COUNTING TILINGS OF FINITE LATTICE REGIONS

BRANDON ISTENES

ABSTRACT. We survey existing methods for counting two-cell tilings of finite planar lattice regions. We provide the only proof of the Pfaffian-squared theorem the author is aware of which is simultaneously modern, elementary, and fully correct. Detailed proofs of Kasteleyn's Rule and Theorem are provided. Theorems of Percus and Gessel-Viennot are proved in the standard fashions, hopefully with some advantage in clarity. Results of Kuperberg's are broken down and proved, again with significantly greater detail than we were able to find in a review of existing literature. We aim to provide a cohesive and compact, if bounded, examination of the structures and techniques in question.

CONTENTS

1. INTRODUCTION

Tiling problems are some of the most fundamental types of problems in combinatorics, and have only with the appearance of the field come under serious study. Even still, solid results on the subject are hard to come by, especially in light of the ease typical in treating them with numerical approximation and exhaustive methods. Three branches of questions arise naturally, given a set of tiles:

- Can we tile a given region?
- How many ways can we tile a given region?
- How do those ways relate to each other?

From these spring many more, related questions:

- Can we tile this region with a smaller set of translation-equivalent tiles?
- Can we tile this region if we change the boundary conditions?
- What mutilations of this region will allow it to remain tileable?

- (The corresponding counting questions to all of the above)
- Does a closed-form exist for counting tilings?
- What is the structure of the quotient space of tilings under various isometry groups?
- Does there exist a finite set of permutations of tiles via which the space of tilings is connected?
- What sizes of regions cannot be tiled by concatenated tilings of smaller regions (i.e., what regions do not admit tilings with fault lines)?
- What local probability distributions describe the space of tilings? (c.f. Arctic Circle Theorem)
- What is the Boltzmann Distribution of the space of tilings?

Of course, due to the close ties between combinatorics and computer science, almost all of these imply a follow-up question: How fast can we find out? In this paper, we restrict our attention to questions on the second branch: those of counting. We will explore three methods, in the order that they were discovered. The first, found by Pieter Kasteleyn in 1967, works for all two-cell tilings of planar graphs. The second is Jerome Percus's 1969 specialization of Kasteleyn's method, which, though it is less powerful, is significantly easier to deal with. The final method is due to Ira Gessel and Laurent Viennot in 1985, adapting a 1973 theorem by Bernt Lindström. The theorem in full generality is due, however, to Samuel Karlin and James McGregor in 1959, where they proved it in the context of probability. We will then prove the equivalence of the Kasteleyn and Gessel-Viennot methods, a result due to Greg Kuperberg in 1998.

2. Preliminaries

Working knowledge of linear algebra and group theory is assumed.

2.1. **Graph Theory.** A *graph* $G = (V, \hat{w})$ is a set of elements we call *vertices* $v_i \in V$ and a function $\hat{w}$ mapping ordered pairs of vertices to elements of a set, usually $\mathbf{R}$. $\hat{w}$ is the weight function, and we say that G has an *edge* from v_i to v_j, denoted (v_i, v_j) wherever $\hat{w}(v_i, v_j)$ is nonzero. We may occasionally refer to the set of these edges as $E \in G$. If $\hat{w}$ is symmetric, we call G an undirected graph and use (v_i, v_j) to denote that directed edge and its reverse, taken together as a single undirected edge. If $\hat{w}$ is asymmetric, then we call G a *directed graph*, or *digraph* for short. We will never have undirected edges in a digraph. We call an assignment of direction to the edges of an undirected graph G an *orientation* of G, resulting in the digraph $G^{\rightarrow}$. We call the map from a digraph to a graph which makes all the edges undirected a *disorientation*. We say if there exists an edge (v_i, v_j) or (v_j, v_i) that v_i and v_j are *adjacent* and each is *incident* on the edge. The number of edges incident on a vertex is termed the *degree* of that vertex. A graph is *finite* if the set of its vertices is finite.

We define a *path* in G as an ordered set of vertices $P = (a_1, a_2, \ldots, a_k)$ such that $\forall i, 1 \le i \le k, \hat{w}(a_i, a_{i+1} \neq 0$. We define a *circuit* in G as a path for which $a_k = a_1$, the repeated start vertex being omitted when we represent it as a tuple. Note that this is in literature sometimes referred to as a cycle, but we use the circuit terminology to avoid collision with the notion of a cycle permutation, which will be important to us. We define the *weight of a (path/circuit) P* by

$$\mathrm{wgt}(P) = \prod_{i=1}^{k-1} \hat{w}(a_i, a_{i+1})$$

and the *length of a (path/circuit) P* as the total number of unique edges in P, not counting edges (v_i, v_i), called *loops*. A (di)graph G is *(strongly) connected* if there exists a path between any two vertices in G. We say that a graph is a *tree* if its disorientation contains no circuits, and we call all of its vertices of degree one *leaves*.

An *embedding* of a graph G on a surface S is a representation of G such that vertices in G are points in S and there is a simple arc connecting two points in S if and only if those points are adjacent in G, and

no two arcs intersect except at those points representing vertices. We consider two embeddings *equivalent* if their respective systems of arcs are homeomorphic.

A graph is *planar* if it admits an embedding into $\mathbf{E}^2$, and we call its embedding the *plane graph* of G, which we will also refer to as G. This permits us to define the *faces* of G as the regions in $\mathbf{E}^2$ separated by the Jordan curves made by the arcs. We define an *outside face* as a face that is unbounded, which for a finite graph is unique, and we will therefore henceforth use the definite article for them. Accordingly, *interior faces* are those that are not the outside face.

The *adjacency matrix* of a graph G with n vertices is the $n \times n$ matrix A where

$$a_{i,j} = \hat{w}(v_i, v_j)$$

.

We will in general mean by a graph G a finite, connected, undirected plane graph. Unless otherwise specified, all weight functions $\hat{w}$ should be assumed to be binary-valued.

2.2. **Tiling.** Given a plane graph G, we can join the faces, or in this context, *cells* of G into larger partitions of the region of the plane that G occupies. If each face of G has, in a copy of itself G', been joined with at least one other, we call G' a *tiling* of G. We call each face of G' a *tile* of G. Of course, this isn't a particularly useful notion for building tilings, so we define the set of *prototiles* as the set of equivalence classes of tiles under translation. In other works, equivalence under other rigid body motions is considered, but it makes no difference for us. We simply have larger sets of prototyles. This notion of a prototile enables us to ask more interesting questions about tilings of graphs which exhibit enough regularity for the set of prototiles to be much smaller than half the number of faces. In all of the interesting problems that the author is aware of, the set of prototiles is so limited. We focus on the sets of prototiles composed of two cells and invariant under rotation.

Clearly, our notion of a tiling is formally inextricabe from that of a tile. However, we will consider tiles as objects on their own that can be moved around, rearranged, placed, and picked up, regardless of the status of the underlying figure. The question of counting tilings is formally equivalent to, "Given a graph G, how many ways can we join the faces of G such that the equivalence class of faces under translation is equal to the set of tiles T?" We will, of course, phrase and approach the problem in its more intuitive form.

2.3. **Lattices.** While many of the results we will prove are both effective and useful on a much wider range of graphs, we will be focusing on *lattice graphs*, or simply *lattices*. A lattice graph is a plane graph with a highly regular structure. In particular, the vertices of a lattice graph are the integer multiples of some set of basis vectors. This definition is very restrictive. As it turns out, there are exactly three similarity classes of lattice graphs: the triangle lattice, the square lattice (or "grid"), and the hexagon lattice. Extrapolation of our results to other types of highly regular graphs, such as uniform tessellations, is very straightforward. A *lattice region* is simply a finite portion of a lattice.

3. The Hafnian-Pfaffian Method

Definition 3.1. We define the *geometric dual graph*, or just *dual graph*, of a plane graph G as the unique graph G^* with a vertex for each face of G and an edge connecting two vertices of G^* if and only if the two corresponding faces of G share an edge. It is easy to verify that G^{**} is equivalent to G. We additionally observe that a square lattice chunk is its own dual, and chunks of the triangle and hexagon lattices are dual each other.

