Enumeration of Messy Polygon Mosaics

Jack Hanke **Richard Schank**
Northwestern University

Abstract

Hong and Oh introduced a model for multiple ring polymers in physics in which an $m \times n$ matrix is constructed from a selection of 7 distinct tiles. These matrices are called *mosaics*. The authors provide bounds on a subset of these mosaics that have the property of being suitably connected, which they call polygon mosaics. We introduce and enumerate mosaics with the related property in which only some tiles must be suitably connected, which we call messy polygon mosaics.

1 Introduction

Hong and Oh [2] introduced a model for multiple ring polymers in which an $m \times n$ matrix is constructed using 7 distinct symbols callet *tiles*. These tiles, diagrammed in Figure 1, are composed of unit squares with dotted lines connecting 2 sides at their midpoint, as well as the "blank" tile T_0.

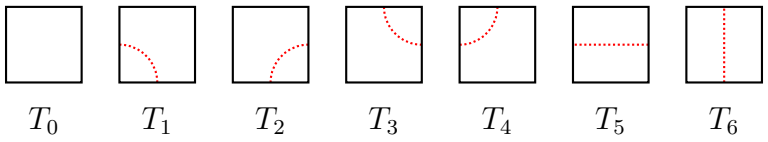
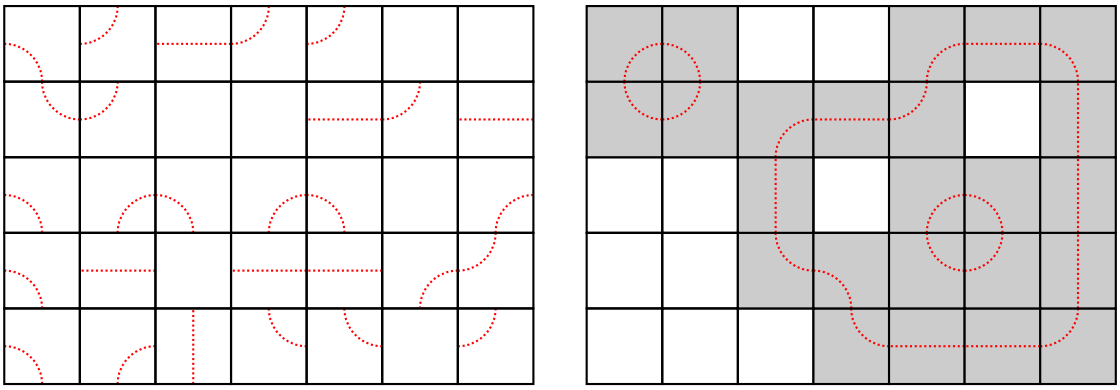


Figure 1: The tile set $\mathbb{T}$

We denote the set of tiles $\mathbb{T} = \{T_0, \ldots, T_6\}$. An (m, n) *mosaic* is an $m \times n$ matrix made up of elements from $\mathbb{T}$. Figure 2a shows an example $(5, 7)$ mosaic. We denote the set of all (m, n) mosaics as $\mathbb{M}^{(m,n)}$. As there are 7 elements in $\mathbb{T}$, there are 7^{mn} mosaics in $\mathbb{M}^{(m,n)}$.

The authors in [2] are interested in mosaics with the property of being *suitably connected*, which is defined as follows. Consider an edge shared between two tiles in Figure 2a. The edge has either 0, 1, or 2 dotted lines drawn from its midpoint. Also note that the outer edges of the tiles on the boundary of the matrix are not shared by another tile. Therefore these edges only have 0 or 1 dotted lines drawn from their midpoint. A mosaic is suitably connected if all edges have 0 or 2 dotted lines drawn from their midpoint. The authors call these *polygon mosaics* because, other than the mosaic consisting of all T_0 tiles, the dotted lines form *polygons*.



(a) A mosaic

(b) A polygon mosaic

Figure 2: Examples of mosaics of size $(5, 7)$ made of tiles in $\mathbb{T}$

Definition 1.1. A *polygon* $\mathcal{P}$ in a mosaic $\mathcal{M}$ is a non-empty subset of tiles in $\mathcal{M}$ such that if all cells in $\mathcal{P}^c$ are removed from $\mathcal{M}$, all remaining edges shared by 2 tiles have 2 dotted lines drawn from the midpoint, and all other edges have 0 dotted lines drawn from their midpoint.[1]

Example 1.1. Figure 2b shows a polygon $(5, 7)$ mosaic that contains 3 polygons, with the background of the tiles that make up the polygons highlighted in gray. Note that a mosaic can contain polygons that surround other polygons, such as in Figure 2b.

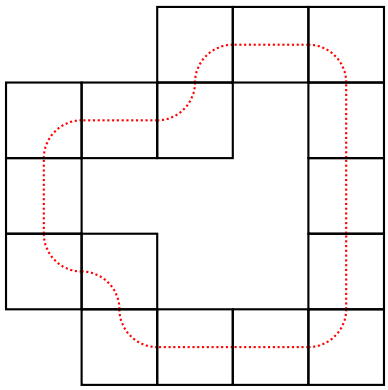


Figure 3: The largest polygon in Figure 2b

Using the notation conventions in [9], we denote the subset of (m, n) mosaics that are polygon mosaics[2] as $\mathbb{P}^{(m,n)}$.

[1] Polygons are more commonly refered to as "self-avoiding polygons" in the literature to highlight their connection with self-avoiding walks.

[2] The array of values of $\mathbb{P}^{(m,n)}$ is sequence A181245 on the OEIS [4, OEIS].

Theorem 1.1 ([2]). *The number of polygon (m,n) mosaics $|\mathbb{P}^{(m,n)}|$ is bounded by*

$$2^{m+n-3}\left(\frac{17}{10}\right)^{(m-2)(n-2)} \leq |\mathbb{P}^{(m,n)}| \leq 2^{m+n-3}\left(\frac{31}{16}\right)^{(m-2)(n-2)}.$$

We refer to the (m,n) mosaic composed of all T_0 tiles as $\mathcal{M}^*$. This work differs in the definition of a polygon mosaic from [2], as we *do* consider $\mathcal{M}^*$ a polygon mosaic. [2] points out that Theorem 1.1 holds irrespective of this choice.

