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. The authors call these mosaics *polygon mosaics* because the tiles in a suitably connected mosaic form polygons. We introduce and enumerate mosaics with the related property of containing at least one polygon, 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.

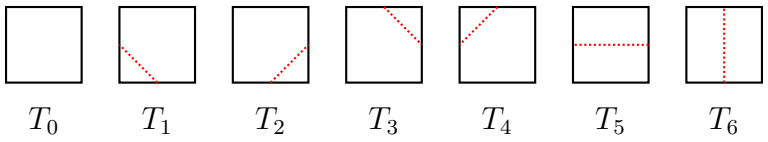
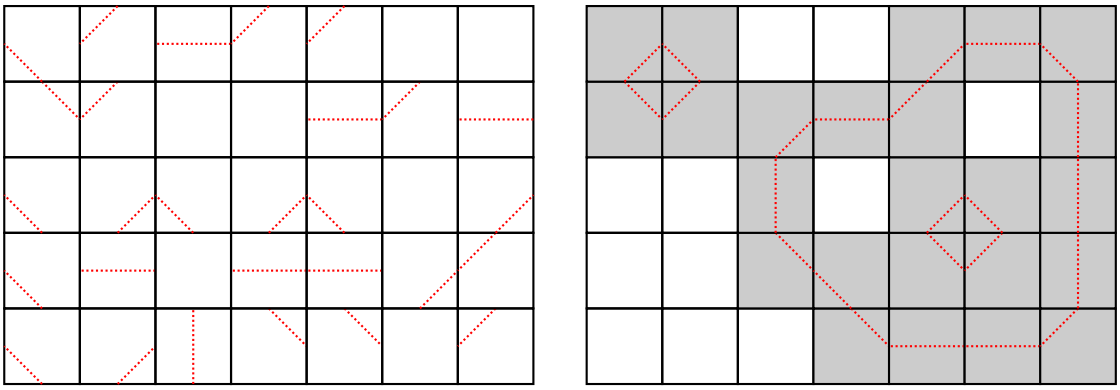


Figure 1: The tile set $\mathbb{T}$

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 a $(5, 7)$ 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] were 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 mosaics that are suitably connected and that contain at least one tile in $\mathbb{T} \setminus T_0$ *polygon mosaics* because



(a) A mosaic (b) A polygon mosaic

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

the dotted lines form *polygons*[1]. Following [2], when we use the term polygon in this work, we specifically refer to the set of tiles for which the dotted lines form a polygon when part of a mosaic.

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

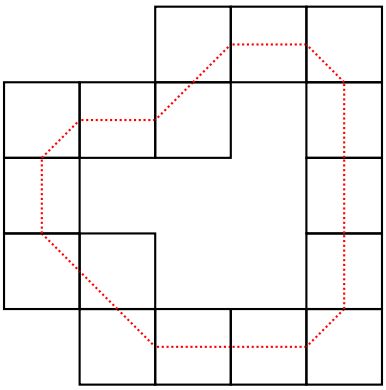


Figure 3: A polygon in Figure 2b

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

Theorem 1.1 ([2])**.** *The number of polygon* (m, n) *mosaics* $|\mathbb{P}^{(m,n)}|$ *for* $m, n \geq 3$ *is bounded by*

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

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

The array of values of $|\mathbb{P}^{(m,n)}| + 1$ is sequence A181245 on the OEIS [4, OEIS].

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 [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 enumerate them 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.