Definition 3.2. We define the *truncated dual graph* as the dual graph G^* with the vertex corresponding to the outer face of G and all edges connecting to it removed.

Definition 3.3. We define a *perfect matching* on G as, for E the set of edges in the disorientation of G, a subset $M \subset E$ such that each vertex is incident on exactly one edge in M.

The first method for counting two-cell tilings on finite lattice regions comes from the observation that these are in bijection with perfect matchings on the truncated dual graph.
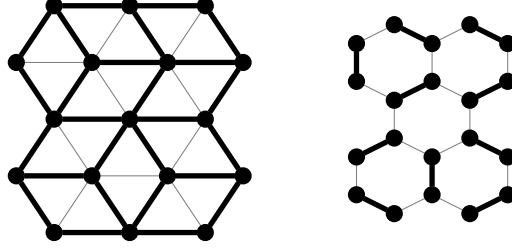

FIGURE 1. A tiling of a graph and corresponding perfect matching of its dual.

We introduce the *Pfaffian*, a function of a matrix that in its definition explicitly pairs entries.

Definition 3.4. A matrix is *skew-symmetric* (alternatively, *antisymmetric*) if it is equal to its negative transpose, i.e., B is skew-symmetric if $B = -B^T$.

Definition 3.5. For an $n \times n$ skew-symmetric matrix B, the Pfaffian is defined as

$$\text{Pf}(B) = \sum_{M \in \text{PM}(n)} \text{sgn}(M) \prod_{(i,j) \in M} b_{i,j}$$

where $\text{PM}(n)$ is defined as the set of ordered disjoint pairings (j,k) where $1 \le j < k \le n$. We define sign function by mapping $\text{PM}(n)$ to the symmetric group of order n:

$$\text{sgn}(M) = \text{sgn}(\phi(M))$$

where $\phi : \text{PM}(n) \to S_n$ is defined by

$$((i_0, j_0), (i_1, j_1), \cdots, (i_{n/2}, j_{n/2})) \mapsto (i_0, j_0, i_1, j_1, \cdots, i_{n/2}, j_{n/2})$$

Definition 3.6. We define a *simple alternating circuit* C over a perfect matching M on G as a circuit oriented counterclockwise on the disorientation of G, containing no repeated vertices, and for which every other edge is contained in M; i.e.,

$$\forall (v_i, v_j) \in C, \quad (v_i, v_j) \in M \ \text{ or } \ (v_j, v_i) \in M$$

. We count a single edge in M as a simple alternating circuit, and observe that every simple alternating circuit must have an even number of vertices.

Definition 3.7. We observe that the action of the k-cycle on a circuit D of length k advances each match in M counterclockwise once along D. We will refer to this as the *circuit permutation* of D, and denote it

$$\sigma_D : PM(G) \to PM(G)$$

We denote by $\text{PM}(G)$ the set of perfect matchings on G, and justify the abuse of notation by noting that in all of our uses of it, matches in $\text{PM}(|G|)$ which do not correspond to a match in $\text{PM}(G)$ will vanish.

Lemma 3.8. *For an arbitrary graph G, $\text{PM}(G)$ is connected under circuit permutations, and all matchings of any $M \in \text{PM}(G)$ are contained in simple alternating circuits.*

Proof. Take M and M' to be two arbitrary perfect matchings in $\mathrm{PM}(G)$ such that $M \neq M'$, and take ϕ to be the map such that $\phi(M) = M'$. For each matching $\{v_i, v_j\} \in M$, we have one of two cases. In the case that $\{v_i, v_j\} \in M'$, we trivially have a simple alternating circuit on two vertices. Now we consider the case where ϕ sends (v_1, v_2), elsewhere. But v_1 and v_2 must now be connected to other vertices, as, say, (v_1, v_4), (v_2, v_3). But v_3 and v_4 must have previously been connected to other vertices. Either they were connected to each other as (v_3, v_4), in which case we have the circuit permutation

$$\sigma : ((v_1, v_2), (v_3, v_4)) \mapsto ((v_4, v_1), (v_2, v_3)) ,$$

or else they connected elsewhere as, say, (v_3, v_5) and (v_4, v_6). Clearly we are growing two alternating paths, and we leave it as an exercise to show that that this process always terminates with a pairing between them.

Thus we have shown that any mapping between two perfect matchings can be decomposed into circuit permutations. We note that there is an obvious isomorphism between $PM(G)$ under simple alternating circuits and $PM(n)$ under the corresponding permutations on n elements. $\square$

Theorem 3.9. *For any skew-symmetric matrix B of even dimension,*

$$\det(B) = \mathrm{Pf}(B)^2$$

Proof. Following [7]:

Assume B is a skew-symmetrix matrix of even dimension n. By the Leibniz determninant formula,

$$\det(B) = \sum_{\pi \in S_n} \mathrm{sgn}(\pi) \prod_{i=1}^{n} b_{i,\pi(i)} \tag{3.1}$$

Using the definition of the Pfaffian, we find that the Pfaffian squared can be brought together so that the sum and product is over the square of the sets used in the Pfaffian itself. This allows us to decompose the set of edges from the Cartesian product of two perfect matchings into those that are in only one matching and those that are in both.

$$
\begin{aligned}
\mathrm{Pf}(B)^2 &= \left(\sum_{M \in \mathrm{PM}(n)} \mathrm{sgn}(M) \prod_{(i,j) \in M} b_{i,j} \right) \left(\sum_{N \in \mathrm{PM}(n)} \mathrm{sgn}(N) \prod_{(l,m) \in N} b_{l,m} \right) \\
&= \sum_{(M,N) \in \mathrm{PM}(n)^2} \mathrm{sgn}(M)\,\mathrm{sgn}(N) \prod_{(i,j) \in M} b_{i,j} \prod_{(l,m) \in N} b_{l,m} \\
&= \sum_{(M,N) \in \mathrm{PM}(n)^2} \mathrm{sgn}(M)\,\mathrm{sgn}(N) \prod_{((i,j),(l,m)) \in M+N} b_{i,j} b_{l,m} \\
&= \sum_{(M,N) \in \mathrm{PM}(n)^2} \mathrm{sgn}(M)\,\mathrm{sgn}(N) \prod_{(i,j) \in M \bigcup N} b_{i,j} \prod_{(l,m) \in M \bigcap N} b_{l,m} \tag{3.2}
\end{aligned}
$$

We now relate these equations by relating $\mathrm{PM}(n)$ to S_n in such a way that cancels some terms of Equation 3.2 and allows us to equate the rest to Equation 3.1. Observe that any element $\pi \in S_n$ can be decomposed into disjoint cycles, and that this decomposition can be again separated into the product of a set of even cycles and a set of odd cycles. Suppose that π_1 is a permutation in S_n which contains an odd cycle σ. Consider the equivalence class $S_n / \sigma^i S_n$ of permutations in S_n which are equal, but with σ cyclically permuted $1 \leq i \leq |\sigma|$ times. σ is of odd length and therefore even sign, so $\mathrm{sgn}(\pi_k) = \mathrm{sgn}(\pi) \quad \forall \pi_k \in S_n / \sigma S_n$. We write, for brevity, $\pi_k = \sigma^k \pi$. The permutation π_0 is the unique permutation in $S_n / \sigma S_n$ for which $\exists i \in \sigma$ such that $i = \pi_0(i)$. Since B's skew-symmetry requires that it is zero along the main diagonal, this term of the determinant is equal to zero. This leaves us with an even number of permutations in $S_n / \sigma S_n$ which can yield a nonzero

term in the determinant. We consider these permutations in disjoint pairs (π_k, π_{k+1}). We observe that $\{(i, \pi_k(i)) : i \in \sigma\} = \{(\pi_{k+1}(j), j) : j \in \sigma\}$, and therefore, since $b_{i,j} = -b_{j,i}$,

$$\operatorname{sgn}(\pi_k) \prod_{i=1}^{n} b_{i,\pi_k(i)} = -\operatorname{sgn}(\pi_{k+1}) \prod_{j=1}^{n} b_{j,\pi_{k+1}(j)}$$

Thus, for skew-symmetric B, all terms of the determinant corresponding to a permutation in S_n that contains an odd cycle in its decomposition cancel each other out. We additionally can eliminate all permutations $\pi \in S_n$ for which $\exists i, 1 \leq i \leq n$ such that $i = \pi(i)$, since these include entries on the diagonal.