In related work, Lomonaco and Kauffman [3] introduced mosaics constructed from a tile set of 11 distinct tiles, of which $\mathbb{T}$ is a subset. The authors call mosaics constructed from this tile set that are suitably connected *knot mosaics*. Oh et al. [9] enumerated the number of knot mosaics.

Theorem 1.2 ([9]). *The number of knot mosaics of size (m,n) for $m,n \geq 2$ is $2\,\|(X_{m-2}+O_{m-2})^{n-2}\|$, where $X_0 = O_0 = \begin{bmatrix}1\end{bmatrix}$ and X_{m-2} and O_{m-2} are $2^{m-2}\times 2^{m-2}$ matrices defined as*

$$X_{k+1} = \begin{bmatrix} X_k & O_k \\ O_k & X_k \end{bmatrix} \text{ and } O_{k+1} = \begin{bmatrix} O_k & X_k \\ X_k & 4O_k \end{bmatrix},$$

for $k = 0, 1, \ldots, m-3$. Here $\|N\|$ denotes the sum of elements of matrix N.

Oh and colleagues refer to these matrices X_k and O_k as *state matrices*. The authors utilize this state matrix recursion to bound the growth rate of knot mosaics $\delta = \lim_{n\to\infty} k_{n,n}^{\frac{1}{n^2}}$ [6, 8, 1], and Oh further adapts the method to solve problems in monomer and dimer tilings [5, 7]. An unexamined direction in this research program is modifying the suitably connected property. This motivates us to introduce *messy polygon mosaics* using the tiles set $\mathbb{T}$ from [2], and which we enumerate with state matrices as in [9].

2 Messy Polygon Mosaics

Definition 2.1. A *messy polygon mosaic* is a mosaic that contains at least one polygon, or is $\mathcal{M}^*$.

Figure 4 shows two examples of messy polygon $(5,7)$ mosaics that contain 3 polygons. In both examples, we highlight the background of the tiles that make up each polygon in gray.

The main goal of this paper is to enumerate messy polygon mosaics, but it turns out to be simpler to enumerate the number of mosaics that *do not* contain a polygon. Therefore, let $\mathbb{S}^{(m,n)}$ be the set of mosaics that do not contain a polygon. Clearly the number of messy polygon mosaics is then $7^{mn} - |\mathbb{S}^{(m,n)}|$.

From the fact that the smallest polygon is made of 4 tiles, shown in Figure 5, we can conclude that $s_{n,1} = 7^n$, and $s_{2,2} = 7^4 - 1$. For $n, m \geq 2$, we first define the state matrices for messy polygon mosaics.

Definition 2.2. Define $A^{(1,2)} = \begin{bmatrix} 7^2 & -1 \\ 1 & 1 \end{bmatrix}$. We recursively define $A^{(1,k+1)} \in \mathbb{Z}^{2^k\times 2^k}$ given $A^{(1,k)}$. Begin by writing $A^{(1,k)} = \begin{bmatrix} A_{[0,0]} & A_{[0,1]} \\ A_{[1,0]} & A_{[1,1]} \end{bmatrix}$, where the block matrices $A_{[i,j]}$ are square block matrices of size $2^{k-1}\times 2^{k-1}$. We then define

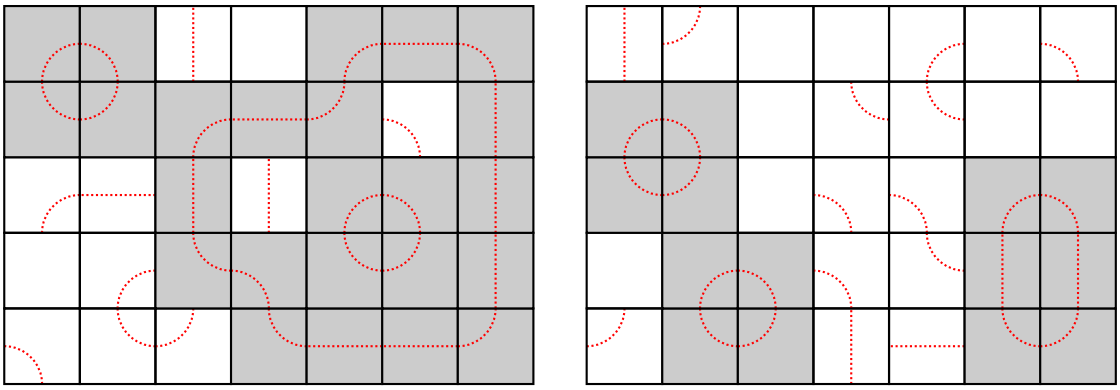


Figure 4: Messy polygon mosaics

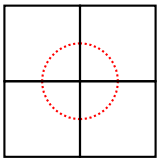


Figure 5: The smallest polygon

$$A^{(1,k+1)} = \begin{bmatrix} 7A_{[0,0]} & 7A_{[0,1]} & 7^{-1}A_{[0,0]} & A_{[0,1]} \\ 7A_{[1,0]} & 7A_{[1,1]} & 0A_{[1,0]} & A_{[1,1]} \\ -7^{-1}A_{[0,0]} & 0A_{[0,1]} & 7^{-1}A_{[0,0]} & A_{[0,1]} \\ A_{[1,0]} & A_{[1,1]} & -A_{[1,0]} & 7A_{[1,1]} \end{bmatrix}.$$

Theorem 2.1. $|\mathbb{S}^{(m,n)}|$ *is the* $(0,0)$ *entry of* $(A^{(1,m)})^n$.

3 Preliminaries

We begin by defining a mapping f between $\mathbb{M}^{(m,n)}$ to a *binary lattice* of size (m,n). A binary lattice of size (m,n) is a rectangular lattice of $m+1$ by $n+1$ vertices, in which the boundary vertices are labeled 0, and the interior vertices are either 0 or 1. An example of a binary lattice of size $(5,7)$ is shown on the right of Figure 6. Also let $\mathbb{L}^{(m,n)}$ be the set of all binary lattices of size (m,n), which gives $\left|\mathbb{L}^{(m,n)}\right| = 2^{(m-1)(n-1)}$.