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

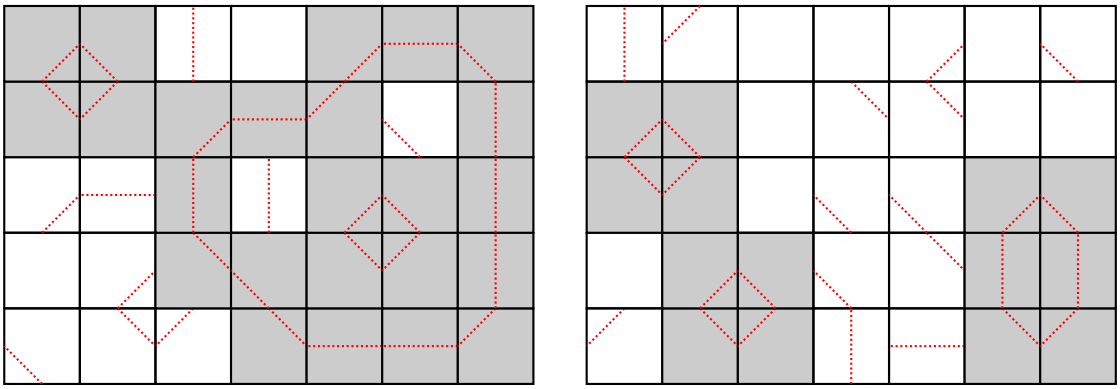


Figure 4: Messy polygon mosaics

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 subset 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 $|\mathbb{S}^{(n,1)}| = 7^n$, and $|\mathbb{S}^{(2,2)}| = 7^4 - 1$. For $n, m \geq 2$, we first define the state matrices for messy polygon mosaics.

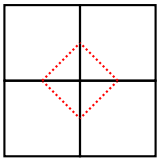


Figure 5: The smallest polygon

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

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

Our main result is Theorem 2.1.

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 an (m,n) *binary lattice*. An (m,n) binary lattice 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 $(5,7)$ binary lattice is shown on the right of Figure 6. Also let $\mathbb{L}^{(m,n)}$ be the set of all (m,n) binary lattices, 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.

Notice that by the definition of f, in a binary lattice polygons draw out the boundary between edge-connected vertices with the same label, excluding the vertices that are edge-connected to the boundary 0's. For example in Figure 6 the three polygons correspond with the boundary of the three edge-connected regions of vertices that are not edge-connected to the boundary 0's.

To enumerate $|\mathbb{S}^{(m,n)}|$, it will be useful to consider how f maps individual tiles in $\mathbb{T}$ to individual *cells* in a binary lattice.

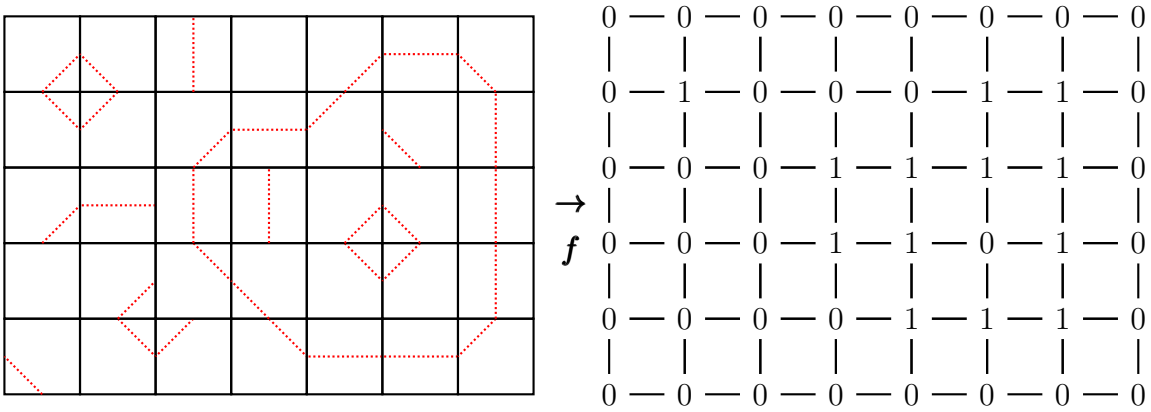
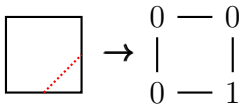


