
14	66	12472	6494	4930	5816	6952	10632	18080	28848	44544	58880	50432	55808	32768

Table A.11: Number of tatami coverings of the $r \times c$ grid with 10 monominoes, and $r \leqslant c$.

$r\backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	.	.												
3	.	.	.											
4	.	.	.	.										
5	.	.	.	.	.									
6	.	.	.	.	.	.								
7	.	.	.	.	.	.	.							
8	.	.	.	.	.	.	.	.						
9	.	.	.	.	.	.	.	.	.					
10	.	.	.	.	.	.	.	.	.	.				
11	1	.	18	.	128	.	672	.	2560	.	11264			
12	.	.	.	.	.	.	.	.	.	.	.	.		
13	12	.	570	.	1040	.	4352	.	14976	.	41984	.	71680	
14	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Table A.12: Number of tatami coverings of the $r \times c$ grid with 11 monominoes, and $r \leqslant c$.

$r\backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	.	.												
3	.	.	.											
4	.	.	.	.										
5	.	.	.	.	.									
6	.	.	.	.	.	.								
7	.	.	.	.	.	.	.							
8	.	.	.	.	.	.	.	.						
9	.	.	.	.	.	.	.	.	.					
10	.	.	.	.	.	.	.	.	.	.				
11	.	1	.	9	.	64	.	304	.	1024	.			
12	1	19	18	48	128	320	736	1536	3072	5632	10752	24576		
13	.	184	.	172	.	928	.	4000	.	13952	.	45056	.	
14	13	1223	642	777	1160	2456	4960	9696	18048	32640	54656	91136	122368	155648

Table A.13: Number of tatami coverings of the $r \times c$ grid with 12 monominoes, and $r \leqslant c$.

$r\backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	.	.												
3	.	.	.											
4	.	.	.	.										
5	.	.	.	.	.									
6	.	.	.	.	.	.								
7	.	.	.	.	.	.	.							
8	.	.	.	.	.	.	.	.						
9	.	.	.	.	.	.	.	.	.					
10	.	.	.	.	.	.	.	.	.	.				
11	.	.	.	.	.	.	.	.	.	.	.			
12	.	.	.	.	.	.	.	.	.	.	.	.		
13	1	.	18	.	128	.	768	.	3456	.	12288	.	53248	
14	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Table A.14: Number of tatami coverings of the $r \times c$ grid with 13 monominoes, and $r \leqslant c$.

$r \backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	.	.												
3	.	.	.											
4	.	.	.	.										
5	.	.	.	.	.									
6	.	.	.	.	.	.								
7	.	.	.	.	.	.	.							
8	.	.	.	.	.	.	.	.						
9	.	.	.	.	.	.	.	.	.					
10	.	.	.	.	.	.	.	.	.	.				
11	.	.	.	.	.	.	.	.	.	.	.			
12	.	.	.	.	.	.	.	.	.	.	.	.		
13	.	1	.	9	.	64	.	368	.	1536	.	4864	.	
14	1	22	18	48	128	320	800	1792	3840	7680	14848	26624	50432	114688

Table A.15: Number of tatami coverings of the $r \times c$ grid with 14 monominoes, and $r \leqslant c$.

$r \backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.													
2	.	.												
3	.	.	.											
4	.	.	.	.										
5	.	.	.	.	.									
6	.	.	.	.	.	.								
7	.	.	.	.	.	.	.							
8	.	.	.	.	.	.	.	.						
9	.	.	.	.	.	.	.	.	.					
10	.	.	.	.	.	.	.	.	.	.				
11	.	.	.	.	.	.	.	.	.	.	.			
12	.	.	.	.	.	.	.	.	.	.	.	.		
13	.	.	.	.	.	.	.	.	.	.	.	.	.	
14	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Table A.16: Number of tatami coverings of the $r \times c$ grid with 15 monominoes (and greater), with $r \leqslant c$.

$r\backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1													
2	2	6												
3	3	13	22											
4	5	29	44	66										
5	8	68	90	126	178									
6	13	156	196	238	325	450								
7	21	357	406	490	584	827	1090							
8	34	821	852	922	1165	1404	1914	2562						
9	55	1886	1778	1714	2030	2828	3262	4618	5890					
10	89	4330	3740	3306	3619	4603	6228	7450	10130	13314				
11	144	9945	7822	6246	6080	7890	10226	14979	16734	23730	29698			
12	233	22841	16404	12102	10987	12475	17114	22803	31218	37154	50434	65538		
13	377	52456	34346	22994	19362	20396	25534	38778	50128	74610	81662	115970	143362	
14	610	120472	72004	43682	35477	34708	41034	54826	81298	109569	150114	178050	241410	311298

Table A.17: Number of tatami coverings of the $r \times c$ grid with any number of monominoes, and $r \leqslant c$; i.e., the sums of Tables A.1-A.16.

$m\backslash n$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	.	2	.	2	.	2	.	2	.	2	.	2	.	2
1	1	.	10	.	10	.	10	.	10	.	10	.	10	.
2	.	4	.	32	.	32	.	32	.	32	.	32	.	32
3	.	.	12	.	88	.	88	.	88	.	88	.	88	.
4	.	.	.	32	.	224	.	224	.	224	.	224	.	224
5	.	.	.	.	80	.	544	.	544	.	544	.	544	.
6	.	.	.	.	.	192	.	1280	.	1280	.	1280	.	1280
7	.	.	.	.	.	.	448	.	2944	.	2944	.	2944	.
8	.	.	.	.	.	.	.	1024	.	6656	.	6656	.	6656
9	.	.	.	.	.	.	.	.	2304	.	14848	.	14848	.
10	.	.	.	.	.	.	.	.	.	5120	.	32768	.	32768
11	.	.	.	.	.	.	.	.	.	.	11264	.	71680	.
12	.	.	.	.	.	.	.	.	.	.	.	24576	.	155648
13	.	.	.	.	.	.	.	.	.	.	.	.	53248	.
total	1	6	22	66	178	450	1090	2562	5890	13314	29698	65538	143362	311298

Table A.18: Number of tatami coverings of the $n \times n$ grid with m monominoes. The last row appears to be A027992 in [27].

