Enumeration of Messy Knot Mosaics

Jack Hanke
Northwestern University

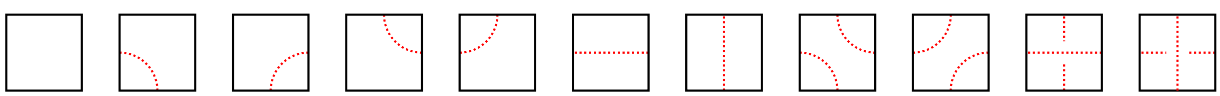
Richard Schank

Abstract

Lomonaco and Kauffman introduced a system of mosaics to model quantum knots. These systems of mosaics are composed of an $m \times n$ rectangular grid of 11 possible tiles. Oh and colleagues introduced a state matrix recursion method to exactly enumerate a subset of these mosaics that have the property of being suitably connected, which they call knot mosaics. We introduce and enumerate mosaics with the related property in which only some tiles must be suitably connected, which we call messy knot mosaics.

1 Introduction

Lomonaco and Kauffman [3] introduced a model for quantum knots in which an $m \times n$ matrix is constructed using 11 distinct symbols called *tiles*. These tiles, diagrammed below, are composed of unit squares with dotted lines connecting 2 or 4 sides at their midpoint.

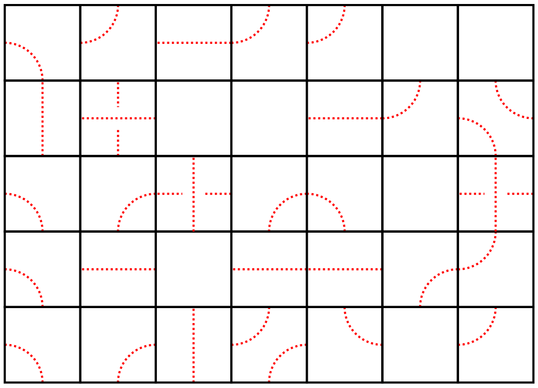


T_0 T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10}

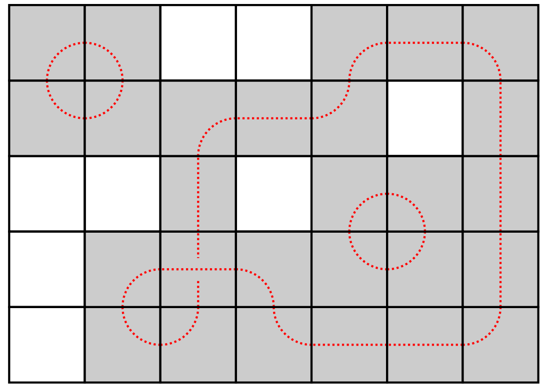
We denote the set of tiles $\mathbb{T} = \{T_0, \ldots, T_{10}\}$. A *mosaic* of size (m, n) is an $m \times n$ matrix made up of elements from $\mathbb{T}$. Figure 1a shows an example mosaic of size $(5, 7)$. We denote the set of all mosaics of size (m, n) as $\mathbb{M}^{(m,n)}$. As there are 11 elements in $\mathbb{T}$, there are 11^{mn} mosaics in $\mathbb{M}^{(m,n)}$. A *mosaic system* is then a subset of $\mathbb{M}^{(m,n)}$ with some property.

We 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 1a. The edge has either 0, 1, or 2 dotted lines drawn from its midpoint. Also note that the 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.

Lomonaco and Kauffman [3] call these *knot mosaics* because, other than the mosaic consisting of all T_0 tiles, the dotted lines form *knots*. Following the notation in [9] we denote the subset of mosaics of size (m, n) that are knot mosaics as $\mathbb{K}^{(m,n)}$. Figure 1b shows a knot mosaic of size $(5, 7)$ that contains 3 knots, with the tiles that make up the knots highlighted in gray. Note that a mosaic can contain knots isomorphic to the unknot, as well as knots that encompass other knots.



(a) A mosaic



(b) A knot mosaic

Figure 1: Examples of mosaics of size $(5, 7)$ made of tiles in $\mathbb{T}$

Let $k_{m,n} = |\mathbb{K}^{(m,n)}|$ be the number of knot mosaics of size (m, n). First notice that if either m or n is 1, one can only construct a knot mosaic using T_0 tiles, so $k_{m,1} = k_{1,n} = 1$. Oh et al. [9] showed the following for $m, n \geq 2$.