It remains to be shown that the remaining terms, whose permutations' disjoint cycle decompositions do not contain cycles of odd length nor identity cycles, are equal to the terms of $Pf(B)^2$. We now pair elements of PM(n). By Lemma 3.8, any two perfect matchings $M, N \in$ PM(n) are connected by circuit permutations on a set of circuits covering all of their matches. Thus for all $i \in 1, \cdots, n$, $M \bigcup N$ has a pair (i, j) for some j. Each of these circuits is represented in S_n by a cycle. Since the circuits are of even length, all the representative cycles are of even length. Thus every element in $M \bigcup N$ is a permutation of the elements $1, \cdots, n$, whose disjoint cycle decomposition contains only even cycles, which is exactly the nonvanishing part of S_n, and $M \bigcap N$ is empty. Thus,

$$\operatorname{Pf}(B)^2 = \sum_{(M,N) \in \operatorname{PM}(n)^2} \operatorname{sgn}(M) \operatorname{sgn}(N) \prod_{(i,j) \in M \bigcup N} b_{i,j} \prod_{(l,m) \in M \bigcap N} b_{l,m}$$

$$= \sum_{(M,N) \in \operatorname{PM}(n)^2} \operatorname{sgn}(M) \operatorname{sgn}(N) \prod_{(i,j) \in M \bigcup N} b_{i,j}$$

$$= \sum_{\pi \in S_n} \operatorname{sgn}(\pi) \prod_{i=1}^{n} b_{i,\pi(i)}$$

$$= \det(B)$$

$\square$

Corollary 3.10. *The Pfaffian vanishes on matrices of odd dimension.*

Proof. Assume B is a skew-symmetric matrix of odd dimension. Then

$$\det(B) = \det(B^T) = \det(-B) = (-1)^n \det(B) = -\det(B)$$

$\square$

Theorem 3.9 is central to the Pfaffian's usefulness to us. It provides a link between a measurement of the combinations of a set of elements into pairs and a determinant, the latter of which is efficiently computable.

Example 3.11. We manually compute the Pfaffian of

$$B = \begin{bmatrix} 0 & a & b & c \\ -a & 0 & d & e \\ -b & -d & 0 & f \\ -c & -e & -f & 0 \end{bmatrix}$$

We begin by enumerating PM(4):

$$\operatorname{PM}(4) = \{\{(1,2),(3,4)\}, \{(1,3),(2,4)\}, \{(1,4),(2,3)\}\}$$

We find the sign of each pairing:

$$\text{sgn}((1,2,3,4)) = 1$$
$$\text{sgn}((1,3,2,4)) = -1$$
$$\text{sgn}((1,4,2,3)) = 1$$

Our result, then, is

$$\text{Pf}(B) = b_{1,2}b_{3,4} + (-1)b_{1,3}b_{2,4} + b_{1,4}b_{2,3}$$
$$= af - be + cd$$

Thus, by summing over a function of the unordered disjoint pairings of integers up to n, the Pfaffian has a term for each set of positive entries which have no repeated row or column index. On a suitable relative of the adjacency matrix, then, we expect the Pfaffian to have a term for each perfect matching on the corresponding graph.

Definition 3.12. The *Kasteleyn Matrix* of a graph G is the skew-symmetric matrix K_G such that

$$K_G = A_G - A_G^T$$

where A_G is the adjacency matrix of G.

We note that K_G vanishes on undirected G, for which A_G is symmetric, and so we will here always be talking about K_G for directed graphs. By our observation before, $Pf(K_G)$ will have a term for each perfect matching on G. In order to actually count those perfect matchings, however, we need all of those terms to have the same sign. Then we will have $Pf(K_G)$ equal to the number of perfect matchings, and via Theorem 3.9, we will be able to compute it easily. We note that we are searching for conditions under which the Pfaffian will be equal to the *Hafnian*, defined as

$$\text{Pf}(B) = \sum_{M \in \text{PM}(n)} \prod_{(i,j) \in M} b_{i,j}$$

Observing that matchings on a graph are independent of the direction of the edges, and that switching the direction of an edge switches the sign of the corresponding entry of K, we introduce the following notions.

Definition 3.13. A *Pfaffian orientation* (alternatively a *Kasteleyn-flat orientation*, *admissable orientation*) of G is an orientation of G such that all terms of $\text{Pf}(K_G)$ have the same sign.

Definition 3.14. We say a circuit C in G is *oddly oriented* or has *odd orientation* if it has an odd number of edges oriented counterclockwise, or for a simple alternating circuit over M on G, if an odd number of edges it covers in G are oriented counterclockwise.

We now pave the way to a critical theorem due to Kasteleyn with a string of lemmas.

Sublemma 3.15. *Let e, v, f be, respectively, the edges, vertices, and faces on the interior of a simple circuit. Then*

$$e = v + f - 1$$

Proof. Let C be an simple circuit on a plane graph G. Let v, e, and f be defined as above. We contract adjacent pairs of vertices in C into single vertices, and allow G in this process to lose its planarity. Let W be the set of vertices not in C that are adjacent to vertices in C. When we contract two vertices v_i, v_j, each pair of edges connecting v_i and v_j to a vertex in W is combined into a single edge. This operation reduces e and f each by the number of vertices to which v_i and v_j are mutually adjacent. Repeat this process until C has been pinched down to a single vertex, pulling G into a convex polyhedron, say G_C.
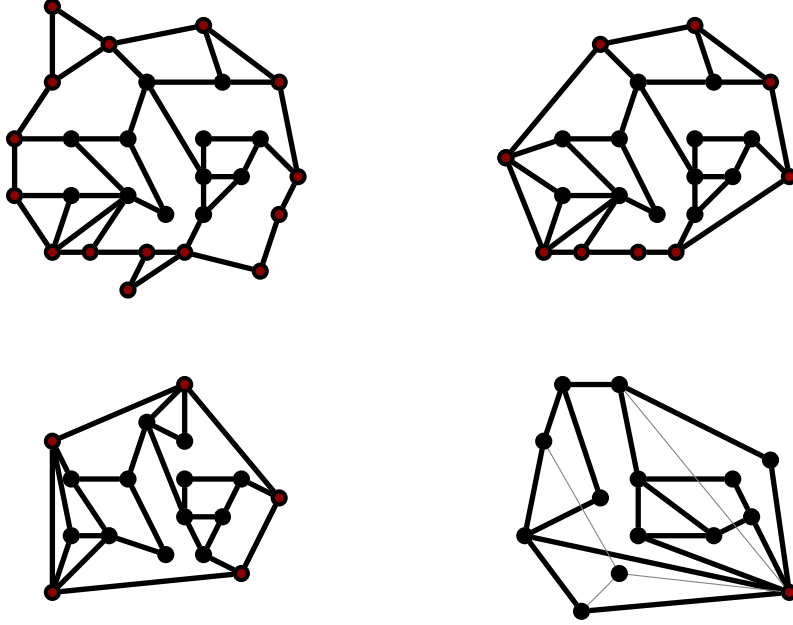
FIGURE 2. Top-left, an arbitrary graph. Bottom right, its contraction into a convex polyhedron.

e and f have been reduced to e_C and f_C respectively, which are now the total number of edges and faces on the polyhedron, since C is a single vertex. The number of vertices not in C is the same for G and G_C, so the total number of vertices in G_C is equal to $v + 1 =: v_C$. By Euler's polyhedron formula, we have

$$
\begin{aligned}
& e_C = v_C + f_C - 2 \\
\Rightarrow \quad & e - |D| + 1 = (v + 1) + (f - |D| + 1) - 2 \\
\Rightarrow \quad & e - |D| + 1 = v + f - |D| \\
\Rightarrow \quad & e = v + f - 1
\end{aligned}
$$

$\square$

Sublemma 3.16. *Simple alternating circuits on plane graphs have an even number of interior vertices.*

Proof. Assume C is a simple alternating circuit on a plane graph. Since all vertices contained in C are matched among each other, all vertices not in C must be matched with each other. C divides these vertices into two regions: those interior and exterior to C. It is easy to see that those interior to C cannot be adjacent those exterior to C, and thus they must be matched among each other, which implies that there is an even number of them. $\square$

Lemma 3.17. *For a plane graph G, if $G^{\rightarrow}$ is oriented so that each face is oddly oriented, then every simple alternating circuit in G is oddly oriented.*

Proof. We demonstrated in Lemma 3.8 that matchings on G decompose to disjoint collections of simple alternating cycles. For an arbitrary decomposition, consider each cycle C. Let:

- c_i be the number of edges oriented counterclockwise on the boundary of the C's interior face i
- c be the number of edges oriented counterclockwise on the boundary of C, and
- f be the number of faces on the interior of C.