$r\backslash c$	1	2	3	4	5	6	7	8	9	10	11	12	13
1		1	1	1	1	1	1	1	1	1	1	1	1
2			1	7	1	10	1	13	1	16	1	19	1
3				13	16	18	18	18	18	18	18	18	18
4					7	40	9	48	9	48	9	48	9
5						81	96	112	120	128	128	128	128
6							40	224	56	288	64	320	64
7								440	512	608	672	736	768
8									208	1152	304	1536	368
9										2224	2560	3072	3456
10											1024	5632	1536
11												10752	12288
12													4864

Table A.19: Number of tatami coverings of the $r \times c$ grid with the maximum number of monominoes and $r < c$ (see Conjecture 3.14).

$n\backslash z^k$	0	1	2	3	4	5	6	7	8	9	10
2	1										
3	1	2									
4	1	2	3	2							
5	1	2	3	6	4	2	2				
6	1	2	3	6	9	8	7	6	2	2	2
7	1	2	3	6	9	14	15	14	14	10	8
8	1	2	3	6	9	14	22	24	25	28	25
9	1	2	3	6	9	14	22	32	37	42	49
10	1	2	3	6	9	14	22	32	46	56	66
11	1	2	3	6	9	14	22	32	46	66	82
12	1	2	3	6	9	14	22	32	46	66	93
13	1	2	3	6	9	14	22	32	46	66	93
14	1	2	3	6	9	14	22	32	46	66	93
15	1	2	3	6	9	14	22	32	46	66	93
16	1	2	3	6	9	14	22	32	46	66	93
17	1	2	3	6	9	14	22	32	46	66	93
18	1	2	3	6	9	14	22	32	46	66	93
19	1	2	3	6	9	14	22	32	46	66	93
20	1	2	3	6	9	14	22	32	46	66	93
$n\backslash z^k$	20	21	22	23	24	25	26	27	28	29	30
8	2	2									
9	12	12	10	6	4	4	2	2	2		
10	51	44	38	34	28	22	20	16	14	12	8
11	145	136	122	108	100	86	74	68	58	52	48
12	313	310	302	286	272	254	229	210	193	170	157
13	533	562	576	588	588	580	567	540	511	482	449
14	761	848	921	980	1035	1074	1100	1118	1112	1092	1072
15	975	1122	1273	1418	1555	1686	1798	1904	1992	2048	2092
16	1161	1368	1593	1836	2086	2334	2586	2828	3060	3282	3474
17	1303	1576	1871	2202	2570	2958	3362	3792	4222	4656	5094
18	1410	1732	2101	2512	2982	3508	4079	4696	5362	6056	6787
19	1494	1848	2271	2764	3324	3966	4695	5506	6394	7372	8425
20	1541	1938	2396	2948	3598	4340	5199	6188	7297	8532	9917
$n\backslash z^k$	40	41	42	43	44	45	46	47	48	49	50
11	6	4	4	2	2	2					
12	48	40	34	30	24	22	18	12	10	8	6
13	190	174	156	138	126	114	102	90	78	66	58
14	585	540	505	466	426	400	368	340	316	286	260
15	1589	1496	1409	1328	1240	1164	1094	1020	962	902	840
16	3771	3658	3525	3378	3236	3082	2925	2780	2630	2490	2360
17	7675	7668	7627	7528	7384	7228	7023	6796	6568	6312	6052
18	13537	13926	14247	14488	14630	14702	14695	14596	14442	14226	13939
19	21175	22370	23479	24490	25406	26200	26861	27406	27804	28066	28211
20	30146	32528	34915	37242	39503	41692	43745	45658	47437	49012	50394
$n\backslash z^k$	60	61	62	63	64	65	66	67	68	69	70
13	8	6	4	4	2	2	2				
14	82	70	60	52	44	36	32	26	20	16	12
15	392	358	322	294	266	238	214	192	170	150	130
16	1322	1238	1162	1084	1012	944	878	812	754	698	640
17	3746	3558	3378	3218	3052	2894	2750	2602	2460	2332	2202
18	9792	9384	8969	8588	8220	7856	7522	7194	6873	6576	6293
19	23882	23118	22359	21574	20792	20036	19282	18544	17837	17142	16475
20	52767	51960	51021	50000	48845	47620	46364	45036	43684	42338	40967
$n\backslash z^k$	80	81	82	83	84	85	86	87	88	89	90
15	24	20	16	12	10	8	6	4	4	2	2
16	232	204	178	156	136	118	102	88	74	64	54
17	1112	1030	944	866	794	722	656	598	540	486	440
18	3844	3640	3440	3238	3052	2870	2686	2516	2352	2190	2042
19	11048	10596	10154	9732	9310	8902	8508	8118	7732	7362	7000
20	28837	27844	26864	25918	25022	24128	23267	22446	21629	20828	20061
$n\backslash z^k$	100	101	102	103	104	105	106	107	108	109	110
16	6	4	4	2	2	2					
17	122	104	90	76	64	56	46	36	30	24	20
18	884	800	724	654	586	528	474	420	372	328	288
19	3920	3670	3426	3192	2972	2762	2560	2372	2192	2014	1852
20	13174	12562	11974	11392	10822	10282	9746	9228	8738	8252	7778
$n\backslash z^k$	120	121	122	123	124	125	126	127	128	129	130
17	2										
18	66	56	44	36	30	24	20	16	12	10	8
19	702	628	562	498	438	388	342	300	262	228	198
20	3986	3692	3416	3160	2908	2672	2456	2246	2054	1876	1706
$n\backslash z^k$	140	141	142	143	144	145	146	147	148	149	150
19	36	30	24	20	16	12	10	8	6	4	4
20	584	518	456	402	354	308	268	234	202	174	150
$n\backslash z^k$	160	161	162	163	164	165	166	167	168	169	170
20	24	20	16	12	10	8	6	4	4	2	2

Table A.20: Table of coefficients of $\mathcal{H}_n(z)$ for $2 \leqslant n \leqslant 20$. The (n,k)th entry represents the number of coverings of $\mathbf{T}_n$ with k vertical dominoes when n is even, and k horizontal dominoes when n is odd (continued on next page).