Theorem 1.1 ([9]). *The number of knot mosaics of size (m, n) for $m, n \geq 2$ is $k_{m,n} = 2\,\|(X_{m-2} + O_{m-2})^{n-2}\|$, where 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 $k = 0, 1, \ldots, m-3$, and $X_0 = O_0 = \begin{bmatrix} 1 \end{bmatrix}$. 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 knot mosaics*.

2 Messy Knot Mosaics

Definition 2.1. A *messy knot mosaic* is a mosaic that contains at least one knot.

Figure 2 shows two examples of messy knot mosaic of size $(5, 7)$ that contains 3 knots[1], with the tiles that make up the knots highlighted in gray. All knot mosaics are messy knot mosaics.

[1]Certain permutations of $\{T_1, T_2, T_3, T_4\}$ and $\{T_7, T_8\}$ can make shapes that appear to be knots but have hanging connections, as seen in the $(0, 4)$ position in the right example in Figure 2. These are not considered knots by this paper and all referenced works.

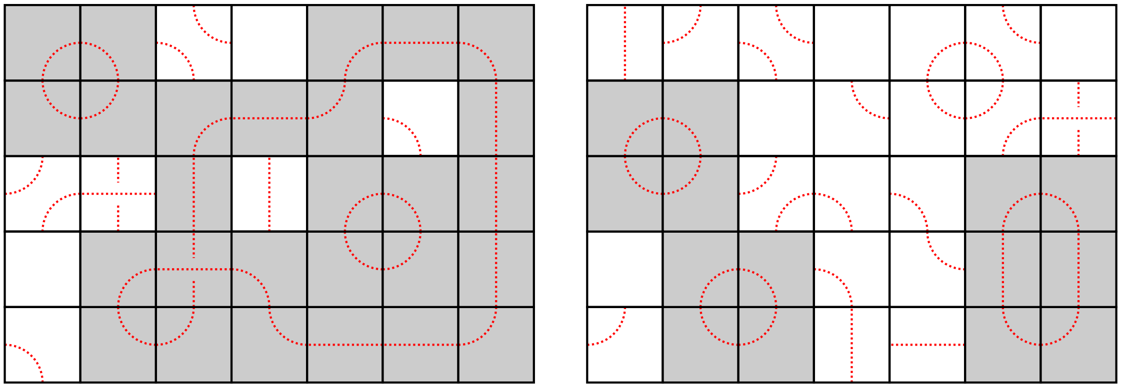


Figure 2: Messy knot mosaics

It turns out to be simpler to enumerate the number of mosaics that *do not* contain a knot. Therefore, let $\mathbb{S}^{(m,n)}$ be the set of mosaics that do not contain a knot, and let $|\mathbb{S}^{(m,n)}| = s_{m,n}$. Clearly the number of messy knot mosaics is then $11^{mn} - s_{m,n}$.

From the fact that the smallest knot is made of four tiles, shown in Figure 3, we can then conclude that $s_{n,1} = 11^n$, and $s_{2,2} = 11^4 - 1$. For $n, m \geq 2$, we first define the state matrices for messy knot mosaics.

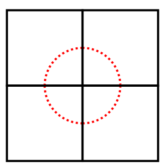


Figure 3: The smallest knot

Definition 2.2. Define $A^{(1,2)} = \begin{bmatrix} 11^2 & -1 \\ 1 & 1 \end{bmatrix}$. 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 have

$$A^{(1,k+1)} = \begin{bmatrix} 11A_{[0,0]} & 11A_{[0,1]} & 11^{-1}A_{[0,0]} & A_{[0,1]} \\ 11A_{[1,0]} & 11A_{[1,1]} & -4A_{[1,0]} & A_{[1,1]} \\ 11^{-1}A_{[0,0]} & 4A_{[0,1]} & 11^{-1}A_{[0,0]} & A_{[0,1]} \\ A_{[1,0]} & A_{[1,1]} & -A_{[1,0]} & 11A_{[1,1]} \end{bmatrix}.$$

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

Theorem 2.1. *The number of mosaics of size* (m, n) *that do not contain a knot is the* $(0, 0)$ *entry of* $(A^{(1,m)})^n$.