Definition 3.1. $f : \mathbb{M}^{(m,n)} \to \mathbb{L}^{(m,n)}$ takes a mosaic and labels each vertex with the following rule. If the vertex is surrounded by an even number of polygons (including 0 polygons), label it 0. If the vertex is surrounded by an odd number of polygons, label it 1. Removing the red dotted lines from the tiles gives the binary lattice.

We want to compute $|\mathbb{S}^{(m,n)}|$ by computing the preimage $f^{-1}(\{\ell\})$ for each $\ell \in \mathbb{L}^{(m,n)}$, and then summing over all ℓ. We do this by examining individual *cells* of the binary lattice.

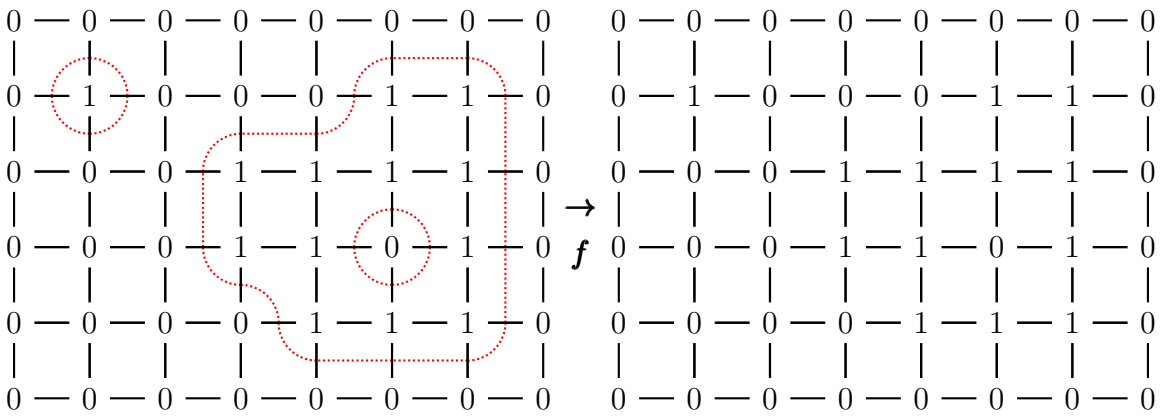
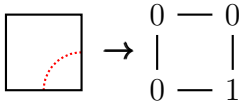


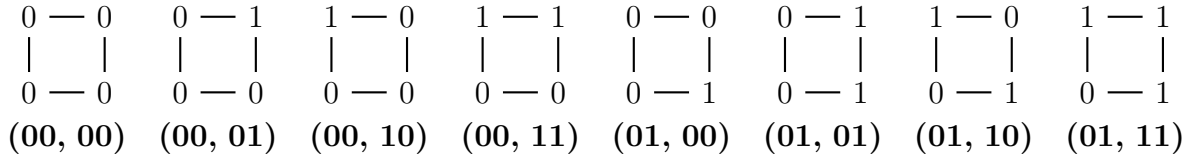
Figure 6: f applied to the left mosaic in Figure 4, resulting in a binary lattice

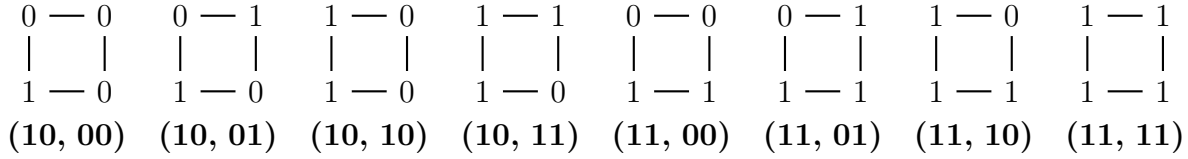
Definition 3.2. Let a *cell* be the unit square and 4 labelled vertices that a given tile maps to, and let C be the set of unique cells.

Example 3.1. Applying f to the mosaic In Figure 6 results in the top-left tile T_2 mapping to the cell diagrammed below.



For convenience, we give a pair of indexes to each of the $|C| = 2^4$ unique cells. The first index is the binary number formed by reading the bottom two vertices from left to right. The second index is the binary number formed by reading the top two vertices from left to right. The cell in Example 3.1 has index $(01, 00)$. Below is a diagram of all 2^4 cells with their indexes listed below.





Next let $u_{(i,j)}$ be the number of tiles in $\mathbb{T}$ that can map to cell (i, j). These values are simple to calculate, as each tile in $\mathbb{T}$ can only be part of a polygon in certain ways. For example, $u_{(01,00)} = 1$, as cell $(01, 00)$ can only be formed by a mosaic with T_2 in that location. We can see that $u_{(01,10)} = 0$, as cell $(01, 10)$ cannot be formed from any tiles in $\mathbb{T}$. Finally, we

Cell (i, j)	Preimage	$\mathbf{u_{(i,j)}}$	Cell (i, j)	Preimage	$\mathbf{u_{(i,j)}}$
$(00, 00)$	$\mathbb{T}$	7	$(11, 11)$	$\mathbb{T}$	7
$(00, 01)$	$\{T_3\}$	1	$(11, 10)$	$\{T_3\}$	1
$(00, 10)$	$\{T_4\}$	1	$(11, 01)$	$\{T_4\}$	1
$(00, 11)$	$\{T_5\}$	1	$(11, 00)$	$\{T_5\}$	1
$(01, 00)$	$\{T_2\}$	1	$(10, 11)$	$\{T_2\}$	1
$(01, 01)$	$\{T_6\}$	1	$(10, 10)$	$\{T_6\}$	1
$(01, 10)$	$\{\}$	0	$(10, 01)$	$\{\}$	0
$(01, 11)$	$\{T_1\}$	1	$(10, 00)$	$\{T_1\}$	1