11	12	13	14	15	16	17	18	19
6	4	2	2	2				
22	19	14	10	10	8	4	4	2
48	49	46	38	34	30	24	20	16
78	84	90	92	88	81	76	69	58
98	120	134	150	164	167	166	165	158
116	143	176	203	236	265	284	301	312
128	163	202	251	298	351	404	451	494
128	176	224	280	352	422	504	593	676
128	176	238	304	384	482	584	707	840
128	176	238	319	410	517	650	796	968
128	176	238	319	426	545	688	868	1066
128	176	238	319	426	562	718	909	1144
128	176	238	319	426	562	736	941	1188
128	176	238	319	426	562	736	960	1222
31	32	33	34	35	36	37	38	39
6	4	4	2	2	2			
40	32	28	24	20	18	14	10	8
144	126	114	102	90	82	74	64	56
416	386	354	322	298	272	248	230	210
1032	987	944	886	830	782	726	673	630
2114	2107	2090	2054	1990	1926	1852	1763	1678
3642	3788	3894	3971	4018	4023	3996	3948	3872
5508	5904	6286	6621	6918	7183	7378	7526	7632
7540	8289	9054	9813	10540	11250	11914	12511	13058
9552	10747	11986	13271	14602	15942	17288	18629	19922
11428	13078	14870	16765	18780	20915	23126	25417	27776
51	52	53	54	55	56	57	58	59
4	4	2	2	2				
50	42	36	30	26	22	16	12	10
238	214	192	174	156	140	126	108	92
788	736	682	636	592	546	508	472	428
2228	2103	1990	1879	1772	1675	1580	1488	1408
5802	5537	5278	5039	4794	4559	4344	4130	3926
13620	13258	12848	12433	12004	11550	11106	10663	10214
28214	28092	27866	27513	27074	26562	25966	25311	24618
51604	52573	53328	53883	54186	54281	54194	53888	53398
71	72	73	74	75	76	77	78	79
10	8	6	4	4	2	2	2	
112	98	86	72	62	54	44	38	32
586	534	484	442	400	360	326	292	260
2072	1952	1830	1710	1604	1494	1390	1298	1202
6010	5743	5484	5222	4980	4742	4500	4278	4060
15846	15231	14636	14070	13522	12990	12486	11988	11502
39606	38292	36984	35698	34472	33263	32092	30986	29892
91	92	93	94	95	96	97	98	99
2								
46	38	30	24	20	16	12	10	8
394	352	314	276	242	214	186	162	142
1902	1760	1632	1508	1384	1272	1166	1062	970
6646	6310	5978	5654	5346	5040	4744	4462	4184
19298	18550	17840	17126	16428	15760	15088	14436	13804
111	112	113	114	115	116	117	118	119
16	12	10	8	6	4	4	2	2
254	222	192	168	146	124	106	92	76
1698	1552	1418	1292	1174	1066	964	868	782
7330	6888	6464	6064	5674	5302	4952	4612	4288
131	132	133	134	135	136	137	138	139
6	4	4	2	2	2			
172	148	126	108	92	78	66	54	44
1550	1408	1272	1146	1034	928	830	744	660
151	152	153	154	155	156	157	158	159
2	2	2						
128	108	94	78	64	54	44	36	30
171	172	173	174	175	176	177	178	179
2								

Table A.20: Table of coefficients of $\mathcal{W}H_n(z)$ (continued from previous page).

A.2 SAT-solver gadget search

We include the python script which calls MiniSat (see [7]), in Listing A.1.

Listing A.1: Python script which calls MiniSat to find gadgets for reduction in Chapter 6. **Disclaimer:** This script serves its intended purpose, however, it has not been optimized in general, and may contains errors and bugs. Note also, that references to tilings are considered to be coverings, within this dissertation.

```
############################
# File: findgadget.py
# Author: Alejandro Erickson
############################

import subprocess
import sys
import csv

# Input two file names (The second will be overwritten with output
# from minisat).  minisat must be in the same directory as findgadget.py

#The first is the input and has the format:

# r
# c
# <r rows and c columns of ., X, #, C.
# . delineates the r x c grid
# X denotes forbidden squares
# # denotes required squares
# C denotes squares that are part of the forced configurations. >
# C (number of desirable (good) configurations)
# <r rows and c columns of ., A,V, <,>.
# A
# V denotes a vertical dimer
# <> denotes a horizontal dimer
# These special symbols can only appear on C squares above.  Repeat
# this for each desirable configuration>
# badC (number of undesirable (bad) configurations)
# <r rows and c columns of ., A,V, <,>. same as above>

# Example input
# 4
# 12
# CC#......#CC
# CC#......#CC
# CC#......#CC
# CC#......#CC
# 2
# .A........<>
# .V........A.
# .A........V.
# .V........<>
#
# <>........A.
# .A........V.
# .V........A.
# <>........V.
# 2
# <>........<>
# .A........A.
# .V........V.
# <>........<>
#
# .A........A.
# .V........V.
# .A........A.
# .V........V.

f = open(sys.argv[1],'r')
```