Assume c_i is odd for all faces. Then $c_i \equiv 1 \pmod 2$, and therefore

$$f = \sum_{i=1}^{f} c_i \pmod 2$$

Any orientation of the edges on the interior of C causes all of them to be counted as a counterclockwise edge on the boundary of some face C_i. Therefore

$$\sum_{i=1}^{f} c_i = c + e$$

$$\Rightarrow \quad f \equiv c + e \pmod 2$$

Applying Sublemma 3.15 we obtain

$$f \equiv c + (v + f - 1) \pmod 2$$

$$\Rightarrow \quad c \equiv (v - 1) \pmod 2$$

And since $v \equiv 0 \pmod 2$ by Sublemma 3.16,

$$c \equiv 1 \pmod 2$$

$\square$

Theorem 3.18 (Kasteleyn's Rule). *For a plane graph G, $G^{\rightarrow}$ is a Pfaffian orientation if the number of counterclockwise-oriented edges around any face is odd.*

Proof. All terms of $\mathrm{Pf}(K_G)$ have the same sign if for all matchings M, N, $\mathrm{sgn}(M) = \mathrm{sgn}(N)$. We observe that the number of transpositions required to orient a counterclockwise oriented circuit D so that it agrees with all the matches it covers in G is equal to the number of edges which are oriented clockwise along D and covered by M. We denote the composition of these transpositions the permutation ϕ. Clearly, ϕ is odd if and only if there are an odd number of clockwise-oriented edges under M along D.

Since k is always even (observe that C contains both vertices in any given matching), every circuit permutation is odd. Thus the permutation from M to N consists of a set of circuit permutations composed with the set of permutations which get the matchings to agree with the direction of the edges they now cover. We abuse notation somewhat to say that

$$\mathrm{sgn}(M) = \mathrm{sgn}(N) = \mathrm{sgn}(\phi(\sigma_D(M)))$$

if and only if the sign of the circuits differ. Since D has even length, this is the case if and only if D is oddly oriented.

Thus we have shown that if all faces in $G^{\rightarrow}$ are oddly oriented then all simple alternating circuits in $G^{\rightarrow}$ are oddly oriented, that two matchings connected by circuit permutations of some circuits C have the same sign if and only if the sign of C is different on each, and since, by Lemma 3.8, the set of matches on G are connected by circuit permutations, we have obtained our result. $\square$

We now utilize this rule to define an algorithm for finding Pfaffian orientations of arbitrary graphs.

Algorithm 3.19 (The Fisher-Kasteleyn-Temperly (FKT) Algorithm). Assume that G is a finite, connected, undirected plane graph. Take T to be a *spanning tree* of G; i.e., a tree such that T contains every vertex in G. Orient the edges of $G \bigcap T$ arbitrarily. Construct a graph T' whose vertices are those of G's dual graph, G^*, with an edge between two vertices if and only if the corresponding faces in G share an edge in $G \setminus T$. T' is again a tree. Orient the edges incident on the leaves of T' corresponding to interior faces of G according to Kasteleyn's Rule. Since T' is a tree, each face of G has only one edge of T' crossing it, and therefore there

is in this step only one edge on each face to orient, and this is always possible. Remove these edges from T' and repeat. When T' has no more edges, G will be oriented, and its orientation will be Pfaffian.

Theorem 3.20 (Kasteleyn's Theorem). *Every planar graph admits a Pfaffian orientation.*

Proof. To prove Kasteleyn's Theorem, we can simply prove that the FKT Algorithm works for any planar graph G. It is well-known that every graph has a spanning tree T. T' is defined as the spanning tree of G^* for which an edge $e \in G^*$ is in T' if and only if it does not cross T. We first show that construction of such a graph according to the latter rule is possible, then show that it is indeed a tree. Assume that some vertex of G^* is not connected to the rest of G. Then that vertex corresponds to a face with no edges that are not in T, making the boundary of that face a cycle, which is a contradiction. Now that we have established that G^* is connected by edges not crossing T, we now must show that T' is a tree as well. By definition, T' is a tree if and only if it contains no cycles. Assume T' has a cycle. Then T' has a face, and that face corresponds to a vertex in T. Since, by definition, the edges of T' do not cross any edges of T, the enclosed vertex cannot be part of T, which contradicts the fact that it is a spanning tree. Therefore T' is a tree.

Finally, we need to show that the leaves of T' always correspond to faces with a single unoriented edge. Assume that some leaf t of T' corresponds to a face with more than one unoriented edge. Then by defintion of T', there exists another edge of T' connecting to the face, which contradicts that t is a leaf. $\square$

We complete our treatment of Kasteleyn's method for counting perfect matchings by using these tools to count tilings of regions of the hexagon lattice by conjoined pairs of hexagons, called *bibones*.

Example 3.21. How many ways can we tile a 4-by-4 hexagon lattice with bibones?

This is equivalent to the number of perfect matchings on a 5-by-5 triangle lattice:
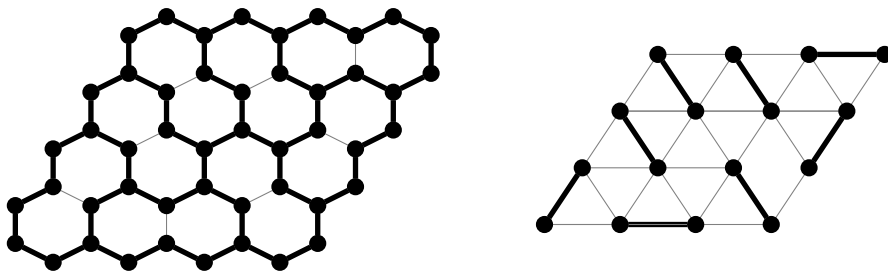

FIGURE 3. A bibone tiling and its corresponding perfect matching.

We carry out the FKT algorithm:
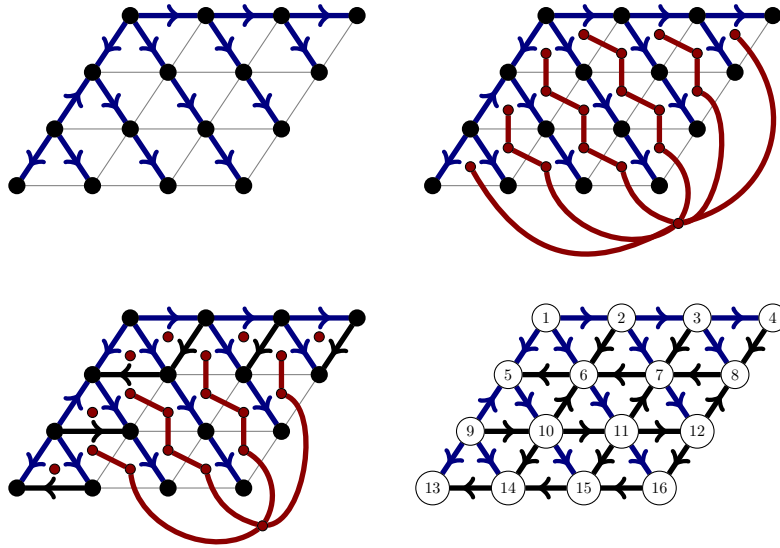
FIGURE 4. Top-left to bottom-right, a possible execution of the FKT algorithm.