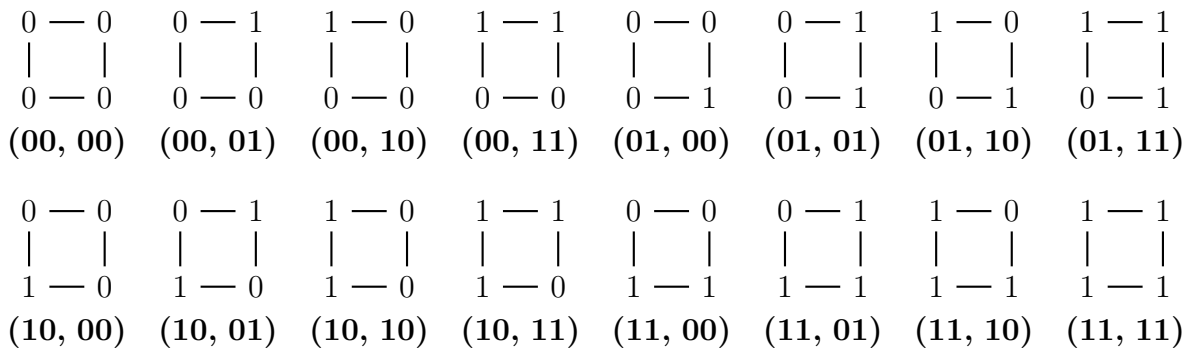
Figure 6: f applied to the left mosaic in Figure 4, resulting in a binary lattice

Definition 3.2. Let a *cell* be the unit square and its 4 labeled vertices in a binary lattice that a given tile maps to, along with its coordinate tuple.

Example 3.1. Applying f to the mosaic in Figure 6 results in the top-left tile T_2 mapping to the cell at coordinates $(0, 0)$ diagrammed below.



Let C be the set of unique cells ignoring location. For convenience, we give a pair of indexes to each of the $|C| = 2^4$ possible vertex configurations. 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 in bold below.



From the definition of f, mosaics that do not contain a polygon all map to the binary lattice of all $(00, 00)$ cells.

Definition 3.3. Let ℓ^* be the binary lattice with all cells having index $(00, 00)$.

Therefore, we have

$$\mathbb{S}^{(m,n)} = f^{-1}(\{\ell^*\}). \tag{1}$$

It suffices to compute $|f^{-1}(\{\ell\})|$ for a given binary lattice ℓ. We begin by examining the number of tiles in $\mathbb{T}$ that can map to a cell with index (i, j), which we denote $u_{(i,j)}$. 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 a cell with index $(01, 00)$ 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 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).

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 under f

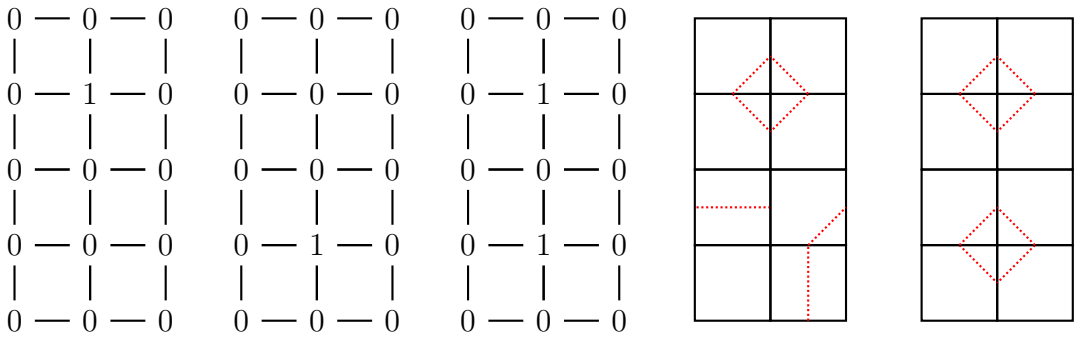
Definition 3.4.

$$U(\ell) := \prod_{\{\text{index}(i,j) \text{ of cell } |\text{cell} \in \ell\}} u_{(i,j)} \tag{2}$$

Intuitively, we would hope that for a binary lattice ℓ, $U(\ell)$ would equal $|f^{-1}(\{\ell\})|$. However this is not necessarily the case, as $U(\ell)$ does not *just* count the number of mosaics that map to ℓ under f.