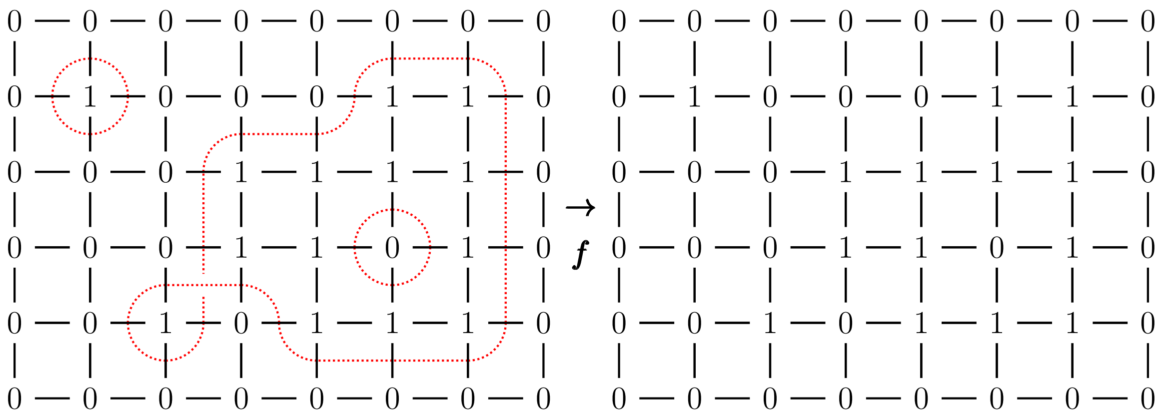


Figure 4: f applied to the left mosaic in Figure 2, resulting in a binary lattice

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 4. 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 knots (including 0 knots), label it 0. If the vertex is surrounded by an odd number of knots, label it 1. Removing the red dotted lines from the tiles gives the binary lattice.

Similarly, define the *preimage* of a set L under f to be

$$f^{-1}(L) = \{m \in \mathbb{M}^{(m,n)} | f(m) \in L\}.$$

We want to compute $s_{m,n}$ by computing $f^{-1}(\{\ell\})$ for each $\ell \in \mathbb{L}^{(m,n)}$, and then summing over all ℓ. We begin by finding a simple way to compute $f^{-1}(\ell)$ for a binary lattice ℓ by examining the structure of ℓ.

Definition 3.2. Let a *cell* be the portion of the binary lattice that an individual tile maps to, and let C be the set of unique cells.

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. Below is a diagram of all 2^4 cells with their indexes listed below.

$$\begin{array}{cccccccc}
\begin{matrix}0 & \text{—} & 0\\ | & & | \\ 0 & \text{—} & 0\end{matrix} &
\begin{matrix}0 & \text{—} & 1\\ | & & | \\ 0 & \text{—} & 0\end{matrix} &
\begin{matrix}1 & \text{—} & 0\\ | & & | \\ 0 & \text{—} & 0\end{matrix} &
\begin{matrix}1 & \text{—} & 1\\ | & & | \\ 0 & \text{—} & 0\end{matrix} &
\begin{matrix}0 & \text{—} & 0\\ | & & | \\ 0 & \text{—} & 1\end{matrix} &
\begin{matrix}0 & \text{—} & 1\\ | & & | \\ 0 & \text{—} & 1\end{matrix} &
\begin{matrix}1 & \text{—} & 0\\ | & & | \\ 0 & \text{—} & 1\end{matrix} &
\begin{matrix}1 & \text{—} & 1\\ | & & | \\ 0 & \text{—} & 1\end{matrix} \\
\mathbf{(00,\ 00)} & \mathbf{(00,\ 01)} & \mathbf{(00,\ 10)} & \mathbf{(00,\ 11)} & \mathbf{(01,\ 00)} & \mathbf{(01,\ 01)} & \mathbf{(01,\ 10)} & \mathbf{(01,\ 11)}
\end{array}$$