$n\backslash z^k$	0	1	2	3	4	5	6	7	8	9	10
2	2	2									
3	2	4	4	2							
4	2	4	6	8	6	4	2				
5	2	4	6	12	12	8	12	12	6	4	2
6	2	4	6	12	18	20	18	16	16	18	20
7	2	4	6	12	18	28	34	32	32	28	28
8	2	4	6	12	18	28	44	52	54	60	58
9	2	4	6	12	18	28	44	64	78	88	102
10	2	4	6	12	18	28	44	64	92	116	136
11	2	4	6	12	18	28	44	64	92	132	168
12	2	4	6	12	18	28	44	64	92	132	186
13	2	4	6	12	18	28	44	64	92	132	186
14	2	4	6	12	18	28	44	64	92	132	186
15	2	4	6	12	18	28	44	64	92	132	186
16	2	4	6	12	18	28	44	64	92	132	186
17	2	4	6	12	18	28	44	64	92	132	186
18	2	4	6	12	18	28	44	64	92	132	186
19	2	4	6	12	18	28	44	64	92	132	186
$n\backslash z^k$	20	21	22	23	24	25	26	27	28	29	30
7	4	2									
8	54	52	44	28	18	12	6	4	2		
9	84	92	96	104	106	104	102	88	78	64	44
10	146	144	144	144	144	146	156	170	180	186	192
11	330	320	300	280	280	268	252	252	252	252	268
12	662	664	652	632	612	588	554	532	514	488	478
13	1090	1156	1196	1228	1236	1232	1218	1180	1138	1096	1054
14	1542	1720	1874	2000	2122	2212	2272	2324	2328	2304	2284
15	1966	2264	2570	2868	3150	3420	3660	3884	4072	4204	4308
16	2334	2752	3206	3696	4204	4708	5220	5716	6196	6656	7056
17	2614	3164	3758	4424	5164	5948	6764	7632	8504	9384	10280
18	2828	3472	4214	5040	5984	7040	8190	9432	10772	12172	13646
19	2992	3704	4550	5540	6664	7952	9414	11044	12828	14792	16910
$n\backslash z^k$	40	41	42	43	44	45	46	47	48	49	50
10	28	18	12	6	4	2					
11	340	308	276	244	200	168	132	92	64	44	28
12	554	588	612	632	652	664	662	648	622	584	542
13	840	844	856	872	896	936	976	1012	1054	1096	1138
14	1646	1600	1582	1564	1532	1536	1536	1532	1564	1582	1600
15	3654	3524	3406	3300	3196	3112	3044	2984	2940	2896	2864
16	8006	7836	7634	7408	7192	6964	6734	6528	6328	6152	6000
17	15778	15820	15806	15684	15472	15244	14926	14564	14216	13820	13416
18	27458	28296	29002	29552	29916	30148	30230	30140	29940	29624	29186
19	42694	45136	47414	49504	51412	53084	54498	55688	56604	57256	57678
$n\backslash z^k$	60	61	62	63	64	65	66	67	68	69	70
12	44	28	18	12	6	4	2				
13	918	820	710	604	506	408	330	256	186	132	92
14	2228	2284	2304	2328	2324	2272	2212	2122	2000	1874	1720
15	3112	3196	3300	3406	3524	3654	3784	3910	4044	4152	4240
16	5288	5292	5300	5328	5374	5432	5514	5604	5714	5852	6000
17	10480	10324	10176	10096	10008	9932	9904	9868	9840	9868	9904
18	22848	22288	21742	21260	20820	20416	20076	19760	19486	19256	19062
19	51160	49940	48746	47532	46328	45192	44088	43032	42058	41136	40290
$n\backslash z^k$	80	81	82	83	84	85	86	87	88	89	90
14	256	186	132	92	64	44	28	18	12	6	4
15	3420	3150	2868	2570	2264	1966	1692	1422	1176	968	772
16	8006	8152	8252	8304	8318	8272	8146	7964	7724	7412	7056
17	11342	11648	11966	12288	12662	13048	13416	13820	14216	14564	14926
18	18656	18766	18900	19062	19256	19486	19760	20076	20416	20820	21260
19	35388	35192	35032	34928	34856	34820	34820	34856	34928	35032	35192
$n\backslash z^k$	100	101	102	103	104	105	106	107	108	109	110
15	28	18	12	6	4	2					
16	2334	1944	1600	1304	1038	824	638	476	352	256	186
17	14610	14044	13422	12724	11936	11128	10280	9384	8504	7632	6764
18	27464	28116	28688	29186	29624	29940	30140	30230	30148	29916	29552
19	39532	40290	41136	42058	43032	44088	45192	46328	47532	48746	49940
$n\backslash z^k$	120	121	122	123	124	125	126	127	128	129	130
16	2										
17	1094	852	638	476	352	256	186	132	92	64	44
18	18240	16690	15168	13646	12172	10772	9432	8190	7040	5984	5040
19	57832	57678	57256	56604	55688	54498	53084	51412	49504	47414	45136
$n\backslash z^k$	140	141	142	143	144	145	146	147	148	149	150
18	476	352	256	186	132	92	64	44	28	18	12
19	19176	16910	14792	12828	11044	9414	7952	6664	5540	4550	3704
$n\backslash z^k$	160	161	162	163	164	165	166	167	168	169	170
19	256	186	132	92	64	44	28	18	12	6	4

Table A.21: Table of coefficients of $R_n(z,1)$ for $2 \leqslant n \leqslant 19$. The (n,k)th entry represents the number of coverings in all four rotations of $\mathbf{T}_n$ with k vertical (or horizontal, by rotational symmetry), dominoes (continued on next page).

11	12	13	14	15	16	17	18	19
18	12	6	4	2				
28	28	32	32	34	28	18	12	6
52	54	48	40	48	54	52	58	60
104	106	104	96	92	84	80	80	80
160	176	188	196	192	186	180	170	156
200	244	276	308	340	350	352	358	352
236	290	356	414	480	542	584	622	648
256	330	408	506	604	710	820	918	1008
256	352	452	564	708	852	1016	1198	1368
256	352	476	612	772	968	1176	1422	1692
256	352	476	638	824	1038	1304	1600	1944
256	352	476	638	852	1094	1380	1740	2140
256	352	476	638	852	1124	1440	1822	2292
256	352	476	638	852	1124	1472	1886	2380
31	32	33	34	35	36	37	38	39
28	18	12	6	4	2			
196	188	176	160	136	116	92	64	44
280	280	300	320	330	352	358	352	350
468	456	456	456	468	478	488	514	532
1012	976	936	896	872	856	844	840	840
2228	2158	2104	2024	1940	1876	1800	1730	1688
4372	4386	4376	4332	4240	4152	4044	3910	3784
7412	7724	7964	8146	8272	8318	8304	8252	8152
11128	11936	12724	13422	14044	14610	15040	15376	15636
15168	16690	18240	19778	21264	22712	24076	25314	26452
19176	21582	24080	26674	29360	32068	34792	37510	40140
51	52	53	54	55	56	57	58	59
18	12	6	4	2				
480	414	356	290	236	186	132	92	64
1180	1218	1232	1236	1228	1196	1156	1090	1008
1646	1688	1730	1800	1876	1940	2024	2104	2158
2848	2836	2836	2848	2864	2896	2940	2984	3044
5852	5714	5604	5514	5432	5374	5328	5300	5292
13048	12662	12288	11966	11648	11342	11092	10856	10632
28688	28116	27464	26806	26132	25432	24756	24094	23444
57832	57748	57468	56954	56280	55472	54516	53458	52340
71	72	73	74	75	76	77	78	79
64	44	28	18	12	6	4	2	
1542	1368	1198	1016	852	708	564	452	352
4332	4376	4386	4372	4308	4204	4072	3884	3660
6152	6328	6528	6734	6964	7192	7408	7634	7836
9932	10008	10096	10176	10324	10480	10632	10856	11092
18900	18766	18656	18564	18516	18484	18484	18516	18564
39532	38830	38196	37628	37124	36672	36280	35932	35624
91	92	93	94	95	96	97	98	99
2								
612	476	352	256	186	132	92	64	44
6656	6196	5716	5220	4708	4204	3696	3206	2752
15244	15472	15684	15806	15820	15778	15636	15376	15040
21742	22288	22848	23444	24094	24756	25432	26132	26806
35388	35624	35932	36280	36672	37124	37628	38196	38830
111	112	113	114	115	116	117	118	119
132	92	64	44	28	18	12	6	4
5948	5164	4424	3758	3164	2614	2140	1740	1380
29002	28296	27458	26452	25314	24076	22712	21264	19778
51160	52340	53458	54516	55472	56280	56954	57468	57748
131	132	133	134	135	136	137	138	139
28	18	12	6	4	2			
4214	3472	2828	2292	1822	1440	1124	852	638
42694	40140	37510	34792	32068	29360	26674	24080	21582
151	152	153	154	155	156	157	158	159
6	4	2						
2992	2380	1886	1472	1124	852	638	476	352
171	172	173	174	175	176	177	178	179
2								