Example 3.2. We have $U(\ell^*) = 7^{mn} = |\mathbb{M}^{(m,n)}|$, which clearly overcounts $|\mathbb{S}^{(m,n)}|$.

Example 3.3. Consider the following binary lattices and mosaics for $(m, n) = (4, 2)$.



We have $U(\ell_1) = 7^4, U(\ell_2) = 7^4$, and $U(\ell_3) = 1$. $U(\ell_1)$ counts Mosaic 1, but $U(\ell_1)$, $U(\ell_2)$ and $U(\ell_3)$ each count Mosaic 2, for which $f(\text{Mosaic 2}) = \ell_3$. This over counting is due to the cell with index $(00, 00)$ coming from any possible tile in $\mathbb{T}$. Each cell in the bottom two rows of ℓ_1 could come from 7 possible tiles, so 1 of the 7^4 permutations forms a new polygon. This is also true for the top two rows of ℓ_2. Therefore we can write $U(\ell_1) = |f^{-1}(\ell_1)| + |f^{-1}(\ell_3)|$, $U(\ell_2) = |f^{-1}(\ell_2)| + |f^{-1}(\ell_3)|$, and $U(\ell_3) = |f^{-1}(\ell_3)|$. This gives

$$U(\ell_1) + U(\ell_2) + U(\ell_3) = |f^{-1}(\ell_1)| + |f^{-1}(\ell_2)| + 3|f^{-1}(\ell_3)|,$$

so Mosaic 2 is counted three times.

Examples 3.2 and 3.3 show we cannot compute $|f^{-1}(\{\ell\})|$ for a given binary lattice ℓ using $U(\ell)$, as cells $(00, 00)$ and $(11, 11)$ lead to over counting mosaics. Instead, U counts the number of mosaics that map to some set of binary lattices under f. This set is defined in Definition 3.8. To define this set, we first point out that a binary lattice can be split into the cells that have $u_{(i,j)} = 7$ and the cells that have $u_{(i,j)} = 1$.

Definition 3.5. A polygon $\mathcal{P}$ is *specified* by a binary lattice ℓ if the cells in ℓ that map to the tiles that make up $\mathcal{P}$ all have $u_{(i,j)} = 1$.

Definition 3.6. Let $P : \mathbb{L}^{(m,n)} \to \mathbb{N}$ be the number of polygons specified in a binary lattice ℓ.

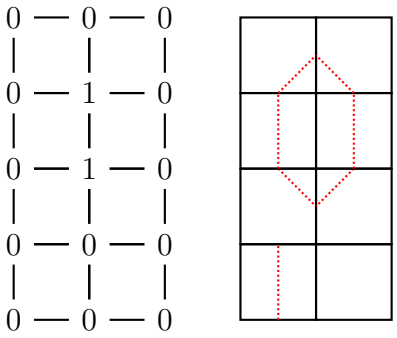
Example 3.4. From Example 3.3, ℓ_1 specifies the polygon in the top 2 rows of Mosaic 2, as the cells in the top 2 rows of ℓ_1 all have $u_{(i,j)} = 1$. However ℓ_1 does not specify the polygon in the bottom 2 rows of Mosaic 2, as the cells in the bottom 2 rows of ℓ_1 all have $u_{(i,j)} = 7$. ℓ_3 specifies both polygons in Mosaic 2. Therefore $P(\ell_1) = P(\ell_2) = 1$, and $P(\ell_3) = 2$.

Example 3.5. The binary lattice ℓ^* specifies 0 polygons, and so $P(\ell^*) = 0$.

Definition 3.7. For binary lattices ℓ, ℓ', we say $\ell \leq \ell'$ if one can construct ℓ' by flipping the parity of some number of vertices (possibly 0 vertices) in ℓ, such that all polygons specified by ℓ are unchanged and no cells with index $(01, 10)$ or $(10, 01)$ are created.