$$\begin{array}{cccccccc}
\begin{matrix}0 & \text{—} & 0\\ | & & | \\ 1 & \text{—} & 0\end{matrix} &
\begin{matrix}0 & \text{—} & 1\\ | & & | \\ 1 & \text{—} & 0\end{matrix} &
\begin{matrix}1 & \text{—} & 0\\ | & & | \\ 1 & \text{—} & 0\end{matrix} &
\begin{matrix}1 & \text{—} & 1\\ | & & | \\ 1 & \text{—} & 0\end{matrix} &
\begin{matrix}0 & \text{—} & 0\\ | & & | \\ 1 & \text{—} & 1\end{matrix} &
\begin{matrix}0 & \text{—} & 1\\ | & & | \\ 1 & \text{—} & 1\end{matrix} &
\begin{matrix}1 & \text{—} & 0\\ | & & | \\ 1 & \text{—} & 1\end{matrix} &
\begin{matrix}1 & \text{—} & 1\\ | & & | \\ 1 & \text{—} & 1\end{matrix} \\
\mathbf{(10,\ 00)} & \mathbf{(10,\ 01)} & \mathbf{(10,\ 10)} & \mathbf{(10,\ 11)} & \mathbf{(11,\ 00)} & \mathbf{(11,\ 01)} & \mathbf{(11,\ 10)} & \mathbf{(11,\ 11)}
\end{array}$$

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 knot in certain ways. For example, $u_{(00,01)} = 1$, as cell $(00,01)$ can only be formed by a mosaic with T_3 in that location. We can see that $u_{(01,10)} = 4$, as cell $(01,10)$ can be from the T_7, T_8, T_9, or T_{10} tiles. Finally, we have $u_{(00,00)} = 11$, as any tile can fail to contribute to forming a knot. 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**	$\mathbf{u_{(i,j)}}$
$(00,00)$	$\mathbb{T}$	11	$(10,00)$	T_1	1
$(00,01)$	T_3	1	$(10,01)$	T_7, T_8, T_9, T_{10}	4
$(00,10)$	T_4	1	$(10,10)$	T_6	1
$(00,11)$	T_5	1	$(10,11)$	T_2	1
$(01,00)$	T_2	1	$(11,00)$	T_5	1
$(01,01)$	T_6	1	$(11,01)$	T_4	1
$(01,10)$	T_7, T_8, T_9, T_{10}	4	$(11,10)$	T_1	1
$(01,11)$	T_1	1	$(11,11)$	$\mathbb{T}$	11

Table 1: Preimages of each unique cell

However, for some binary lattice ℓ the quantity

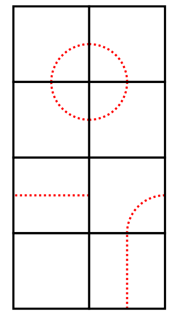
$$U(\ell) := \prod_{\text{Cell } (i,j) \in \ell} u_{(i,j)} \tag{1}$$

is not necessarily equal to $\left|f^{-1}\left(\{\ell\}\right)\right|$, as $U(\ell)$ does not *just* count the number of mosaics that map to ℓ under f.

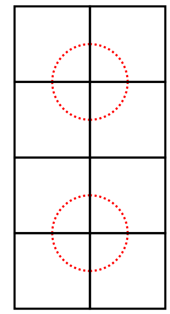
Example 3.1. Consider the following binary lattices for $s_{4,2}$.

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

$U(\ell_1) = 11^4$ $\quad U(\ell_2) = 11^4$ $\quad U(\ell_3) = 11^0$



Mosaic 1



Mosaic 2

$U(\ell_1)$ 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 11 possible cells, though 1 of those 11^4 permutations contains a new knot.

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

Example 3.2. From Example 3.1, ℓ_1 specifies the knot in the top 2 rows of Mosaic 1 and Mosaic 2, but not the knot in the bottom 2 rows of Mosaic 2. ℓ_3 specifies both knots in Mosaic 2.

Definition 3.4. Let $K : \mathbb{L}^{(m,n)} \to \mathbb{N}$ be the number of knots specified in ℓ.

Example 3.3. From Example 3.1, $K(\ell_1) = 1$, $K(\ell_2) = 1$, and $K(\ell_3) = 2$.

Definition 3.5. For a binary lattice ℓ, let $\mathbb{U}(\ell)$ be the set of binary lattices whose preimage mosaics under f are counted by $U(\ell)$. 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 specifying new knots in the mosaics in the preimage of ℓ while leaving all knots specified by ℓ unchanged.