Table 1: Preimages of each unique cell

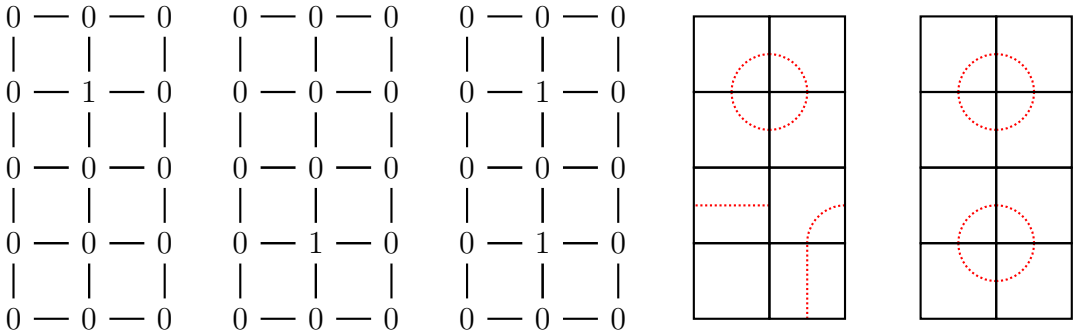
have $u_{(00,00)} = 7$, as any tile can fail to contribute to forming a polygon. Table 1 summarizes the tiles in the preimage for each cell (i, j).

However, for some binary lattice ℓ the quantity

$$U(\ell) := \prod_{\text{Cell } (i,j) \in \ell} u_{(i,j)} \tag{1}$$

is not necessarily equal to $|f^{-1}(\{\ell\})|$, as $U(\ell)$ does not *just* count the number of mosaics that map to ℓ under f. Therefore, there is not a simple way to compute $f^{-1}(\{\ell\})$ for a binary lattice ℓ just by examining the structure of ℓ.

Example 3.2. Consider the following binary lattices and mosaics for $m = 4, n = 2$.



We have $U(\ell_1) = 7^4, U(\ell_2) = 7^4$, and $U(\ell_3) = 1$. $U(\ell_1)$ uniquely counts Mosaic 1, but both $U(\ell_1)$ and $U(\ell_2)$ count Mosaic 2, for which $f(\text{Mosaic 2}) = \ell_3$. This is because each cell in the bottom two rows of $(00, 00)$ cells in ℓ_1 could have come from 7 possible cells, though 1 of the 7^4 permutations contains a new polygon.

Definition 3.3. A polygon is *specified* by a binary lattice ℓ if all mosaics in $f^{-1}(\{\ell\})$ contain the polygon.

Example 3.3. From Example 3.2, ℓ_1 specifies the polygon in the top 2 rows of Mosaic 1 and Mosaic 2, but not the polygon in the bottom 2 rows of Mosaic 2. ℓ_3 specifies both polygons in Mosaic 2.

We next point out how specified mosaics arise in binary lattices.

Definition 3.4. A subset of vertices X in a binary lattice ℓ is said to be *connected* if all vertices are the same value, and for any pair of vertices $x_1, x_2 \in X$ there exists a path between x_1 and x_2 of unit vertical, and horizontal moves so that all of vertices in the path are in X.

In a binary lattice, a specified polygon corresponds with a set of connected 0 or 1 vertices, that if 0 are not connected to the boundary 0's. This is illustrated in Figure 6, where the 3 polygons correspond with 3 connected regions, excluding the region that is connected to the boundary.

Definition 3.5. Let $P : \mathbb{L}^{(m,n)} \to \mathbb{N}$ be the number of polygons specified in ℓ.

Example 3.4. From Example 3.2, $P(\ell_1) = 1$, $P(\ell_2) = 1$, and $P(\ell_3) = 2$.

Definition 3.6. For a binary lattice ℓ, let $\mathbb{U}(\ell)$ be the set of binary lattices whose preimage mosaics under f are counted by $U(\ell)$. That is, $\mathbb{U}(\ell)$ is the set of binary lattices such that for all ℓ

$$U(\ell) = \sum_{\ell' \in \mathbb{U}(\ell)} |f^{-1}(\ell')|. \tag{2}$$

Example 3.5. From Example 3.2, $\mathbb{U}(\ell_1) = \{\ell_1, \ell_3\}$, $\mathbb{U}(\ell_2) = \{\ell_2, \ell_3\}$, and $\mathbb{U}(\ell_3) = \{\ell_3\}$.

A binary lattice ℓ' is in $\mathbb{U}(\ell)$ if one can replace either some number of $(00, 00)$ or $(11, 11)$ cells in ℓ with other cells in C to create ℓ'. This corresponds with fixing all polygons specified in ℓ and then specifying new polygons in the remaining space.

Definition 3.7. Let $\ell^* \in \mathbb{L}^{(m,n)}$ be the binary lattice made up of all $(00, 00)$ cells.

The binary lattice ℓ^* specifies 0 polygons and has $U(\ell^*) = 7^{mn} = \left|\mathbb{M}^{(m,n)}\right|$, and$\mathbb{U}(\ell^*) = \mathbb{L}^{(m,n)}$. This immediately gives $|\mathbb{M}^{(m,n)}| \neq \sum_{\ell \in \mathbb{L}^{(m,n)}} U(\ell)$, as the sum overcounts mosaics for all ℓ that have $\mathbb{U}(\ell) \neq \{\ell\}$.

Surprisingly, we can recover $|\mathbb{S}^{(m,n)}|$ by calculating how much each preimage of ℓ is over counted.

Proposition 3.1. *By regrouping terms we have*

$$\sum_{\ell \in \mathbb{L}^{(m,n)}} U(\ell) = \sum_{\ell \in \mathbb{L}^{(m,n)}} \sum_{\ell' \in \mathbb{U}(\ell)} |f^{-1}(\ell')| = \sum_{\ell \in \mathbb{L}^{(m,n)}} \left(\binom{P(\ell)}{0} + \cdots + \binom{P(\ell)}{P(\ell)} \right) |f^{-1}(\ell)|. \tag{3}$$

Proof. The first equality follows directly from Equation 2. For the second equality, notice that the number of times $|f^{-1}(\ell)|$ appears in the second sum of Equation 3 is the number of times a binary lattice ℓ appears in the set

$$\bigcup_{\ell \in \mathbb{L}^{(m,n)}} \{\mathbb{U}(\ell)\}.$$

If ℓ has $P(\ell) > 0$, the definition of $\mathbb{U}$ gives that ℓ appears once in the $\mathbb{U}$ set for the binary lattice that specifies 0 polygons (ie. ℓ^*). ℓ also appears in the $\mathbb{U}$ set for each binary lattice that specifies 1 of the polygons in ℓ. ℓ also appears in the $\mathbb{U}$ set for each binary lattice that specifies 2 of the polygons in ℓ, and so on up to specifying $P(\ell)$ polygons. As the number of ways $p \leq P(\ell)$ polygons can specify $P(\ell)$ polygons are the binomial coefficients, this gives the second equality for all $\ell \neq \ell^*$.