Definition 3.8. For a binary lattice ℓ, let $\mathbb{U}(\ell) = \{\ell' | \ell \leq \ell'\}$.

Example 3.6. From Example 3.3, $\ell_1 \leq \ell_3$ as one can construct ℓ_3 from ℓ_1 by flipping the vertex at $(3,1)$ from 0 to 1. As ℓ_1 specifies the polygon in the top two rows of Mosaic 2, and this vertex flip does not change the top polygon, $\ell_1 \leq \ell_3$. We also have $\ell_1 \leq \ell_1$ simply by selecting no vertices to flip. As diagrammed below, constructing ℓ_4 from ℓ_1 by flipping the vertex at $(2,1)$ does change a polygon specified by ℓ_1, and so $\ell_1 \not\leq \ell_4$.



ℓ_4 Mosaic 3

Therefore $\mathbb{U}(\ell_1) = \{\ell_1, \ell_3\}$, $\mathbb{U}(\ell_2) = \{\ell_2, \ell_3\}$, and $\mathbb{U}(\ell_3) = \{\ell_3\}$.

Example 3.7. Every binary lattice ℓ has $\ell^* \leq \ell$, as ℓ^* specifies 0 polygons. Therefore for a given (m,n), we have $\mathbb{U}(\ell^*) = \mathbb{L}^{(m,n)}$.

Proposition 3.1. *For all ℓ we have*

$$U(\ell) = \sum_{\ell' \in \mathbb{U}(\ell)} |f^{-1}(\{\ell'\})|. \tag{3}$$

Proof. For a binary lattice $\ell' \in \mathbb{U}(\ell)$, ℓ' specifies new polygons without changing any of the polygons specified in ℓ. Consider the set of cells S that differ between ℓ and ℓ'. By definition, these cells in ℓ all have $u_{(i,j)} = 7 = |\mathbb{T}|$, while in ℓ' they all have $u_{(i,j)} = 1$. Also by definition $U(\ell)$ enumerates every possible combination of tiles that could map to each cell in S. Therefore, the tiles that for the new polygons specified in ℓ' is counted, and so $|f^{-1}(\ell')|$ is also counted by U. □

Surprisingly, we can recover $|\mathbb{S}^{(m,n)}|$ by first considering how much each mosaic is over-counted in

$$\sum_{\ell \in \mathbb{L}^{(m,n)}} U(\ell). \tag{4}$$

Proposition 3.2.

$$\sum_{\ell \in \mathbb{L}^{(m,n)}} U(\ell) = \sum_{\ell \in \mathbb{L}^{(m,n)}} \sum_{\ell' \in \mathbb{U}(\ell)} |f^{-1}(\{\ell'\})| = \sum_{\ell \in \mathbb{L}^{(m,n)}} \left(\sum_{p=0}^{P(\ell)} \binom{P(\ell)}{p} \right) |f^{-1}(\{\ell\})|. \tag{5}$$

Proof. The first equality follows directly from Proposition 3.1. For the second equality, for a given binary lattice ℓ the coefficient of $|f^{-1}(\{\ell\})|$ is the number of elements in $\{\ell' | \ell' \leq \ell\}$. Therefore, a binary lattice is in $\{\ell' | \ell' \leq \ell\}$ if the binary lattice only specifies $0 \leq p \leq P(\ell)$ of the $P(\ell)$ polygons specified by ℓ. As there are $\binom{P(\ell)}{p}$ ways to specify p polygons from a choice of $P(\ell)$ polygons, this gives the second equality for all $P(\ell) > 0$.