Table A.21: Table of coefficients of $R_n(z,1)$ (continued from previous page).

```
g = f.readlines()
f.close()

try:
    r = int(g[0])
    c = int(g[1])
except ValueError:
    quitError('Invalid input file')

goodClauses = [] #use this to store all the good clauses and do a
                 #single write.

s = 0
#These record which squares are part of the
#configurations.  They have less restrictions on them to allow for
#dominoes which are not contained in the r x c grid.
Cregion = [[0]*c for i in range(r)]
#These are the X_i variables for grid squares.  There are r*c of them
X = [[0]*c for i in range(r)]
i = 0 #i, j indices for X
j = 0

C = int(g[r+2]) #Number of good configurations
badC = int(g[r+2+C*(r+1)]) # Number of bad configurations
nGoodVars = r*c + C*((r-1)*c + r*(c-1))
nBadVars =  r*c + (r-1)*c + r*(c-1)
HH = []
VV = []
# store list of clauses describing bad configs for each k in badC
badClauses=[]
for i in range(badC):
#one list of clauses for each bad configuration
    badClauses.append([])

#For convenience, we use r x c arrays to represent each of these.
for i in range(C):
    HH.append([[0]*c for i in range(r)])
    VV.append([[0]*c for i in range(r)])

#if a literal is 0, we do not include it
def clause(L):
    global goodClauses
    goodClauses.append([])
    for l in L:
        if(not l == 0):
            goodClauses[-1].append(l)

def badclause(k,L):
    global badClauses
    #each clauses is a list of literals
    i = 0
    badClauses[k].append([])
    for l in L:
        if(not l==0):
            i = i + 1
            badClauses[k][-1].append(l)
    if(i==0):
        quitError("adding empty clause to badClauses["+str(k)+"] " +  str(badClauses[k]))

#R is a ordered list of integers from +/-1 to +/-r*c
def displayRegion(R):
    global r,c
    s = ''
    print R
    for i in range(len(R)):
        if(R[i]>0):
            s = s + '#'
        else:
            s = s + '.'
        if(R[i]%c == 0):
            s = s + '\n'
    s = s + '\n'
    print s