Example 3.4. From Example 3.1, $\mathbb{U}(\ell_1) = \{\ell_1, \ell_3\}$, $\mathbb{U}(\ell_2) = \{\ell_2, \ell_3\}$, and $\mathbb{U}(\ell_3) = \{\ell_3\}$.

From the definition of $\mathbb{U}$, we have

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

Therefore $s_{m,n} \neq \sum_{\ell \in \mathbb{L}^{(m,n)}} U(\ell)$, as the sum overcounts mosaics for all ℓ that have $\mathbb{U}(\ell) \neq \{\ell\}$.

Definition 3.6. Let $\ell^* \in \mathbb{L}^{(m,n)}$ be the binary lattice made up of all $(00, 00)$ cells.

ℓ^* specifies 0 knots and has $U(\ell^*) = 11^{mn} = \left|\mathbb{M}^{(m,n)}\right|$, which clearly overcounts $s_{m,n}$. Also note that $\mathbb{U}(\ell^*) = \mathbb{L}^{(m,n)}$.

Proposition 3.1. *By regrouping terms we have*

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

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

If $\ell = \ell^*$, we have that $K(\ell^*) = 0$, 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 knot $s_{m,n}$ has*

$$s_{m,n} = \sum_{\ell\in\mathbb{L}^{(m,n)}} (-1)^{K(\ell)} U(\ell). \quad (4)$$

Proof. By the binomial theorem,

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

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

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

□

It is important to note here that a knot that contains tiles T_7 or T_8 can appear to be two separate knots which we consider as 1 knot.

Example 3.5. Figure 5 shows a mosaic that appears to have 2 knots, but we only consider as 1 knot. A rule of thumb that can be followed is if knots that contain T_7 or T_8 tiles can be replaced by T_9 or T_{10} tiles to form a single knot, then the original "knots" we consider as 1 knot.

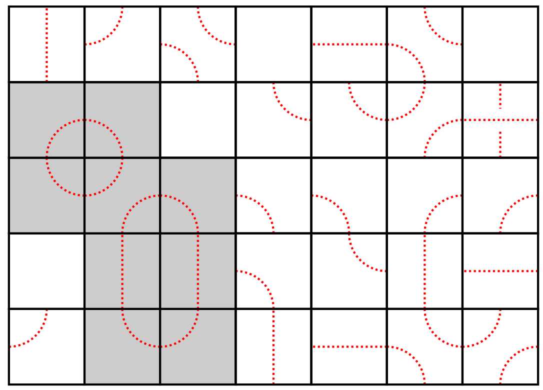


Figure 5: A messy knot mosaic with 1 knot

4 A Cell-Level Identity

Though Equation 4 does compute $s_{m,n}$, computing the number of knots specified in a binary lattice $K(\ell)$ requires examining the entire, global structure of ℓ. It will be more efficient to recover the $(-1)^{K(\ell)}$ term at the cell level. The idea is to add a coefficient $p_{(i,j)}$ to each value of $u_{(i,j)}$ so that the $(-1)^{K(\ell)}$ term is recovered.

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

$$\prod_{\textit{Cell } (i,j)\in\mathcal{K}} p_{(i,j)} = -1, \tag{5}$$

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

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

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

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 knot, and so must be positive.

For the remaining values of $p_{(i,j)}$, we examine the cells that map to the smallest knot, shown in Figure 3.

$$\begin{array}{ccccc} 0 & — & 0 & — & 0 \\ | & & | & & | \\ 0 & — & 1 & — & 0 \\ | & & | & & | \\ 0 & — & 0 & — & 0 \end{array} \qquad\qquad \begin{array}{ccccc} 1 & — & 1 & — & 1 \\ | & & | & & | \\ 1 & — & 0 & — & 1 \\ | & & | & & | \\ 1 & — & 1 & — & 1 \end{array}$$

Figure 6: Portions of binary lattices associated with the smallest knot

Condition 4.1 amounts to the following two equations

$$\begin{aligned} p_{(00,01)}p_{(00,10)}p_{(01,00)}p_{(10,00)} &= -1 \\ p_{(11,10)}p_{(11,01)}p_{(10,11)}p_{(01,11)} &= -1, \end{aligned} \tag{6}$$

one for each portion of a binary lattice in Figure 6. We refer to the equations above as *constraints*, as they constrain the possible assignments of $p_{(i,j)}$. Let the constraints in Equation 6 be numbered 1 and 2.