If ℓ' has $P(\ell') = 0$, then $\ell' = \ell^*$. As $\{\ell' | \ell' \leq \ell^*\} = \{\ell^*\}$, the term $|f^{-1}(\ell')|$ is only counted once. As $1 = \binom{0}{0}$, this completes the proof. □

Proposition 3.3. *The number of (m, n) mosaics that do not contain a polygon $|\mathbb{S}^{(m,n)}|$ has*

$$|\mathbb{S}^{(m,n)}| = \sum_{\ell \in \mathbb{L}^{(m,n)}} (-1)^{P(\ell)} U(\ell). \tag{6}$$

Proof. If we group terms like the second equality of Equation 5, we get

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

By the binomial theorem, for $P(\ell) > 0$ we have

$$\sum_{p=0}^{P(\ell)} (-1)^{P(\ell)} \binom{P(\ell)}{p} = (1-1)^{P(\ell)} = 0,$$

so all terms where $P(\ell) > 0$ are 0. As the only binary lattice with $P(\ell) = 0$ is ℓ^*, we have

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

□

4 A Cell-Level Identity

Though Equation 6 does compute $|\mathbb{S}^{(m,n)}|$, computing the number of polygons specified in a binary lattice $P(\ell)$ requires examining the entire structure of ℓ. It will be more efficient to recover the $(-1)^{P(\ell)}$ term at the level of individual cells. 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

$$(-1)^{P(\ell)} U(\ell) = V(\ell). \tag{7}$$

The idea is to add a coefficient $p_{(i,j)} \in \{-1, 1\}$ to each value of $u_{(i,j)}$ in such a way as to recover the $(-1)^{P(\ell)}$ term. With this in mind 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*

$$\prod_{Cell\ (i,j)\in\mathcal{S}} p_{(i,j)} = -1, \tag{8}$$

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

Cells in a binary lattice either do or do not specify a polygon. To recover the $(-1)^{P(\ell)}$ term, the following must be true. Cells that cannot specify a polygon must all have $p_{(i,j)} = 1$. Moreover, if a set of cells specifies any one polygon, it must meet Condition 4.1.

Proposition 4.2. *There exists values $p_{(i,j)}$ for which the cells that do not specify a polygon have $p_{(i,j)} = 1$, and Condition 4.1 holds for any set of cells that specify a polygon.*

Proof. We can immediately see $p_{(00,00)} = p_{(11,11)} = p_{(01,10)} = p_{(10,01)} = 1$, as cells $(00, 00)$, $(11, 11)$, $(01, 10)$, and $(10, 01)$ can never be in the subset 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

For the portions of binary lattices in Figure 7, Condition 4.1 amounts to the following two equations

$$\begin{aligned} p_{(00,01)}p_{(00,10)}p_{(01,00)}p_{(10,00)} &= -1 \\ p_{(11,10)}p_{(11,01)}p_{(10,11)}p_{(01,11)} &= -1. \end{aligned} \tag{9}$$

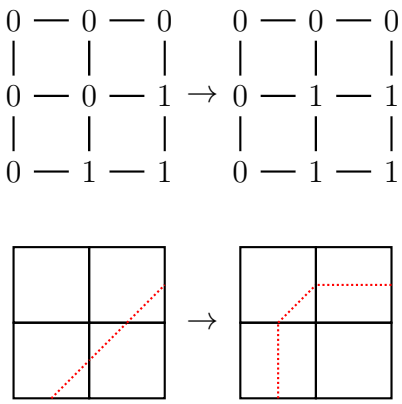
We refer to the equations above as *constraints*, as they constrain the possible assignments of $p_{(i,j)}$.

To define the remaining constraints, consider the unbounded square lattice with all vertices labeled 0, except for the origin labeled 1. We refer to this as the *starting position*.

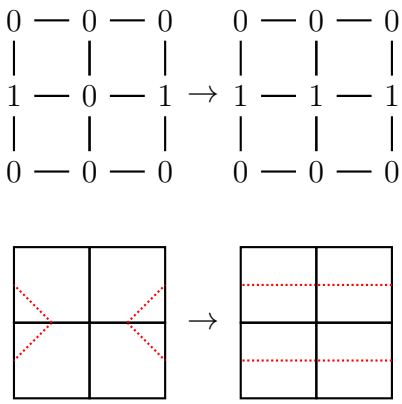
From the definition of f we know that polygons are the boundary of edge-connected vertices of the same label. Clearly one can construct arbitrary sets of edge-connected 0's and 1's from the starting position by flipping the parity of a vertex one-by-one.