```

```
#displays the kth tiling in the list of good configurations.  Note that the bad configurations take k=0
def displayTiling(T,k):
    global r,c
    print T
    C = [[0]*(2*c+1) for i in range(r+1)]

    for i in range(r):
        for j in range(c-1):
            if(HH[k][i][j] in T):
                C[i][2*j+1] = C[i][2*j+3] = C[i+1][2*j] = C[i+1][2*j+4] = C[i+1][2*j+1] = C[i+1][2*j+3] = 1

    for i in range(r-1):
        for j in range(c):
            if(VV[k][i][j] in T):
                C[i][2*j+1] = C[i+1][2*j] = C[i+1][2*j+2] = C[i+2][2*j] = C[i+2][2*j+1] = C[i+2][2*j+2] = 1

    s = ''
    for i in range(r+1):
        for j in range(2*c+1):
            if(C[i][j] == 1):
                if(j%2 == 0):
                    s = s + '|'
                else:
                    s = s + '_'
            else:
                s = s + ' '
        s = s + '\n'
    print s + '\n'


#input 'good' or 'bad', and an index
#output index of first line with that configuration
def confIndex(gb,i):
    if(gb == 'good'):
        if(not i < C):
            return -1
        else:
            return r+3+i*(r+1)
    elif(gb == 'bad'):
        if(not i < badC):
            return -1
        else:
            return r+3+(C+i)*(r+1)
    else:
        return -1

#read a file with a SAT assignment and return the assignment as a
#list.
def getSATAssignment(satoutFilename):
    g = open(satoutFilename,'r')
    g.readline()#skip the line that says SAT
        #convert input to a list of integers
    csvg = csv.reader(g,delimiter=' ')
    for row in csvg:
        csvg = row
        break
    csvg = map(int,csvg)
    g.close()
    return csvg

def quitError(msg):
    sys.stdout.flush()
    sys.stderr.write(msg)
    sys.stderr.flush()
    sys.exit(1)

s=0 #this is an index that is incremented to label each variable.
    #input region restrictions into good CNF
i=0
j=0
for a in g[2:r+2]:
    for aa in a:
        if( aa in ['X', '.', '#', 'C']):
            s = s + 1
```

```
            X[i][j] = s
            if(aa == 'X'): #disallow this square, or require it
                clause([-X[i][j]])
            if(aa == '#'):
                clause([X[i][j]])
            if(aa == 'C'):
                clause([X[i][j]])
                Cregion[i][j] = 1
            j = (j+1)%c
            if(j == 0):
                i = (i + 1)%r

#get good clauses for H and V requirements
for k in range(C):
    i=0
    j=0
    for a in g[confIndex('good',k):confIndex('good',k)+r]:
        for aa in a:
            if( aa in ['.','<','>','A','V']): #there should be no # in
                                              #this part of the data
                if(j<c-1):
                    s = s + 1
                    HH[k][i][j] = s
                if(i<r-1):
                    s = s + 1
                    VV[k][i][j] = s
                if(aa=='<'): #require this matched horizontal edge (don't
                             #need to record both <>)
                    clause([HH[k][i][j]])
                if(aa=='A'): #require this matched vertical edge
                    clause([VV[k][i][j]])
                j = (j+1)%c
                if(j == 0):
                    i = (i+1)%r

badHH=HH[0] #these cannot be defined earlier
badVV=VV[0]

for k in range(badC):
    i=0
    j=0
    for a in g[confIndex('bad',k):confIndex('bad',k)+r]:
        for aa in a:
            if( aa in ['.','<','>','A','V']):
                if(aa=='<'): #require this matched horizontal edge (don't
                             #need to record both <>)
                    badclause(k,[badHH[i][j]])
                if(aa=='A'): #require this matched vertical edge
                    badclause(k,[badVV[i][j]])
                j = (j+1)%c
                if(j == 0):
                    i = (i+1)%r

for k in range(badC):
    #Build a CNF with no region which forces the bad configuration.
    #Later we enforce a region and check if it is satisfiable.  We can
    #use the same rules as we do below, because those are region
    #independent.
    for i in range(r):
        for j in range(c):
            #Rule 1 note that the (j<c-1) etc are redundant
            if(not (Cregion[i][j] == 1)):
                badclause(k,[-X[i][j],(j<c-1)*badHH[i][j],(i<r-1)*badVV[i][j],(j>0)*badHH[i][j-1],
                             (i>0)*badVV[i-1][j]])
            #Rule 2.  Uncovered squares must not be matched.
            if(j<c-1):
                badclause(k,[X[i][j],-badHH[i][j]])
            if(i<r-1):
                badclause(k,[X[i][j],-badVV[i][j]])
            if(j>0):
                badclause(k,[X[i][j],-badHH[i][j-1]])
            if(i>0):
                badclause(k,[X[i][j],-badVV[i-1][j]])
```

```
            #Rule 3.  Tatami condition. Since this only applies to
            #interior intersections, it also applies to Cregion.  ie
            #at least one of the dimers will be contained among these
            #four squares.
            if(i<r-1 and j<c-1):
                badclause(k,[-X[i][j],-X[i+1][j],-X[i][j+1],-X[i+1][j+1],badHH[i][j],badVV[i][j],badHH[i+1][j],
                              badVV[i][j+1]])
            #Rule 4.  Also applies to Cregion.  I never want to match
            #adjacent edges.  Perhaps redundant, but it simplifies the
            #code.
            if(i<r-1 and j<c-1):
                #  _
                # |
                badclause(k,[-badHH[i][j],-badVV[i][j]])
            if(i>0 and j<c-1):
                #  |_
                badclause(k,[-badHH[i][j],-badVV[i-1][j]])
            if(i>0 and j > 0): #_|
                #  _|
                badclause(k,[-badHH[i][j-1],-badVV[i-1][j]])
            if(i<r-1 and j>0): #
                # _
                #  |
                badclause(k,[-badHH[i][j-1],-badVV[i][j]])
            if(j<c-1 and j>0):
                # _ _
                badclause(k,[-badHH[i][j-1],-badHH[i][j]])
            if(i<r-1 and i>0):
                # |
                # |
                badclause(k,[-badVV[i-1][j],-badVV[i][j]])

#print 'badClauses',badClauses

#CNF for tilings with good clauses
for k in range(C):
    for i in range(r):
        for j in range(c):
            #Rule 1 note that the (j<c-1) etc are redundant
            if(not (Cregion[i][j] == 1)):
                clause([-X[i][j],(j<c-1)*HH[k][i][j],(i<r-1)*VV[k][i][j],(j>0)*HH[k][i][j-1],(i>0)*VV[k][i-1][j]])
            #Rule 2
            if(j<c-1):
                clause([X[i][j],-HH[k][i][j]])
            if(i<r-1):
                clause([X[i][j],-VV[k][i][j]])
            if(j>0):
                clause([X[i][j],-HH[k][i][j-1]])
            if(i>0):
                clause([X[i][j],-VV[k][i-1][j]])

            #Rule 3
            if(i<r-1 and j<c-1):
                clause([-X[i][j],-X[i+1][j],-X[i][j+1],-X[i+1][j+1],HH[k][i][j],VV[k][i][j],HH[k][i+1][j],
                        VV[k][i][j+1]])
            #Rule 4
            if(i<r-1 and j<c-1):
                #  _
                # |
                clause([-HH[k][i][j],-VV[k][i][j]])
            if(i>0 and j<c-1):
                #  |_
                clause([-HH[k][i][j],-VV[k][i-1][j]])
            if(i>0 and j > 0): #_|
                #  _|
                clause([-HH[k][i][j-1],-VV[k][i-1][j]])
            if(i<r-1 and j>0): #
                # _
                #  |
                clause([-HH[k][i][j-1],-VV[k][i][j]])
            if(j<c-1 and j>0):
                # _ _
                clause([-HH[k][i][j-1],-HH[k][i][j]])
```

```
            if(i<r-1 and i>0):
                # |
                # |
                clause([-VV[k][i-1][j],-VV[k][i][j]])



satoutFilename = sys.argv[2][:-4] + 'out.txt'
satinFilename = sys.argv[2]
badsatinFilename = '__tmp.txt'
badsatoutFilename = '__tmpout.txt'


#DO A SINGLE WRITE FOR ALL GOOD CLAUSES
CNFstring = 'p cnf ' + str(nGoodVars) + ' ' + str(len(goodClauses)) + ' \n'
for _clause in goodClauses:
    for lit in _clause:
        CNFstring = CNFstring + ' ' + str(lit)
    CNFstring = CNFstring + ' 0\n'
f = open(satinFilename, 'w')
f.write(CNFstring)
f.close()


#make a single string for each set of bad clauses (to which we later
#append the region R)
badCNFstring = []
for k in range(badC):
    badCNFstring.append('p cnf ' + str(nBadVars) + ' ' + str(len(badClauses[k]) + r*c) + ' \n')
    for _clause in badClauses[k]:
        if(_clause == []):
            quitError('empty clause' + str(badClauses[k]))
        for lit in _clause:
            #I used [k] instead of [-1] so we would have one more
            #chance to throw an exception if something goes amiss.
            badCNFstring[k] = badCNFstring[k] + ' ' + str(lit)
        badCNFstring[k] = badCNFstring[k] + ' 0\n'

print "X",X
print "HH",HH
print 'VV',VV
print "badHH",badHH
print 'badVV',badVV
print 'CNFstring', CNFstring
print 'badClauses', badClauses
print 'badCNFstring', badCNFstring
print 'badCNFstring[0]',badCNFstring[0]

#we use this to negate the X_i literals of the region R.
neg = lambda x: -x

numRegions = 0 #count the number of regions we have tried
prevR = []
while(True):
    numRegions += 1
    sp =
    subprocess.Popen(['./minisat',satinFilename,satoutFilename],
                     stdout=subprocess.PIPE)
    sp.wait()
    if(numRegions%100 == 0):
        print "number of regions checked", numRegions
    if(sp.returncode==10): #satisfied
        g = getSATAssignment(satoutFilename)
        R = g[:r*c] #the region output from last minisat of f
        if(prevR == R):
            quitError('error: two regions the same')
        if(numRegions%100 == 0):
#            print R
            displayRegion(R)
            print "good configurations"
            for k in range(C):
                displayTiling(g,k)
        prevR = R
```