If $\ell = \ell^*$, we have that $P(\ell^*) = 0$, so $|f^{-1}(\ell^*)|$ is only counted once. As $\binom{0}{0} = 1$, this completes the proof. □

Proposition 3.2. *The number of mosaics of size (m, n) that do not contain a polygon $|\mathbb{S}^{(m,n)}|$ has*

$$|\mathbb{S}^{(m,n)}| = \sum_{\ell \in \mathbb{L}^{(m,n)}} (-1)^{P(\ell)} U(\ell). \tag{4}$$

Proof. By the binomial theorem,

$$0 = (1-1)^{P(\ell)} = \left(\binom{P(\ell)}{0} - \binom{P(\ell)}{1} + \cdots + (-1)^{P(\ell)} \binom{P(\ell)}{P(\ell)} \right),$$

so if we group terms as in Equation 3, we get that all terms where $\ell \neq \ell^*$ are 0. Therefore,

$$\sum_{\ell \in \mathbb{L}^{(m,n)}} (-1)^{P(\ell)} U(\ell) = \binom{0}{0} |f^{-1}(\ell^*)| = |\mathbb{S}^{(m,n)}|.$$

□

4 A Cell-Level Identity

Though Equation 4 does compute $|\mathbb{S}^{(m,n)}|$, computing the number of polygons specified in a binary lattice $P(\ell)$ requires examining the entire, global structure of ℓ. It will be more efficient to recover the $(-1)^{P(\ell)}$ term at the cell level. More concretely, we seek a function V of the form

$$V(\ell) := \prod_{\text{Cell } (i,j) \in \ell} v_{(i,j)},$$

such that for all ℓ

$$(-1)^{P(\ell)} U(\ell) = V(\ell). \tag{5}$$

The idea is to add a coefficient $p_{(i,j)} \in \{-1, 1\}$ to each value of $u_{(i,j)}$ so that the $(-1)^{P(\ell)}$ term is recovered. Therefore we define $v_{(i,j)} := p_{(i,j)} u_{(i,j)}$.

Condition 4.1. *A subset of cells $\mathcal{S}$ in a binary lattice ℓ meets this condition if*

$$\prod_{Cell\ (i,j)\in\mathcal{S}} p_{(i,j)} = -1, \tag{6}$$

with $p_{(i,j)} \in \{-1, 1\}$.

By the definition of $P(\ell)$, if there exists values $p_{(i,j)}$ for which Condition 4.1 holds only for cells $\mathcal{S}$ that specify a polygon, then the $(-1)^{P(\ell)}$ is recovered.

Proposition 4.2. *There exists values $p_{(i,j)}$ for which Condition 4.1 holds for any set of cells that specify a polygon.*

Proof. We can immediately see $p_{(00,00)} = p_{(11,11)} = 1$, as cells $(00, 00)$ and $(11, 11)$ can never be in the collection of cells that specify a polygon, and so must be positive.

For the remaining values of $p_{(i,j)}$, we first examine the cells that map to the smallest polygon, shown in Figure 5.

Figure 7: Portions of binary lattices associated with the smallest polygon

Condition 4.1 amounts to the following two equations

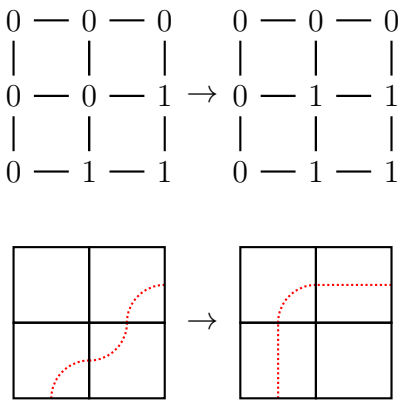
$$\begin{aligned} p_{(00,01)}p_{(00,10)}p_{(01,00)}p_{(10,00)} &= -1 \\ p_{(11,10)}p_{(11,01)}p_{(10,11)}p_{(01,11)} &= -1, \end{aligned} \tag{7}$$

one for each portion of a binary lattice in Figure 7. We refer to the equations above as *constraints*, as they constrain the possible assignments of $p_{(i,j)}$. Let the constraints in Equation 7 be numbered 1 and 2.

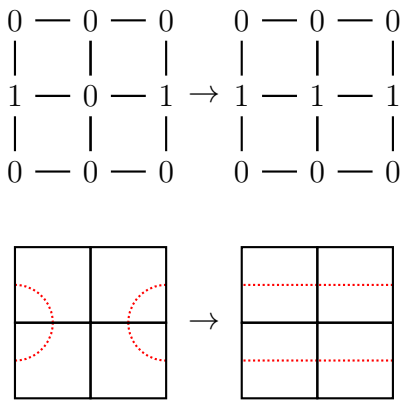
To define the remaining constraints, we start with a single vertex labeled 1. We can construct an arbitrary set of connected vertices labeled 1 by continually flipping a neighboring 0s to 1s. This *bit flip* operation, depicted in Figure 8, corresponds with changing the identity of the four cells that share that vertex.

As each cell has its associated $p_{(i,j)}$ value, it must be the case that the bit flip preserves Condition 4.1 for the cells involved. Additionally, as the surrounding 8 vertices are unchanged, this creates 2^8 constraints. In each of these constraints, the parity of the number of connected regions of 1s must either change or stay the same after the bit flip. If the parity remains unchanged, the bit flip is of *Type 1*, and if the parity changes the bit flip is of *Type 2*.