The Kasteleyn matrix of the resulting digraph, numbered row-wise, is

$$
K = \begin{bmatrix}
0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & -1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 & 0
\end{bmatrix}
$$

Plugging this into our favorite computation software, we obtain $\det(K) = 3136$, so our answer is

$$
\mathrm{Pf}(K) = \sqrt{3136} = 56
$$

4. THE PERMANENT-DETERMINANT METHOD

Definition 4.1. A graph G is *bipartite* if the set of its vertices V is the union of two disjoint sets U_1 and U_2 such that every edge of G is incident on a vertex from each. We will refer this decomposition of the set of vertices as coloring them black and white, so that a bipartite graph is one for which every edge has both a black end and a white end.

Definition 4.2. For a bipartite digraph $G^\rightarrow$, colored black and white, we define the *biadjacency matrix of* $G^\rightarrow$ as the matrix M with rows corresponding to black vertices b_i and columns corresponding to white vertices w_i, so that

$$m_{i,j} = \begin{cases} \hat{w}(b_i, w_j) & \text{if } (b_i, w_j) \in E \\ -\hat{w}(w_j, b_i) & \text{if } (w_j, b_i) \in E \end{cases}$$

Definition 4.3. For a graph G, we define a *Kasteleyn-Percus matrix of* G as the biadjacency matrix P_G of a Pfaffian orientation $G^\rightarrow$ of G.

Theorem 4.4. *For a bipartite graph G, its Kasteleyn matrix K_G, and its Kasteleyn-Percus matrix P_G,*

$$\det(P_G)^2 = \pm \det(K_G)$$

Proof. Observe that for bipartite G, the indices of K_G can be chosen such that

$$K_G = \left[\begin{array}{c|c} 0 & P \\ \hline -P^T & 0 \end{array} \right]$$

We use the Leibniz formula for the determinant of K and factor it into parts corresponding to the two nonzero blocks, doing some bookkeeping for the sign of the determinant.

$$\det(K_G) = \sum_{sigma \in S_{|G|}} \text{sgn}(\sigma) \prod_{i=1}^{|G|} k_{i,\sigma(i)}$$

$$= \left(\sum_{sigma \in S_{|G|/2}} \text{sgn}(\sigma) \prod_{i=1}^{|G|/2} k_{i,\sigma(i)} \right) \left(\sum_{sigma \in S_{|G|/2}} \text{sgn}(\sigma) \prod_{j=|G|/2+1}^{|G|} k_{j,\sigma(j)} \right)$$

$$= \left(\sum_{sigma \in S_P} \text{sgn}(\sigma) \prod_{i=1}^{P} -p_{\sigma(i),i} \right) \left(\sum_{sigma \in S_P} \text{sgn}(\sigma) \prod_{j=1}^{|P|} p_{j,\sigma(j)} \right)$$

$$= (-1)^{|P|} \det(P) \det(P)$$

$$= \pm \det(P)^2$$

$\square$

We note that the determinant has an analogue to the Pfaffian's Hafnian, the *permanent*, which is defined

$$\det(K_G) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} k_{i,\sigma(i)}$$

Clearly, the Kasteleyn-Percus matrix has been crafted to have determinant equal to its permanent, for which the terms count matchings on G.

We complete our treatment of the permanent-determinant method with an example.

Example 4.5. How many ways can we tile a 5-by-4 triangle lattice region with lozenges?

As before, we start by taking the dual graph, which is a hexagon lattice region G^*. We color it as such, demonstrating that it is bipartite.
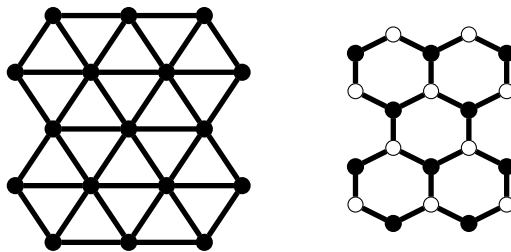
FIGURE 5. Bipartite coloring of the triangle lattice's dual graph.

We observe that since each face has six edges, we can simply orient every edge from black to white. This makes every entry in the Kasteleyn-Percus matrix positive, as shown below.

$$P = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

We find that $\det(P) = 9$, and therefore there are 9 lozenge tilings of the 5-by-4 triangle lattice region. We enumerate these in Figure 6, and the observation that they correspond to zig-zagging paths which cross only horizontal lines leads us to our next topic.
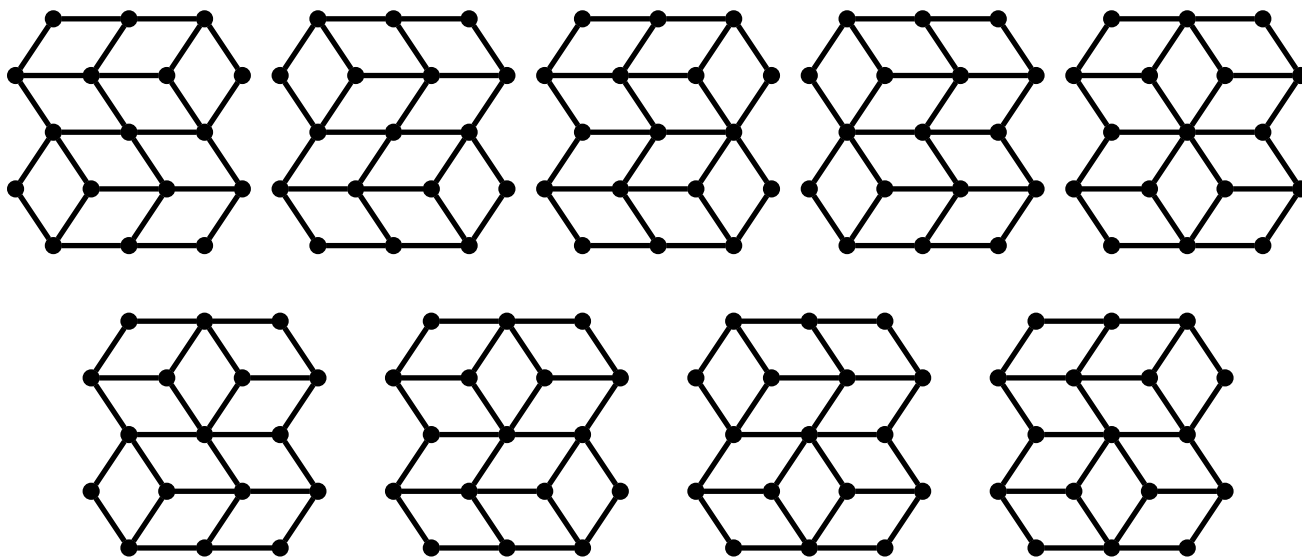
FIGURE 6. The nine possible lozenge tilings of the 5-by-4 triangle lattice.

5. The Gessel-Viennot Method

Another important tool for counting tilings and other enumerable features of graphs is the method of Gessel-Viennot, when appropriately applied. Much of its importance derives from its generality: unlike the Hafnian-Pfaffian and permanent-determinant methods, the Gessel-Viennot method can be used with nonplanar graphs. Rather than considering the perfect matchings of a graph, the Gessel-Viennot method considers sets of paths across it. We will frequently abbreviate Gessel-Viennot as 'GV'.

Definition 5.1. A set of paths is *vertex-disjoint* if they have no common vertices.

Definition 5.2. For a finite acyclic digraph G, consider two disjoint sets of vertices $A \in G$ and $B \in G$, both of cardinality l. We define a *path system* U from A to B as a set of paths such that each vertex in A is at the start of exactly one path and each vertex in B is at the terminus of exactly one path.

**Definition 5.3.** Given an ordering of A and B and an according index, we define the sign of a path system, $\mathrm{sgn}(U)$, as the sign of the permutation which sends the indices of the vertices in A to the indices of the vertices in B which end the corresponding paths in U.

Definition 5.4. We define a finite, directed, acyclic graph G together with two disjoint sets of vertices $A \in G$ and $B \in G$, such that $|A| = |B| = l$, a *l-node Gessel-Viennot system* $\mathcal{G}$.