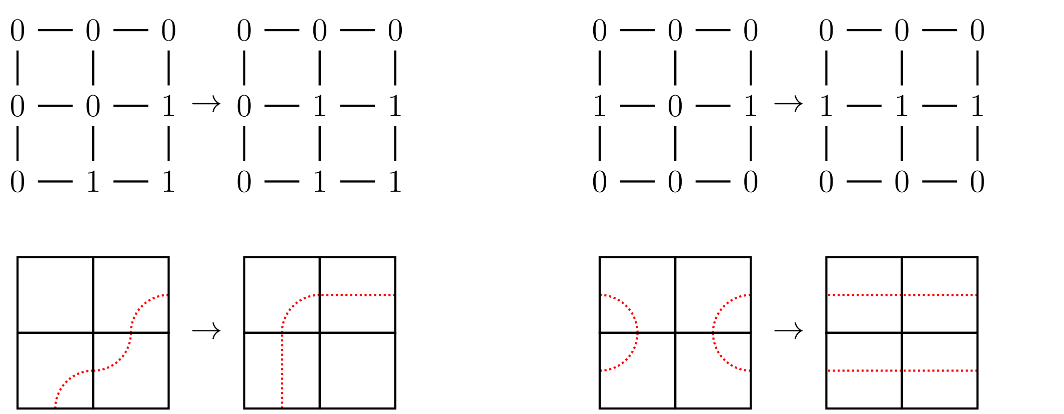
To define the remaining constraints, we next examine how specified knots arise in binary lattices.

Definition 4.1. 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, horizontal, and diagonal moves so that all of vertices in the path are in X.

In a binary lattice, a specified knot 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 4, where the 3 knots correspond with 3 edge-and-corner connected regions, excluding the region that is connected to the boundary.

The idea is, starting 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 7, 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*.



(a) Type 1 bit flip and associated tiles

(b) Type 2 bit flip and associated tiles

Figure 7: 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 7a represents the constraint

$$p_{(00,00)}p_{(00,01)}p_{(00,01)}p_{(01,11)} = p_{(00,01)}p_{(00,11)}p_{(01,01)}p_{(01,11)}. \tag{7}$$

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 7b represents the constraint[2]

$$p_{(00,10)}p_{(00,01)}p_{(10,00)}p_{(01,00)} = -p_{(00,11)}p_{(00,11)}p_{(11,00)}p_{(11,00)}. \tag{8}$$

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.

□

If we let $v_{(i,j)} := p_{(i,j)}u_{(i,j)}$, and similarly define

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

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

Cell (i, j)	$\mathbf{u_{(i,j)}}$	$\mathbf{p_{(i,j)}}$	$\mathbf{v_{(i,j)}}$	**Cell (i, j)**	$\mathbf{u_{(i,j)}}$	$\mathbf{p_{(i,j)}}$	$\mathbf{v_{(i,j)}}$
$(00,00)$	11	1	11	$(10,00)$	1	-1	-1
$(00,01)$	1	1	1	$(10,01)$	4	1	4
$(00,10)$	1	1	1	$(10,10)$	1	1	1
$(00,11)$	1	1	1	$(10,11)$	1	1	1
$(01,00)$	1	1	1	$(11,00)$	1	1	1
$(01,01)$	1	1	1	$(11,01)$	1	1	1
$(01,10)$	4	-1	-4	$(11,10)$	1	-1	-1
$(01,11)$	1	1	1	$(11,11)$	11	1	11

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

We can then simplify Equation 4 to

$$s_{m,n} = \sum_{\ell\in\mathbb{L}^{(m,n)}} (-1)^{K(\ell)}U(\ell) = \sum_{\ell\in\mathbb{L}^{(m,n)}} V(\ell) = \sum_{\ell\in\mathbb{L}^{(m,n)}} \prod_{\text{Cell } (i,j)\in\ell} v_{(i,j)}. \tag{9}$$

Following other work in mosaic systems, $s_{m,n}$ can be calculated more efficiently than in Equation 9 using the state matrix recursion introduced in [9].