For a bit flip of *Type 1*, to adhere to Condition 4.1, the associated constraint is that the sign of the product of the related cells must stay the same after the flip. For example, Figure 8a represents the constraint



(a) Type 1 bit flip and associated tiles



(b) Type 2 bit flip and associated tiles

Figure 8: Bit flips for binary lattices

$$p_{(01,00)}p_{(11,01)}p_{(00,00)}p_{(01,00)} = p_{(01,01)}p_{(11,11)}p_{(01,00)}p_{(11,00)}. \tag{8}$$

For a bit flip of *Type 2*, to adhere to Condition 4.1, the associated constraint is that the sign of the product of the related cells must change after the flip. For example, Figure 8b represents the constraint

$$p_{(00,10)}p_{(00,01)}p_{(10,00)}p_{(01,00)} = -p_{(00,11)}p_{(00,11)}p_{(11,00)}p_{(11,00)}. \tag{9}$$

In Figure 8b the transformation corresponds with *either* two distinct polygons joining into one polygon *or* one polygon splitting into two distinct polygons. In either case, we want the sign of the product to change.

This defines a procedure to define the 2^8 bit flip constraints on $p_{(i,j)}$. We can then use software to verify that all $2^8 + 2$ constraints admit a solution.

□

We summarize the values of $p_{(i,j)}$ and $v_{(i,j)}$ in Table 2.

Cell (i, j)	$\mathbf{u}_{(i,j)}$	$\mathbf{p}_{(i,j)}$	$\mathbf{v}_{(i,j)}$	**Cell (i, j)**	$\mathbf{u}_{(i,j)}$	$\mathbf{p}_{(i,j)}$	$\mathbf{v}_{(i,j)}$
$(00, 00)$	7	1	7	$(11, 11)$	7	1	7
$(00, 01)$	1	1	1	$(11, 10)$	1	-1	-1
$(00, 10)$	1	1	1	$(11, 01)$	1	1	1
$(00, 11)$	1	1	1	$(11, 00)$	1	1	1
$(01, 00)$	1	1	1	$(10, 11)$	1	1	1
$(01, 01)$	1	1	1	$(10, 10)$	1	1	1
$(01, 10)$	0	-1	0	$(10, 01)$	0	1	0
$(01, 11)$	1	1	1	$(10, 00)$	1	-1	-1

Table 2: Values of $p_{(i,j)}$ and $v_{(i,j)}$

Following other work in mosaic systems, $|\mathbb{S}^{(m,n)}|$ can be calculated more efficiently than in Equation 5 using the state matrix recursion introduced in [9].

5 Proof of Theorem 2.1

This argument is an adaptation of the proof in [9] for binary lattices.

Let a *binary sub-lattice* of size (p, q) be a rectangular lattice of $p + 1$ by $q + 1$ vertices, in which only the left and right boundary vertices must be labeled 0, and all other vertices can be labeled 0 or 1. Also let $\hat{\mathbb{L}}^{(p,q)}$ be the set of all binary sub-lattices of size (p, q). We choose a similar indexing convention for binary sub-lattices as individual cells, in which we, ignoring the first and last 0, read the bottom row and the top row as two binary numbers (i, j) respectively. For example, Figure 9 is a binary sub-lattice of size $(2, 4)$ with index $(011, 100)$.

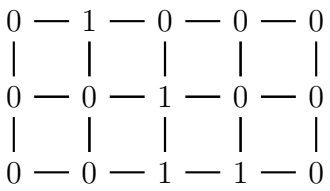


Figure 9: A binary sub-lattice of size $(2, 4)$ with index $(011, 100)$

Note that for $p > 1$, this index does not uniquely define the binary sub-lattice.

As with binary lattices, we can compute $V(\hat{\ell})$ for a binary sub-lattice $\hat{\ell} \in \hat{\mathbb{L}}^{(p,q)}$. We can now define the *state matrix* for $\hat{\mathbb{L}}^{(p,q)}$ to be the $2^q \times 2^q$ matrix $A^{(p,q)} = (A_{i,j})$ where element

$$A_{i,j} = \sum_{\hat{\ell} \text{ with index } (i,j)} V(\hat{\ell}) = s_{p,q}.$$

Here $A_{i,j}$ is the entry in the i-th row of the matrix, read top-to-bottom, and in the j-th column of the matrix read left-to-right. As the binary sub-lattices $\hat{\mathbb{L}}^{(p,q)}$ with index $(0 \ldots 0, 0 \ldots 0)$ are just the binary lattices $\mathbb{L}^{(p,q)}$, we have for a state matrix $A^{(p,q)}$ that

$$A_{0,0} = \sum_{\ell \in \mathbb{L}^{(p,q)}} V(\ell).$$

Theorem 2.1 amounts to an efficient procedure to compute $A^{(p,q)}$.

Proposition 5.1. *For the set $\hat{\mathbb{L}}^{(1,q)}$ the associated state matrix $A^{(1,q)}$ can be computed by first defining $A^{(1,2)} = \begin{bmatrix} 7^2 & -1 \\ 1 & 1 \end{bmatrix}$. We recursively define $A^{(1,q)} \in \mathbb{Z}^{2^q \times 2^q}$ given $A^{(1,q-1)}$. Begin by writing $A^{(1,k)} = \begin{bmatrix} A_{[0,0]} & A_{[0,1]} \\ A_{[1,0]} & A_{[1,1]} \end{bmatrix}$, where the block matrices $A_{[i,j]}$ are square block matrices of size $2^{k-1} \times 2^{k-1}$. We then have*

$$A^{(1,k+1)} = \begin{bmatrix} 7A_{[0,0]} & 7A_{[0,1]} & 7^{-1}A_{[0,0]} & A_{[0,1]} \\ 7A_{[1,0]} & 7A_{[1,1]} & 0A_{[1,0]} & A_{[1,1]} \\ -7^{-1}A_{[0,0]} & 0A_{[0,1]} & 7^{-1}A_{[0,0]} & A_{[0,1]} \\ A_{[1,0]} & A_{[1,1]} & -A_{[1,0]} & 7A_{[1,1]} \end{bmatrix}.$$

Construct $A^{(1,q)}$ by starting with $k = 2$ and recursing until $k = q$.