Definition 5.5. Given an l-node Gessel-Viennot system $\mathcal{G}$, we define the *Gessel-Viennot matrix* (alternately, the *path matrix*) as the *l*-by-*l* matrix V such that $v_{i,j}$ is the sum of the weights of all possible paths in G beginning at $a_i \in A$ and terminating at $b_j \in B$.

Definition 5.6. Recalling that the weight of a path is the product of the weights of the edges in G that it contains, we define the weight of a path system as the product of the path weights:

$$\mathrm{wgt}(U) = \prod_{P \in U} \mathrm{wgt}(P)$$

Theorem 5.7 (Lindström, Gessel-Viennot). *For a l-node Gessel-Viennot system $\mathcal{G}$, take V to be the Gessel-Viennot matrix. Then*

$$\det(V) = \sum_{\substack{vertex\text{-}disjoint \\ path\ systems\ U}} \mathrm{sgn}(U)\,\mathrm{wgt}(U)$$

Proof. We begin by applying the Leibniz determinant formula to V,

$$\det(V) = \sum_{\pi \in S_n} \mathrm{sgn}(\pi) \prod_{i=1}^{n} v_{i,\pi(i)}$$

By definition of the GV matrix, we have

$$v_{i,\pi(i)} = \sum_{P_i : a_i \to b_{\pi(i)}} \mathrm{wgt}(P)$$

This gives

$$\det(V) = \sum_{\pi \in S_n} \mathrm{sgn}(\pi) \prod_{i=1}^{n} \left(\sum_{P_i : a_i \to b_{\pi(i)}} \mathrm{wgt}(P) \right)$$

Multiplying out, the definition of the weight of a path system immediately gives

$$\det(V) = \sum_{\pi \in S_n} \mathrm{sgn}(\pi) \sum_{\substack{path\ systems\ U \\ along\ \pi}} \mathrm{wgt}(U)$$

We then sum over all permutations and use the definition of the sign of a path system to obtain

$$\det(V) = \sum_{\text{path systems } U} \text{sgn}(U)\,\text{wgt}(U)$$

Our task now is to show that the terms corresponding to path systems which are not vertex-disjoint cancel. Consider any such path system U. Pick an intersection of two paths, some vertex x. This vertex connects a_i to $b_{\pi(i)}$ and a_j to $b_{\pi(j)}$. Swap the portions of paths leaving x, forming a new path system U', so that now U' contains a path from a_i to $b_{\pi(j)}$ and from a_j to $b_{\pi(i)}$.

Since U and U' are made up of the same edges, $\text{wgt}(U) = \text{wgt}(U')$. However, the permutation of U' is obtained via a transposition of that for U, so $sgn(U) = -\text{sgn}(U')$. It can be seen that each vertex of intersection admits a pair of identical path systems with inverse sign. Thusly all of these terms cancel, leaving us with our result. $\qquad\square$
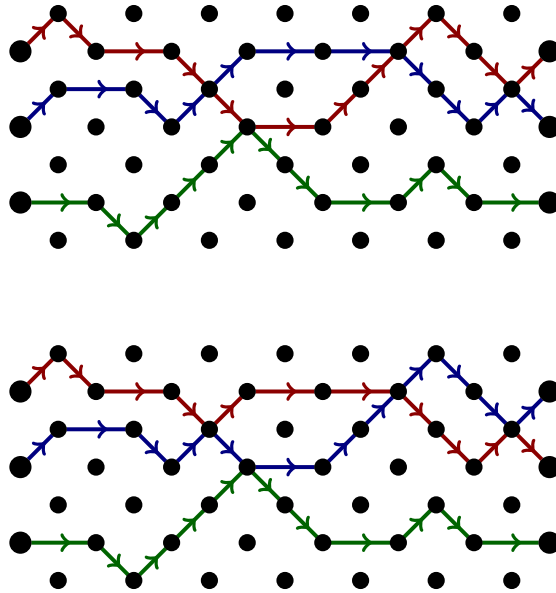
FIGURE 7. Arbitrary U and U', where we swap at the first intersection of the red and blue paths.

Corollary 5.8. *If there is only one nonvanishing permutation, then $\pm \det(V)$ is the sum of the weights of all vertex-disjoint path systems and we call the GV system* nonpermutable.

By setting all edge weights equal to one, the determinant of the GV matrix counts the number of vertex-disjoint path systems from A to B if the GV system is nonpermutable.

Definition 5.9. Given a planar graph G and a set of prototiles T, we define the *tile-path system* $\mathcal{G}^T$ as the GV system for which vertex-disjoint path systems are equivalent to tilings of G by T.

Example 5.10. Leaving lozenge tilings of triangle lattice regions as an excercise for the reader, we approach the somewhat less intuitive application of the Gessel-Viennot method to domino tilings of rectangles. We observe that with the following three prototile markings, we can for each tiling of a $2m$-by-n rectangular grid draw a unique set of m paths from left to right:

We seek to answer the question: how many ways can one tile a chessboard with dominoes? We start by finding the GV tile-path system of a chessboard using the marked dominoes shown above. All edges implicitly run left-to-right.
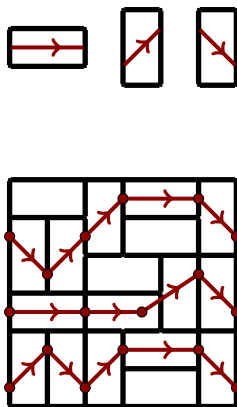
FIGURE 8. Correspondence of tilings to disjoint path systems.
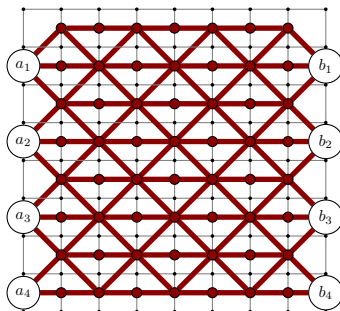
FIGURE 9. Gessel-Viennot System for domino tilings of a chessboard.

This is the familiar-looking 7-by-7 triangle lattice region. Now we set out to count the number of paths for each pair (a_i, b_j). A simple algorithm for doing so on general graphs uses recursion and a store of known numbers of paths between nodes and the end node. However, for this sufficiently small example your author opted simply to exponentiate the adjacency matrix of $\mathcal{G}^T$ 8 times (as $A_{\mathcal{G}^T}^9 = 0$), sum, and take the relevant minor (see Appendix A). Unlike the other algorithm, which is nearly linear in the number of paths, this one is in $O(\sqrt{2}n^{3.5})$, where n is the number of vertices. Our GV matrix is

$$V = \begin{bmatrix} 236 & 216 & 84 & 14 \\ 216 & 320 & 230 & 70 \\ 84 & 230 & 306 & 146 \\ 14 & 70 & 146 & 90 \end{bmatrix}$$

and

$$\det(V) = 12,988,816$$

According to the Gessel-Viennot method, there are therefore 12,988,816 tilings of the chessboard by dominoes. We verify this using the permanent-determinant method. Clearly, the truncated dual of a chess board is bipartite: the faces, which are the vertices of the dual, are explicitly colored. We find a Pfaffian orientation with the FKT algorithm, the result of which is shown in Figure 10.

FIGURE 10. Pfaffian orientation of the chessboard dual.

Numbering row-wise with a black vertex in the top-left face, we obtain the following Kasteleyn-Percus matrix:

$$P = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{bmatrix}$$

For which our favorite computation software verifies for us that indeed

$$\det(P) = 12,988,816$$

This mysterious correspondence between the Gessel-Viennot system and the Pfaffian-oriented dual graph leads us to our next topic.

6. EQUIVALENCE

We now begin our exploration of Kuperberg's results regarding the equivalence of Kasteleyn and Gessel-Viennot methods for counting tilings. First, a few prerequisite definitions.

Definition 6.1. We define a *source* to be a vertex with in-degree 0, and a *sink* to be a vertex with out-degree 0.

Definition 6.2. We define a *transit vertex* of G as a vertex in G which is neither a source nor a sink.