[2]In Figure 7b the transformation corresponds with *either* two distinct knot joining into one knot *or* one knot splitting into two distinct knot. In either case, we want the sign of the product to change.

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 8 is a binary sub-lattice of size $(2, 4)$ with index $(011, 100)$.

$$\begin{array}{ccccccccc}
0 & — & 1 & — & 0 & — & 0 & — & 0 \\
| & & | & & | & & | & & | \\
0 & — & 0 & — & 1 & — & 0 & — & 0 \\
| & & | & & | & & | & & | \\
0 & — & 0 & — & 1 & — & 1 & — & 0
\end{array}$$

Figure 8: 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 $(0\ldots0, 0\ldots0)$ 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 $A^{(1,2)} = \begin{bmatrix} 11^2 & -1 \\ 1 & 1 \end{bmatrix}$. 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*

$$A^{(1,k+1)} = \begin{bmatrix} 11A_{[0,0]} & 11A_{[0,1]} & 11^{-1}A_{[0,0]} & A_{[0,1]} \\ 11A_{[1,0]} & 11A_{[1,1]} & -4A_{[1,0]} & A_{[1,1]} \\ 11^{-1}A_{[0,0]} & 4A_{[0,1]} & 11^{-1}A_{[0,0]} & A_{[0,1]} \\ A_{[1,0]} & A_{[1,1]} & -A_{[1,0]} & 11A_{[1,1]} \end{bmatrix}.$$

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 9.

$$\begin{array}{ccccc} 0 & — & 0 & — & 0 \\ | & & | & & | \\ 0 & — & 0 & — & 0 \end{array} \qquad \begin{array}{ccccc} 0 & — & 1 & — & 0 \\ | & & | & & | \\ 0 & — & 0 & — & 0 \end{array}$$

$$\begin{array}{ccccc} 0 & — & 0 & — & 0 \\ | & & | & & | \\ 0 & — & 1 & — & 0 \end{array} \qquad \begin{array}{ccccc} 0 & — & 1 & — & 0 \\ | & & | & & | \\ 0 & — & 1 & — & 0 \end{array}$$

Above gives

$$\begin{bmatrix} v_{(00,00)}v_{(00,00)} & v_{(01,00)}v_{(10,00)} \\ v_{(00,01)}v_{(00,10)} & v_{(01,01)}v_{(10,10)} \end{bmatrix} = \begin{bmatrix} 11^2 & -1 \\ 1 & 1 \end{bmatrix}.$$

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 $(0,\dots,0\dots)$, 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 $(0,\dots,1\dots)$, and so on.

We then append one of each of the four cells in Figure 9 to the left of every size $(1,k)$ binary sub-lattices, such that the former left boundary 0s are replaced.

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

(00, 00) **(00, 01)**

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

(01, 00) **(01, 01)**

Figure 9: Appending cells

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

$$\begin{array}{ccc} 0 & — & 0 \\ | & & | \\ 0 & — & 1 \end{array} + \begin{array}{ccccccc} 0 & — & 1 & — & 0 & — & 0 \\ | & & | & & | & & | \\ 0 & — & 0 & — & 1 & — & 0 \end{array} \rightarrow \begin{array}{ccccccccc} 0 & — & 0 & — & 1 & — & 0 & — & 0 \\ | & & | & & | & & | & & | \\ 0 & — & 1 & — & 0 & — & 1 & — & 0 \end{array}$$

This creates the size $(1, k+1)$ binary sub-lattice $(101, 010)$. Notice this operation only changes the identity of the two left-most cells, namely changing the $(00, 01)$ cell into cells $(01, 00)$ and $(10, 01)$.

By construction, as every binary sub-lattice counted in the block matrix $A_{[0,1]}$ has an index of the form $(0\ldots, 1\ldots)$, the entries in $A_{[0,1]}$ are all divisible by $v_{00,01}$. Therefore, the size $(1, k+1)$ binary sub-lattices with index of the form $(10\ldots, 01\ldots)$ are counted by $(v_{01,00}v_{10,01}/v_{00,01})A_{[0,1]}$.