Proof. We use induction on q. We can immediately calculate the entries of $A^{(1,2)}$ by listing all size $(1, 2)$ binary sub-lattices, then using Equation 5.

$$\begin{array}{ccc} 0 & - & 0 & - & 0 \\ | & & | & & | \\ 0 & - & 0 & - & 0 \end{array} \qquad \begin{array}{ccc} 0 & - & 1 & - & 0 \\ | & & | & & | \\ 0 & - & 0 & - & 0 \end{array}$$

$$\begin{array}{ccc} 0 & - & 0 & - & 0 \\ | & & | & & | \\ 0 & - & 1 & - & 0 \end{array} \qquad \begin{array}{ccc} 0 & - & 1 & - & 0 \\ | & & | & & | \\ 0 & - & 1 & - & 0 \end{array}$$

Above gives

$$\begin{bmatrix} v_{(00,00)}v_{(00,00)} & v_{(00,01)}v_{(00,10)} \\ v_{(01,00)}v_{(10,00)} & v_{(01,01)}v_{(10,10)} \end{bmatrix} = \begin{bmatrix} 7^2 & -1 \\ 1 & 1 \end{bmatrix}.$$

We then assume that the statement for $A^{(1,k)}$ is true up to k. Within $A^{(1,k)}$, consider all the size $(1, k)$ binary sub-lattices counted in the entries of the upper left $2^{k-1} \times 2^{k-1}$ block matrix, which we denote $A_{[0,0]}$. By construction, all of these binary sub-lattices have indexes of the form $(0 \ldots, 0 \ldots)$, in which both indexes begin with a 0. Similarly, the upper right $2^{k-1} \times 2^{k-1}$ block matrix $A_{[0,1]}$ counts all binary sub-lattices with indexes like $(0, \ldots, 1 \ldots)$, and so on.

We then append one of each of the four cells in Figure 10 to the left of every size $(1, k)$ binary sub-lattices, such that the former left boundary 0s are replaced.

$$\begin{array}{ccc} 0 & - & 0 \\ | & & | \\ 0 & - & 0 \\ \multicolumn{3}{c}{\textbf{(00, 00)}} \end{array} \qquad \begin{array}{ccc} 0 & - & 1 \\ | & & | \\ 0 & - & 0 \\ \multicolumn{3}{c}{\textbf{(00, 01)}} \end{array}$$

$$\begin{array}{ccc} 0 & - & 0 \\ | & & | \\ 0 & - & 1 \\ \multicolumn{3}{c}{\textbf{(01, 00)}} \end{array} \qquad \begin{array}{ccc} 0 & - & 1 \\ | & & | \\ 0 & - & 1 \\ \multicolumn{3}{c}{\textbf{(01, 01)}} \end{array}$$

Figure 10: Appending cells

An example of appending cell $(01, 00)$ to the size $(1, 3)$ binary sub-lattice $(11, 10)$ is shown below.

$$\begin{array}{ccccccccccccc} 0 & - & 0 & & 0 & - & 1 & - & 0 & - & 0 & & 0 & - & 0 & - & 1 & - & 0 & - & 0 \\ | & & | & + & | & & | & & | & & | & \to & | & & | & & | & & | & & | \\ 0 & - & 1 & & 0 & - & 1 & - & 1 & - & 0 & & 0 & - & 1 & - & 1 & - & 1 & - & 0 \end{array}$$

This creates the size $(1, k+1)$ binary sub-lattice $(111, 010)$. Notice this operation only changes the identity of the two left-most cells, namely changing the $(01, 01)$ cell into cells $(01, 00)$ and $(11, 01)$.

By construction, as every binary sub-lattice counted in the block matrix $A_{[1,1]}$ has an index of the form $(1\ldots, 1\ldots)$, the entries in $A_{[1,1]}$ are all divisible by $v_{01,01}$. Therefore, the size $(1, k+1)$ binary sub-lattices with index of the form $(11\ldots, 01\ldots)$ are counted by $(v_{01,00}v_{11,01}/v_{01,01})A_{[1,1]}$.

Performing the appending operation each of the four cells from Figure 10 to each other gives 16 possible pairs, which we diagram below such that it is consistent with the indexing for $A^{(1,k)}$.

$$\begin{array}{cccc} 0-0-0 & 0-0-1 & 0-1-0 & 0-1-1 \\ |\ \ |\ \ | & |\ \ |\ \ | & |\ \ |\ \ | & |\ \ |\ \ | \\ 0-0-0 & 0-0-0 & 0-0-0 & 0-0-0 \end{array}$$

$$\begin{array}{cccc} 0-0-0 & 0-0-1 & 0-1-0 & 0-1-1 \\ |\ \ |\ \ | & |\ \ |\ \ | & |\ \ |\ \ | & |\ \ |\ \ | \\ 0-0-1 & 0-0-1 & 0-0-1 & 0-0-1 \end{array}$$

$$\begin{array}{cccc} 0-0-0 & 0-0-1 & 0-1-0 & 0-1-1 \\ |\ \ |\ \ | & |\ \ |\ \ | & |\ \ |\ \ | & |\ \ |\ \ | \\ 0-1-0 & 0-1-0 & 0-1-0 & 0-1-0 \end{array}$$

$$\begin{array}{cccc} 0-0-0 & 0-0-1 & 0-1-0 & 0-1-1 \\ |\ \ |\ \ | & |\ \ |\ \ | & |\ \ |\ \ | & |\ \ |\ \ | \\ 0-1-1 & 0-1-1 & 0-1-1 & 0-1-1 \end{array}$$

The associated appending equations for the $v_{i,j}$ values are then

$$\begin{array}{cccc} \frac{v_{(00,00)}v_{(00,00)}}{v_{(00,00)}} = 7 & \frac{v_{(00,00)}v_{(00,01)}}{v_{(00,01)}} = 7 & \frac{v_{(00,01)}v_{(00,10)}}{v_{(00,00)}} = 7^{-1} & \frac{v_{(00,01)}v_{(00,11)}}{v_{(00,01)}} = 1 \\ \frac{v_{(00,00)}v_{(01,00)}}{v_{(01,00)}} = 7 & \frac{v_{(00,00)}v_{(01,01)}}{v_{(01,01)}} = 7 & \frac{v_{(00,01)}v_{(01,10)}}{v_{(01,00)}} = 0 & \frac{v_{(00,01)}v_{(01,11)}}{v_{(01,01)}} = 1 \\ \frac{v_{(01,00)}v_{(10,00)}}{v_{(00,00)}} = -7^{-1} & \frac{v_{(01,00)}v_{(10,01)}}{v_{(00,01)}} = 0 & \frac{v_{(01,01)}v_{(10,10)}}{v_{(00,00)}} = 7^{-1} & \frac{v_{(01,01)}v_{(10,11)}}{v_{(00,01)}} = 1 \\ \frac{v_{(01,00)}v_{(11,00)}}{v_{(01,00)}} = 1 & \frac{v_{(01,00)}v_{(11,01)}}{v_{(01,01)}} = 1 & \frac{v_{(01,01)}v_{(11,10)}}{v_{(01,00)}} = -1 & \frac{v_{(01,01)}v_{(11,11)}}{v_{(01,01)}} = 7 \end{array}$$