Definition 6.3. We say that a vertex v in a plane graph G is *segregated* if the edges incident on v can be grouped into a group on either side of v, with one group containing edges directed into v and the other group containing those directed out.

Definition 6.4. We define a *transit split* of a segregated transit vertex v as the operator which divides the edges incident on v among two new vertices u_0 and u_1 so that u_0 is a sink and u_1 is a source. An edge (u_0, u_1) is oriented accordingly. The *transit-free resolution* of a graph G is the result of transit splits of all
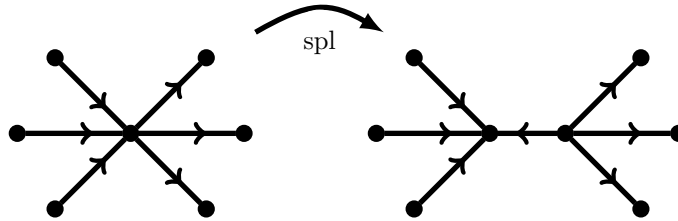
FIGURE 11. The transit split of a vertex of indegree and outdegree each equal to 3.

of its transit vertices.

Remark 6.5. Since the minors A' of adjacency matrices A (and therefore Kasteleyn and Kasteleyn-Percus matrices) correspond to subgraphs G' of graphs G, and since, via Laplace's formula, the determinant is a function of the minors of G, operations which are determinant-preserving on G' and the identity on $G \setminus G'$ are determinant-preserving on G.

Lemma 6.6. *Transit splits are determinant-preserving.*

Proof. We employ Theorem 5.7. Suppose v is a segregated transit vertex in a finite, connected, acyclic graph G (i.e., one for which the Lindström-Gessel-Viennot theorem applies). Consider the subgraph G' of G containing v and the vertices to which v is adjacent. Suppose u_0 and u_1 are, respectively, the sink and source of the transit split of v. Defining $\mathrm{spl}_v(G')$ as the graph G' after the transient split of v, we observe that there is a bijection between the paths across G' to the perfect matchings in $\mathrm{spl}_v(G')$. Simply, the edges in any given path through G' constitute the perfect matching in $\mathrm{spl}_v(G')$. $\square$

We denote as $M_{i,[:]}$ the ith row of a matrix M, and by $M_{[:],j}$ the jth column.

Definition 6.7. Given a matrix M and a matrix coordinate (i, j), we define the following three objects:

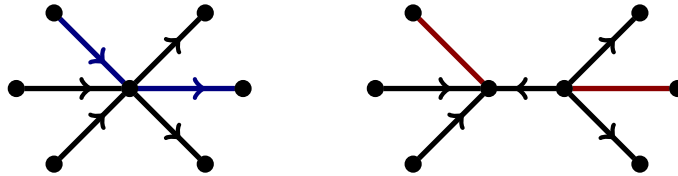- X, the ith column-vector of M with the jth element removed.

FIGURE 12. A path in some G' and corresponding matching of the split vertex.

- Y^T, the jth row-vector of M with the ith element removed.
- M', the minor of M obtained by removing the ith row and jth column.

For $(i, j) = (1, 1)$, we have

$$M = \left[\begin{array}{c|c} m_{1,1} & Y^t \\ \hline X & M' \end{array} \right]$$

We define a *deleted pivot* on M at (i, j) as the operator $\nu_{i,j}$ defined by

$$\nu_{i,j}(M) = M' - X \otimes Y^T$$

Lemma 6.8. *Deleted pivots preserve the determinant.*

Proof. We omit the proof. $\square$

Definition 6.9. A planar Gessel-Viennot system is *segregated* if the start vertices and end vertices are all on the outside face and a hypothetical vertex v can be placed on the outside face and connected to all of the start and end vertices, all edges oriented from start vertices to v and from v to end vertices, such that v is a segregated transit vertex.

Lemma 6.10. *The transit-free resolution of the graph in a segregated Gessel-Viennot system is Pfaffian.*

Proof. Following [6]:

Assume $\mathcal{G}$ is a segregated, non-Pfaffian, planar Gessel-Viennot system, arranged so that all sources $a \in A$ are on the left and all sinks $b \in B$ are on the right. We reason that $\mathcal{G}$ has a particular structure, namely that all transit vertexes are segregated, and the edges of each internal face are arranged such that all clockwise edges are pairwise coincident and counterclockwise edges are pairwise coincident. We do this by examining the contributions of each element of the graph to the Euler characteristic, which for any connected planar graph is equal that of a sphere, 2. Consider each vertex and each face of the graph contributing 1 to the Euler characteristic (we say they have characteristic 1). Now consider each pair of coincident edges in the graph which share a face. If those edges both point into or out of a vertex, subtract $\frac{1}{2}$ from the characteristic of the corresponding face. Otherwise, subtract $\frac{1}{2}$ from the characteristic of the vertex they meet at. At the end of this process, sources and sinks will have characteristic 1, segregated transit vertices will have characteristic 0, and all other vertices will have negative characteristic. Since $\mathcal{G}$ is acyclic, each interior face must have two vertices at which clock orientation changes, and so all interior faces will have characteristic 0. Since the outer face must experience clock orientation changes both at and between each source and sink, it has characteristic equal to $1 - (1/2)(2|A| - 1) - (1/2)(2|B| - 1) = 2 - 2|A|$. Thus, if r is the total characteristic of vertices that are neither sinks, sources, nor segregated transit vertices, we have total characteristic equal to

$$|A| + |B| + 2 - 2|A| + r = 2|A| + 2 - 2|A| + r = 2 + r$$

Which is equal to 2 if and only if r equals zero. Thus $\mathcal{G}$ has the desired structure.

Take $\mathcal{G}'$ to be the transit-free resolution of $\mathcal{G}$. Since the transit vertices are segregated, $\mathcal{G}'$ is planar. Since for each face f, all clockwise edges are pairwise coincident and all counterclockwise edges are pairwise coincident, if f has k edges on its boundary then the corresponding face f' of $\mathcal{G}'$ has $2k-2$, since every vertex about f is split, but the splittings of the two for which clock orientation about f changes do not contribute edges to f'. Since we assumed $\mathcal{G}$ is non-Pfaffian, we reason that $k \equiv 0 \pmod 4$. Therefore $2k - 2 \equiv 2 \pmod 4$, so f' has twice an odd number of edges about it, and since transient splitting contributed clockwise and counterclockwise edges to f' in equal parts, f' has an odd number of counterclockwise edges about it. Therefore Kasteleyn's rule is satisfied and $\mathcal{G}'$ is Pfaffian. $\square$

With these two results in hand, we present the algorithm for translating between Gessel-Viennot and Kasteleyn-Percus matrices.

Algorithm 6.11 (Kuperberg). Let $\mathcal{G}$ be a Gessel-Viennot system on a graph G such that all the endpoints A and B are on the outside face of G. Take $\mathcal{G}'$ to be the transit-free resolution of $\mathcal{G}$. If all transit vertices in $\mathcal{G}$ are segregated, then the Gessel-Viennot matrix V of $\mathcal{G}$ is obtained from a Kasteleyn-Percus matrix P of $\mathcal{G}'$ by taking the (i, j) deleted pivot of P wherever the corresponding i, j vertex pair in $\mathcal{G}'$ is contracted in $\mathcal{G}$. If $\mathcal{G}$ is not segregated, then V is obtained by a combination of deleted pivots and other matrix operations.

Example 6.12. We return to example 4.5. We begin by transit splitting every transit vertex in its Gessel-Viennot system $\mathcal{G}$.
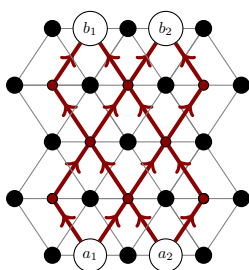
FIGURE 13. Gessel-Viennot system for 5-by-4 triangle lattice.
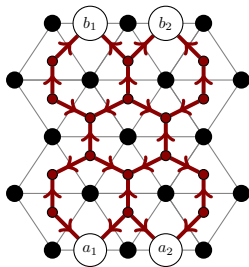

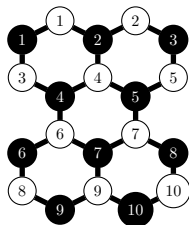FIGURE 14. The transit-free resolution.