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 udner this operation, we must define 2^8 constraints.

In each of the constraints, the parity of the number of edge-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*.



(a) Type 1 bit flip and associated tiles



(b) Type 2 bit flip and associated tiles

Figure 8: Bit flips for binary lattices

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

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

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

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

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 to adhere to Condition 4.1.

This gives a procedure to define the 2^8 bit flip constraints on $p_{(i,j)}$. Before solving for $p_{(i,j)}$, we can discard any constraint that involves cells $(01, 10)$ or $(10, 01)$, as we don't consider any binary lattices with these cells.

Finally, we can then use software to verify that all remaining constraints admit a solution. □

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

We can then write

$$\sum_{\ell \in \mathbb{L}^{(m,n)}} V(\ell) = |\mathbb{S}^{(m,n)}|. \tag{12}$$

Following other work with mosaics, $|\mathbb{S}^{(m,n)}|$ can be calculated more efficiently than in Equation 12 using the state matrix recursion introduced in [9].

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 $v_{(i,j)}$

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 $(011, 100)$.

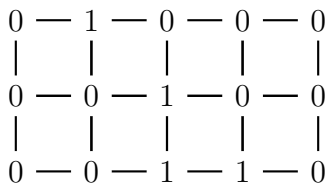


Figure 9: A binary sub-lattice of size $(2, 4)$ with index $(011, 100)$

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 $A^{(p,q)} = (A^{(p,q)}_{i,j})$ where element

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

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

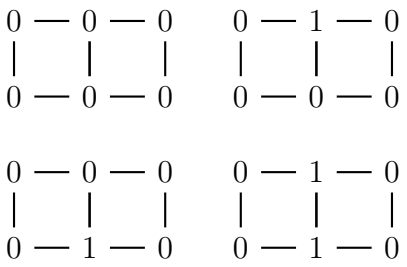
$$A^{(p,q)}_{0,0} = \sum_{\ell \in \mathbb{L}^{(p,q)}} V(\ell).$$

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*

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

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 the definition of V.



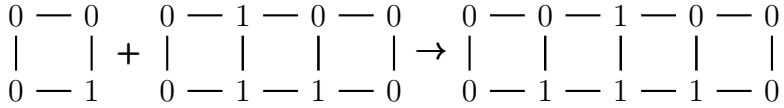
Above gives

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

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.

To construct binary sub-lattices of size $(1,k+1)$, 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 0's are replaced.

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



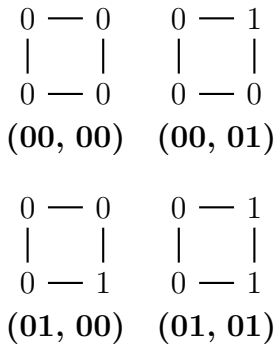
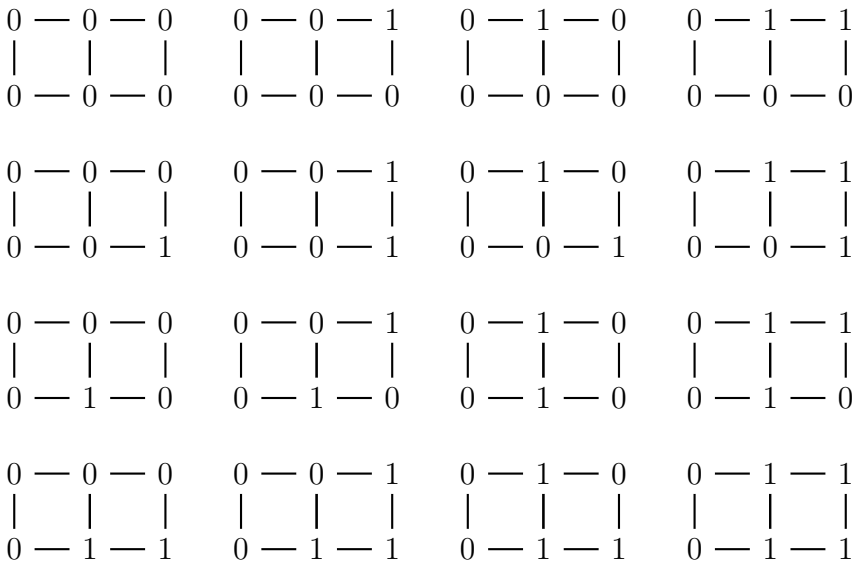