We can then determine

$$A^{(1,k+1)} = \begin{bmatrix} 7A_{[0,0]} & 7A_{[0,1]} & 7^{-1}A_{[0,0]} & A_{[0,1]} \\ 7A_{[1,0]} & 7A_{[1,1]} & 0A_{[1,0]} & A_{[1,1]} \\ -7^{-1}A_{[0,0]} & 0A_{[0,1]} & 7^{-1}A_{[0,0]} & A_{[0,1]} \\ A_{[1,0]} & A_{[1,1]} & -A_{[1,0]} & 7A_{[1,1]} \end{bmatrix}.$$

□

Proposition 5.2. *For the set $\mathbb{L}^{(p,q)}$, the state matrix has*

$$A^{(p,q)} = (A^{(1,q)})^p.$$

Proof. We use induction on p, and assume $A^{(k,q)} = (A^{(1,q)})^k$. Therefore, $A_{i,j}^{(k,q)}$

$$A_{i,j}^{(k,q)} = \sum_{\hat{\ell} \text{ with index } (i,j)} V(\hat{\ell})$$

for $\hat{\ell}$ of size (k,q) and index (i,j). Notice that we can build all binary sub-lattices of size $(k+1,q)$ with index (i,j) by adjoining a size $(1,q)$ binary sub-lattice of index (i,s) and a size (k,q) binary sub-lattice of index (s,j). For example, Figure 11 adjoins a size $(2,3)$ binary sub-lattice with index $(10,01)$ to a size $(1,3)$ binary sub-lattice with index $(01,11)$.

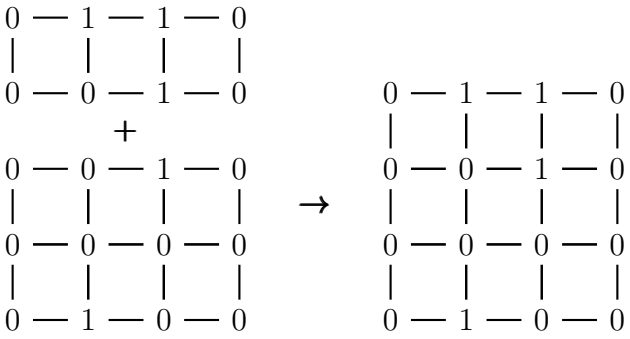


Figure 11: Adjoins a size $(1,3)$ binary sub-lattice to a size $(2,3)$ sub-lattice

This operation amounts to

$$A_{i,j}^{(k+1,q)} = \sum_{s=1}^{2^q} A_{i,s}^{(k,q)} A_{s,j}^{(1,q)},$$

which gives

$$A^{(k+1,q)} = A^{(k,q)} A^{(1,q)} = (A^{(1,q)})^{k+1}.$$

□

6 Acknowledgements

The authors would like to thank Michael Maltenfort for the edits, improvements and ideas for this paper.

References

[1] Dooho Choi et al. "Quantum knot mosaics and bounds of the growth constant". In: *Reviews in Mathematical Physics* 36.10 (2024), p. 2450025. DOI: 10.1142/S0129055X24500259. eprint: https://doi.org/10.1142/S0129055X24500259. URL: https://doi.org/10.1142/S0129055X24500259.

[2] Kyungpyo Hong and Seungsang Oh. "Bounds on Multiple Self-avoiding Polygons". In: *Canadian Mathematical Bulletin* 61.3 (Sept. 2018), pp. 518–530. ISSN: 1496-4287. DOI: 10.4153/cmb-2017-072-x. URL: http://dx.doi.org/10.4153/CMB-2017-072-x.

[3] Samuel J. Lomonaco and Louis H. Kauffman. "Quantum knots and mosaics". In: *Quantum Information Processing* 7.2 (2008), pp. 85–115. DOI: 10.1007/s11128-008-0076-7. URL: https://doi.org/10.1007/s11128-008-0076-7.

[4] OEIS Foundation Inc. *The On-Line Encyclopedia of Integer Sequences*. Published electronically at http://oeis.org.

[5] Seungsang Oh. "Domino tilings of the expanded Aztec diamond". In: *DISCRETE MATHEMATICS* 341.4 (Apr. 2018), pp. 1185–1191. ISSN: 0012-365X. DOI: 10.1016/j.disc.2017.10.016.

[6] Seungsang Oh. "Quantum knot mosaics and the growth constant". In: *Topology and its Applications* 210 (2016), pp. 311–316. ISSN: 0166-8641. DOI: https://doi.org/10.1016/j.topol.2016.08.011. URL: https://www.sciencedirect.com/science/article/pii/S0166864116301857.

[7] Seungsang Oh. "State matrix recursion method and monomer-dimer problem". In: *DISCRETE MATHEMATICS* 342.5 (May 2019), pp. 1434–1445. ISSN: 0012-365X. DOI: 10.1016/j.disc.2019.01.022.

[8] Seungsang Oh and Youngin Kim. "Growth rate of quantum knot mosaics". In: *Quantum Information Processing* 18.8 (2019), p. 238. DOI: 10.1007/s11128-019-2353-z. URL: https://doi.org/10.1007/s11128-019-2353-z.

[9] Seungsang Oh et al. "Quantum knots and the number of knot mosaics". In: *Quantum Information Processing* 14.3 (2015), pp. 801–811. DOI: 10.1007/s11128-014-0895-7. URL: https://doi.org/10.1007/s11128-014-0895-7.