FIGURE 15. Bipartite graph numbering for the transit-free resolution.

The transit-free resolution of $\mathcal{G}$ has left us with G' in a Pfaffian orientation. Starting from the Kesteleyn-Percus matrix P of G', we carry out the eight deleted pivots corresponding to the eight transit splits.

$$P = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix}$$

We verify that the determinant of this matrix is indeed equal to 9. The pairs (i, j) which are contracted in $\mathcal{G}$ are $(1, 3), (2, 4), (3, 5), (4, 6), (5, 7), (6, 8), (7, 9)$ and $(8, 10)$. We carry out the first deleted pivot, $\nu_{1,3}$.

$$\nu_{1,3}(P) = P' - X \otimes Y^T$$

$$= \begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix}$$

Using the Octave script provided in Appendix B, we compute the remaining seven deleted pivots, leaving us with

$$\begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

We can quickly verify by examination of Figure 13 that these entries count the corresponding paths in the Gessel-Viennot system, and we compute that its determinant is again 9.

Appendix A. Gessel-Viennot Matrix Algorithms

There are two algorithms of particular interest for calculating the Gessel-Viennot matrix. The first is asymptotically much faster. The latter is more mathematically instructive. We will examine both here.

We provide the first in pseudocode form.

Algorithm A.1. From [10]

```
1     let 𝒢 be a Gessel−Viennot system with start vertices a_i ∈ A and end vertices b_i ∈ B
2     let l equal |A| (which equals |B|)
3     initialize V to an empty l × l matrix
4     let num_paths be a map from vertices to integers
5     for each (a_i, b_j) ∈ A × B:
6         initialize all num_paths to 0
7         num_paths[b_j] = 1
8         let stk be an empty stack
9         push ('enter', a_i) onto stk
10        while stk is not empty:
11            pop (op, v) from stk
12            if (op == 'enter'):
13                push ('leave', v) onto stk
14                for each edge (v, w):
15                    push ('enter', w) onto stk
16            else (op == 'leave'):
17                for each edge (v, w):
18                    num_paths[v] = num_paths[v] + num_paths[w]
19        V[i][j] = num_paths[a_i]
```

The general process expressed here is to iterate over the vertices in $\mathcal{G}$ in reverse topological order, keeping track of number of paths from each vertex to the end vertex, and summing them along the way. To calculate the asymptotic time, we observe that the structure of the algorithm is a while loop that executes once for each edge and once for each vertex, inside of a loop that runs over the pairs $(a_i, b_j) \in A \times B$. Thus the asymptotic complexity is $\mathcal{O}(l^2|V||E|)$.

The other algorithm uses the fact that the $a_{i,j}$ element of the nth power of an adjacency matrix A gives the number of n-paths between vertex i and vertex j. Clearly, since $\mathcal{G}$ is acyclic, the maximum path length is finite, so A is nilpotent. It makes sense, then, to take the (partial) series

$$W = \sum_{i=1}^{k} A^i$$

since this sum converges after a finite number of terms (because the terms go to zero). Each entry of W, then, is the total number of paths of any length between two vertices. This is precisely what we seek for pairs of vertices in $A \times B$, so the Gessel-Viennot matrix is simply the minor of W corresponding to these pairs. The implementation in Octave is extremely simple, and is given in Algorithm A.2.

Intuitively, we expect this to be less efficient since it calculates the number of paths not just between A and B, but between every pair of vertices in the graph. Indeed, even in the asymptotic case where we get to disregard huge constants, the fastest matrix multiplication is the Coppersmith-Winograd algorithm, which multiplies matrices in $\mathcal{O}(|V|^{2.3727})$. Let m be the minimal dimension of $\mathcal{G}$ and l the number of edges per face along this dimension. If we orient G so that paths go across this minimal dimension (alternately, we could view this as the maximization of A and B, getting the maximal number of useful entries from W), it

Algorithm A.2. Let A be the adjacency matrix of $\mathcal{G}$.

```
1  function [ P ] = path_matrix( A, l )
2     sz = length(A)
3     numsum = floor(sqrt(2*sz) + 4)
4     Apows = zeros(sz, sz, numsum)
5     for i=1:numsum
6           Apows([1:sz], [1:sz], i) = A^i
7     end
8     bigsum = sum(Apows, 3)
9     P = bigsum([1:l],[sz-l+1:sz], 1)
10 end
```

is evident that the longest path in $\mathcal{G}$, and thus the number of times we need to exponentiate A, will be ml. Therefore the asymptotic runtime will be in $\mathcal{O}(|V|^{2.3727ml})$, which gets unpleasantly large rather quickly, but is well within reason for the 64-vertex chessboard graph.

Appendix B. Matlab Code for Deleted Pivot

The following Matlab function returns the result of a deleted pivot operation at i, j on an arbitrary matrix M.

Algorithm B.1. Let M be any matrix.

```
1  function res = deleted_pivot(M, i, j)
2     % Calculates the deleted pivot of matrix M at row i, column j
3
4     % X = the jth column of M with element i removed
5     X = [M(:,j)(1:i-1); M(:,j)(i+1:end)];
6     % Yt = the ith row of M with element j removed
7     Yt = [M(i,:)(1:j-1), M(i,:)(j+1:end)];
8     % Mxy is what we're calling M', which is the matrix M with row i
9     % and column j removed.
10    Mx = [M(:, 1:j-1), M(:, j+1:end)];
11    Mxy = [Mx(1:i-1, :); Mx(i+1:end, :)];
12    % The result is given as M' - X*Yt, where * is the outer product
13    res = Mxy - X*Yt;
14 endfunction
```

When doing successive deleted pivots, we must after each round update the indices targeted by subsequent rounds, since the indices of the matrix may have changed. Code to do multiple deleted pivots in succession, taking care of this bookkeeping, can be found at `https://gist.github.com/brandones/cb866936bb4ca3311b0516a23d6a92b5`

Endnotes

This paper was handed in incomplete in March 2013. In particular, the final example, demonstrating the conversion from a Gessel-Viennot Path System to a graph in Pfaffian orientation and from a Kastelyn-Percus Matrix to a GV-Matrix was only half-complete. A few other minor updates have also been made:

In line 3 of Equation 3.2, $M \times N$ has been changed to $M + N$.

In the matrix under Definition 6.7, the 1 in the upper-left quadrant has been changed to $m_{1,1}$.

Above Algorithm A.1 used to read "We provide the first in pseudocode form, where it is most naturally expressed." At the time I clearly didn't know anything about making code "naturally expressive".

References

[1] Aigner, Martin. *Lattice Paths and Determinants*. Computational Discrete Mathematics - Advanced Lectures. Springer (2001)

[2] Gessel and Viennot. *Binomial Determinants, Paths, and Hook Length Formulae*. Advances in Mathematics 58, 300-321 (1985).

[3] Kasteleyn, P.W. *Dimer Statistics and Phase Transitions*. Journal of Mathematical Physics, Vol. 4, Num. 2 (February 1963).

[4] *Introduction to the dimer model*. Lecture notes. 2002.

[5] Kuperberg, Greg. *An exploration of the permanent-determinant method*. The Electronic Journal of Combinatorics 5 (1998), #R46

[6] Kuperberg, Greg. *Kasteleyn Cokernels*. The Electronic Journal of Combinatorics 9 (2002), #R29

[7] Nguyen, Jeanette. *Perfect Matchings and Pfaffian Orientation*. Bachelor's Thesis, Universiteit van Amsterdam. 2008

[8] Percus, Jerome. *One More Technique for the Dimer Problem*. Journal of Mathematical Physics Vol. 10, Num. 10 (October 1969).

[9] Reid, Homer. *Determinants That Count*. Lecture notes. 2013.

[10] User "rich". Answer to "Simple Paths between 2 nodes". `https://stackoverflow.com/a/11105908/1464495` Edited 20 June 2012. Accessed 2 March 2013.

[11] de Tiliére, Béatrice. *The dimer model in Statistical mechanics*. Lecture notes. 2009.