Figure 10: Appending cells

This creates the size $(1, 4)$ binary sub-lattice $(111, 010)$. Notice this operation only changes the identity of the two left-most cells, namely changing the left-most $(01, 01)$ 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 with 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 $A^{(1,k)}$.



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

$$\frac{v_{(00,00)}v_{(00,00)}}{v_{(00,00)}} = 7 \quad \frac{v_{(00,00)}v_{(00,01)}}{v_{(00,01)}} = 7 \quad \frac{v_{(00,01)}v_{(00,10)}}{v_{(00,00)}} = 7^{-1} \quad \frac{v_{(00,01)}v_{(00,11)}}{v_{(00,01)}} = 1$$
$$\frac{v_{(00,00)}v_{(01,00)}}{v_{(01,00)}} = 7 \quad \frac{v_{(00,00)}v_{(01,01)}}{v_{(01,01)}} = 7 \quad \frac{v_{(00,01)}v_{(01,10)}}{v_{(01,00)}} = 0 \quad \frac{v_{(00,01)}v_{(01,11)}}{v_{(01,01)}} = 1$$
$$\frac{v_{(01,00)}v_{(10,00)}}{v_{(00,00)}} = -7^{-1} \quad \frac{v_{(01,00)}v_{(10,01)}}{v_{(00,01)}} = 0 \quad \frac{v_{(01,01)}v_{(10,10)}}{v_{(00,00)}} = 7^{-1} \quad \frac{v_{(01,01)}v_{(10,11)}}{v_{(00,01)}} = 1$$
$$\frac{v_{(01,00)}v_{(11,00)}}{v_{(01,00)}} = 1 \quad \frac{v_{(01,00)}v_{(11,01)}}{v_{(01,01)}} = 1 \quad \frac{v_{(01,01)}v_{(11,10)}}{v_{(01,00)}} = -1 \quad \frac{v_{(01,01)}v_{(11,11)}}{v_{(01,01)}} = 7$$

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,

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

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 the following. Append 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 $(01,11)$.

Doing this append operation for all s amounts to

$$A_{i,j}^{(k+1,q)} = \sum_{s=0}^{2^q-1} A_{i,s}^{(k,q)} A_{s,j}^{(1,q)},$$

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.

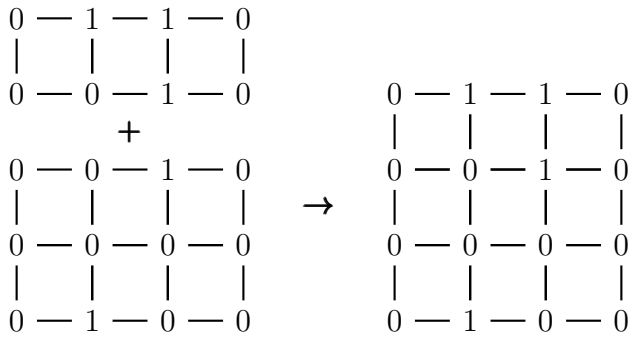


Figure 11: Appending a size $(1, 3)$ binary sub-lattice to a size $(2, 3)$ sub-lattice

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.