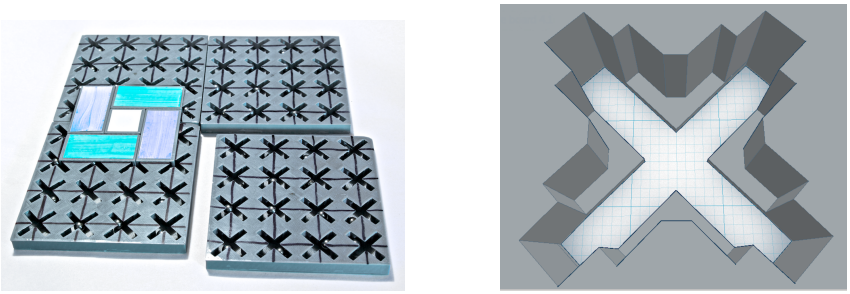
```
        rClauses = '' #make clauses to enforce that region
        for _clause in R:
            rClauses = rClauses + str(_clause) + ' 0\n'
        badFlag = False
        for k in range(badC):
            #for each bad configuration, check if it can be completed
            #in the region R.
            badConfig = open(badsatinFilename,'w')
            badConfig.write(badCNFstring[k] + rClauses)
            badConfig.close()
            sp = subprocess.Popen(['./minisat',badsatinFilename,badsatoutFilename],stdout=subprocess.PIPE)
            sp.wait()
            if(sp.returncode==10):
                badFlag = True
                if(numRegions%100==0):
                    print 'bad configuration'
                    displayTiling(getSATAssignment(badsatoutFilename),0)
                break
            elif(sp.returncode != 20):
                quitError('bad minisat returned bad code: ' + str(sp.returncode))
        if(badFlag == False):
            #we have found a good region!
            print "HORRAY", R
            sys.stdout.flush()
            sys.exit(0)

       #we are going to append a forbidden region to satinFilename
        f = open(satinFilename,'r+')
        #change the first line with the number of clauses
        f.seek(0,0)
        f.write('p cnf ' + str(nGoodVars) + ' ' + str(len(goodClauses)) + '\n')
        #make a clause from the forbidden region
        clause(map(neg,R))
        CNFstring = ''
        for lit in goodClauses[-1]:
            CNFstring = CNFstring + ' ' + str(lit)
        CNFstring = CNFstring + ' 0\n'
        #append this to the end of the file
        f.seek(0,2)
        f.write(CNFstring)
        f.close()
    elif(sp.returncode != 20):
        quitError('good minisat returned bad code: ' + str(sp.returncode))
    else:
        sys.stdout.write('There is no region that satisfies the input.')
        sys.stdout.flush()
        sys.exit(0)
```

A.3 Tatami Maker: a combinatorially rich mechanical game board

The Tatami Maker game board forms a rectilinear grid that enforces the tatami restriction when it is covered by the accompanying tile pieces. Arbitrary rectilinear grids can be created by placing Tatami Maker's modules alongside each other (see Figure A.1(a)). A simple mechanism is embedded at each grid line intersection, which obstructs the placement of a tile if the other three incident grid squares are covered (see Figure A.1(b)).

A tile piece covers one or two grid squares, and the underside of each of its four corners has the specially shaped foot shown in Figure A.2(b). The feet



(a) Tatami Maker modules are placed alongside each other to create larger grids.

(b) The inside of Tatami Maker's intersection mechanism. Each arm of the X-shape extends to a different grid square.

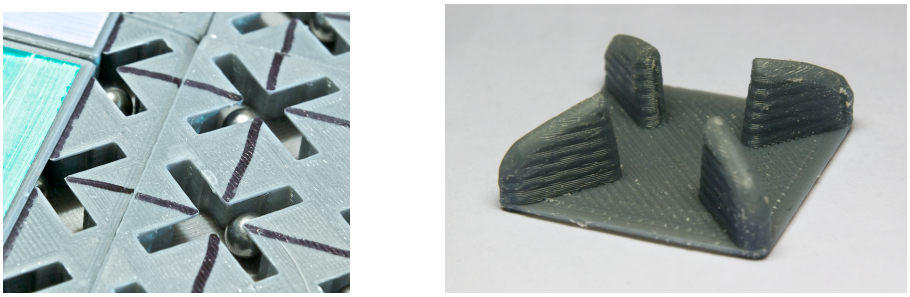
Figure A.1: Tatami Maker modules and mechanism.

interact with the mechanism in the game board by pushing an obstructing ball onto an unoccupied grid square, as well as guiding the tile correctly onto the grid.

Each mechanism occupies one grid intersection, which consists of one quadrant from each of four incident grid squares. A cavity in the game board contains a ball which may travel freely to any unoccupied quadrant. If a quadrant is occupied by a tile's corner, then one of tile's feet occupies the part of the cavity that is otherwise available to the ball. When three of the quadrants are occupied by tile corners, the ball is forced to occupy the remaining quadrant, thereby preventing a tile corner from being placed here.

A minimal game board module is one grid intersection. As a result, the dimensions of a module are given in terms of its intersections rather than its grid squares. Our prototype's modules are 4×4, but they need not be square, or even rectangular.

The main design challenge is ensuring that the ball can be pushed by an incoming foot, unobstructed, to an available quadrant. We label the quadrants Q1, Q2, Q3, and Q4, in counterclockwise order. A critical case occurs when Q2 and Q4 are occupied by tile corners (and feet), and a tile foot placed in Q1 must push the ball to Q3. Intuitively, the ball must disturb the feet in Q2 and Q4 on the way to Q3, otherwise the midpoint of the intersection would accommodate the ball when all four quadrants are occupied. The mechanism is designed, therefore, so that the ball lifts the tiles in Q2 and Q4 as it passes to Q3, and so that it will not become stuck against another part of the mechanism before it arrives in Q3.