Performing the appending operation each of the four cells from Figure 9 to each other gives 16 possible pairs, which we diagram below such that it is consistent with the indexing for $A^{(}1, k)$.

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

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

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

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

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

$$
\begin{array}{llll}
\frac{v_{(00,00)}v_{(00,00)}}{v_{(00,00)}} = 11 & \frac{v_{(00,00)}v_{(00,01)}}{v_{(00,01)}} = 11 & \frac{v_{(00,01)}v_{(00,10)}}{v_{(00,00)}} = 11^{-1} & \frac{v_{(00,01)}v_{(00,11)}}{v_{(00,01)}} = 1 \\
\frac{v_{(00,00)}v_{(01,00)}}{v_{(01,00)}} = 11 & \frac{v_{(00,00)}v_{(01,01)}}{v_{(01,01)}} = 11 & \frac{v_{(00,01)}v_{(01,10)}}{v_{(01,00)}} = -4 & \frac{v_{(00,01)}v_{(01,11)}}{v_{(01,01)}} = 1 \\
\frac{v_{(01,00)}v_{(10,00)}}{v_{(00,00)}} = -11^{-1} & \frac{v_{(01,00)}v_{(10,01)}}{v_{(00,01)}} = 4 & \frac{v_{(01,01)}v_{(10,10)}}{v_{(00,00)}} = 11^{-1} & \frac{v_{(01,01)}v_{(10,11)}}{v_{(00,01)}} = 1 \\
\frac{v_{(01,00)}v_{(11,00)}}{v_{(01,00)}} = 1 & \frac{v_{(01,00)}v_{(11,01)}}{v_{(01,01)}} = 1 & \frac{v_{(01,01)}v_{(11,10)}}{v_{(01,00)}} = -1 & \frac{v_{(01,01)}v_{(11,11)}}{v_{(01,01)}} = 11
\end{array}
$$

We can then determine that

$$
A^{(1,k+1)} = \begin{bmatrix} 11A_{[0,0]} & 11A_{[0,1]} & 11^{-1}A_{[0,0]} & A_{[0,1]} \\ 11A_{[1,0]} & 11A_{[1,1]} & -4A_{[1,0]} & A_{[1,1]} \\ 11^{-1}A_{[0,0]} & 4A_{[0,1]} & 11^{-1}A_{[0,0]} & A_{[0,1]} \\ A_{[1,0]} & A_{[1,1]} & -A_{[1,0]} & 11A_{[1,1]} \end{bmatrix}.
$$

□

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, $A^{(k,q)}_{i,j}$

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

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 10 adjoins a size $(2,3)$ binary sub-lattice with index $(10,01)$ to a size $(1,3)$ binary sub-lattice with index $(01,11)$.

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

Figure 10: Adjoins a size $(1,3)$ binary sub-lattice to a size $(2,3)$ sub-lattice

This operation amounts to

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

which gives

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

□

This state matrix recursion method, as explored in [6, 8, 1, 5, 7] is extremely versatile in various combinatorial questions. We highlight the most obvious extensions

6 Extensions

Hong and Oh [2] study the mosaic system with the tile set $\mathbb{T}^* = \{T_0, \ldots, T_7\}$. This tile set constructs shapes we call *polygons*[3]. If we let $p_{m,n}$ be the number of polygon mosaics of size

[3]Polygons are more commonly called ”self-avoiding polygons” in the literature to emphasize their relationship with self-avoiding walks.

(m, n), Hong and Oh showed the following results[4].

Theorem 6.1 ([2])**.** *The number of polygon mosaics of size* (m, n) $p_{m,n}$ *for* $m, n \geq 2$ *has*

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

The array $p_{n,m}$ is A181245 on the OEIS [4, OEIS]. Though not stated in Hong and Oh [2], $p_{m,n}$ is exactly enumerated by Theorem 1.1 by replacing the 4 in the definition of O_{k+1} with a 0.

We can achieve similar enumerations results to Theorem 2.1 when considering messy polygon mosaics with $\mathbb{T}^*$, simply by assigning $v_{10,01} = v_{01,10} = 0$.

7 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.

[4]The authors did not consider the mosaic containing all T_0 tiles a polygon mosaic, and so define $p_{m,n}$ as one less than what we define.

[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.