(a) (b)

Figure A.2: *(a)*, Grid intersections and several Tatami Maker modules. *(b)*, The feet of a tile piece.

The availability of desktop 3D printers has lowered the cost of creating prototypes sufficiently enough that rough estimation and iterative trial and error was the most economical way of solving these design challenges. Tatami Maker was prototyped with a Solidoodle 2 printer; the printer is a 3D CNC machine that extrudes a filament of hot ABS plastic into a $15 \times 15 \times 15$ cm print area to create real-life versions of virtual 3D models. With each iteration of the prototype, changes to the shape of the cavity and the feet of the tile pieces were made to tease out the required behaviour of the mechanism.

Tatami Maker was presented in [9] at Bridges, 2013.

$n\backslash z^k$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
2	1																			
3	1	2																		
4	1	1	2																	
5	1	1	2	4	0	2														
6	1	0	1	2	2	-2	2													
7	1	0	1	2	2	4	-2	4	0	2	-2	2								
8	1	0	1	1	2	3	4	-2	2	0	4	-2	2	-2	2					
9	1	0	1	1	2	3	4	6	-2	6	0	8	-2	4	-4	6	-2	4	-2	2
10	1	-1	1	0	1	1	1	2	2	-6	6	-2	6	-6	4	-4	6	-6	6	-4
11	1	-1	1	0	1	1	1	2	2	4	-8	10	-4	10	-8	8	-8	10	-10	12
12	1	-1	1	0	1	0	2	1	2	3	3	-7	8	-2	6	-6	12	-10	10	-14
13	1	-1	1	0	1	0	2	1	2	3	3	5	-8	12	-4	12	-10	20	-14	16
14	1	-1	1	-1	2	-1	2	0	2	1	2	3	3	-9	7	-4	16	-14	18	-16
15	1	-1	1	-1	2	-1	2	0	2	1	2	3	3	5	-11	12	-8	24	-20	28
16	1	-1	1	-1	2	-1	2	-1	3	0	3	1	4	3	5	-10	9	-4	19	-15
17	1	-1	1	-1	2	-1	2	-1	3	0	3	1	4	3	5	6	-11	14	-7	27
18	1	-2	2	-2	3	-3	3	-3	4	-3	3	-2	3	-1	2	1	1	-15	21	-14
19	1	-2	2	-2	3	-3	3	-3	4	-3	3	-2	3	-1	2	1	1	3	-19	28
20	1	-2	2	-2	3	-3	3	-3	4	-4	5	-4	5	-4	5	-2	4	-1	4	-19
$n\backslash z^k$	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
9	-2	2																		
10	4	-4	2																	
11	-8	10	-12	10	-6	6	-6	6	-4	4	-4	2								
12	16	-12	14	-14	16	-14	12	-12	12	-10	10	-8	6	-4	4	-4	2			
13	-18	22	-20	26	-22	26	-24	26	-28	28	-24	26	-24	24	-20	20	-20	18	-14	12
14	18	-26	26	-24	34	-36	38	-36	40	-44	42	-40	42	-42	40	-34	36	-36	30	-26
15	-24	30	-36	38	-34	44	-48	56	-50	60	-66	66	-68	72	-74	78	-70	76	-80	76
16	19	-17	32	-32	34	-40	50	-56	62	-62	80	-80	84	-90	98	-104	106	-108	118	-120
17	-21	31	-26	48	-44	52	-54	70	-74	82	-82	108	-102	116	-124	138	-144	152	-160	176
18	26	-36	37	-38	54	-68	70	-76	92	-108	120	-128	154	-176	186	-198	220	-242	252	-262
19	-22	38	-51	56	-58	78	-94	100	-108	128	-144	156	-164	190	-214	236	-250	276	-308	328
20	26	-17	30	-40	41	-41	57	-71	90	-100	112	-139	162	-168	194	-234	270	-290	326	-364
$n\backslash z^k$	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
13	-10	10	-8	6	-4	4	-4	2												
14	26	-24	20	-16	14	-12	10	-6	4	-4	2									
15	-70	72	-72	68	-62	60	-58	56	-46	42	-42	36	-28	26	-24	20	-16	14	-12	10
16	120	-116	124	-122	116	-116	116	-110	104	-98	96	-90	82	-74	72	-64	54	-50	46	-38
17	-182	192	-190	208	-208	212	-218	224	-224	224	-220	224	-220	216	-208	210	-200	190	-184	178
18	284	-306	318	-322	340	-356	358	-362	376	-386	382	-376	380	-382	372	-358	356	-352	334	-316
19	-346	378	-414	438	-450	484	-516	536	-552	580	-612	626	-632	654	-678	686	-684	698	-712	710
20	398	-426	460	-514	562	-592	636	-688	732	-764	810	-860	904	-934	968	-1014	1046	-1062	1090	-1126
$n\backslash z^k$	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
15	-6	4	-4	2																
16	32	-28	24	-18	14	-12	10	-6	4	-4	2									
17	-166	156	-146	138	-126	116	-106	98	-86	78	-70	60	-52	46	-38	32	-28	24	-18	14
18	308	-296	274	-254	242	-228	206	-186	174	-158	138	-122	110	-98	84	-70	60	-52	42	-32
19	-700	702	-704	690	-670	662	-656	634	-606	592	-576	548	-516	492	-472	444	-410	386	-366	336
20	1148	-1154	1168	-1182	1188	-1180	1174	-1178	1170	-1144	1126	-1112	1084	-1048	1016	-986	952	-906	864	-830
$n\backslash z^k$	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
17	-12	10	-6	4	-4	2														
18	26	-22	16	-10	8	-6	2													
19	-302	278	-260	234	-204	186	-170	146	-124	110	-98	84	-70	60	-52	42	-32	26	-22	16
20	788	-736	692	-654	608	-560	520	-480	438	-398	360	-326	294	-258	228	-204	176	-150	132	-114
$n\backslash z^k$	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
19	-10	8	-6	2																
20	94	-78	66	-54	42	-32	26	-22	16	-10	8	-6	2							

Table A.22: Table of coefficients of $P_n(z)$ for $2 \leqslant n \leqslant 20$. It is irreducible for $2 \leqslant n < 200$.