# Exact Enumeration of Polygon Mosaics

### Jack Hanke

Richard Schank

Northwestern University

### Michael Maltenfort

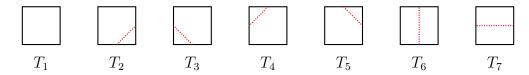
Northwestern University

#### Abstract

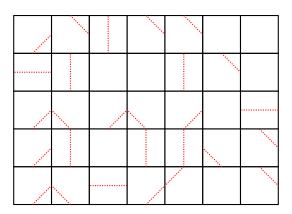
Hong and Oh calculated upper and lower bounds on the number of polygon mosaics for 7 distinct tiles that together model multiple ring polymers in physics. We exactly enumerate these polygon mosaics. The method we introduce generalizes to various other tile sets. We also introduce and enumerate a variation on polygon mosaics called messy polygon mosaics for various tile sets.

### 1 Introduction

Consider the following set of 7 symbols labelled  $\{T_1, \ldots T_7\}$  composed of unit squares and dotted lines connecting pairs of sides at their midpoints.



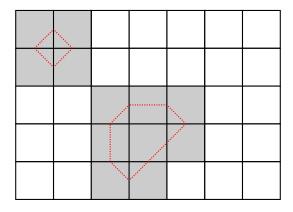
Call these symbols *tiles*. A mosaic of size (m, n) is an  $m \times n$  matrix of these tiles. For example, below is a mosaic of size (5,7).



Consider a edge shared between two tiles in the above figure. 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. We define a *polygon mosaic* to be a mosaic that has all unit edges having 0 or 2 dotted lines drawn from their midpoint.

We call these polygon mosaics because, other than the mosaic consisting of all  $T_1$  tiles, the dotted lines form shapes we call  $polygons^1$ .

**Example 1.1.** Below is an example of a polygon mosaic of size (5,7) that contains 2 polygons, with the squares that make them up hilighted in gray.



Note that in large enough mosaics, it is possible for larger polygons to surround smaller polygons.

Let  $p_{m,n}$  be the number of polygon mosaics of size (m,n). First notice that if either m or n is 1, one cannot construct a polygon mosaic, so

$$p_{m,1} = p_{1,n} = 0.$$

Hong and Oh [1, Hong2018] gave the following upper and lower bounds<sup>2</sup> for  $m, n \ge 2$ .

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

In Theorem 2, we provide an exact expression for  $p_{m,n}$  for  $m, n \geq 2$ . We also introduce messy polygon mosaics—a variant of polygon mosaics—and enumerate them using a similar strategy.

First we define  $A(m) \in \mathbb{Z}^{2^{m-1} \times 2^{m-1}}$ , which we use to enumerate  $p_{m,n}$ .

**Definition 1.** First define  $A(2) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ . We recursively define A(k+1) given A(k). Begin by writing  $A(k) = \begin{bmatrix} A_{0,0} & A_{0,1} \\ A_{1,0} & A_{1,1} \end{bmatrix}$ , where the block matrices  $A_{i,j}$  are square block matrices of size  $2^{k-2} \times 2^{k-2}$ . We then have

 $<sup>^{1}</sup>$ Polygons are more commonly called "self-avoiding polygons" in the literature to emphasize their relationship with self-avoiding walks.

<sup>&</sup>lt;sup>2</sup>The authors did not consider the mosaic containing all  $T_1$  tiles a polygon mosaic, and so define  $p_{m,n}$  as one less than what we define.

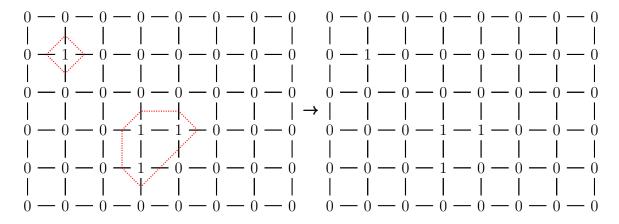
$$A(k+1) = \begin{bmatrix} A_{0,0} & A_{0,0} & A_{0,1} & A_{0,1} \\ A_{0,0} & A_{0,0} & 0A_{0,1} & A_{0,1} \\ A_{1,0} & 0A_{1,0} & A_{1,1} & A_{1,1} \\ A_{1,0} & A_{1,0} & A_{1,1} & A_{1,1} \end{bmatrix},$$

Construct A(m) by starting with k=2 and recursing until k=m.

**Theorem 2.** The number of polygon mosaics  $p_{m,n}$  is the (0,0)-th entry of  $A(m)^n$ .

### 2 Proof of Theorem 2

*Proof.* For a given mosaic, label the vertices of the tiles as follows. If the vertex is surrounded by an even number of polygons, label it 0. If the vertex is surrounded by an odd number of polygons, label it 1. To make a *vertex labeling* we also remove the dotted lines from all tiles in the mosaic. Using the mosaic from Example 1.1, we show both the labeling of the vertices, and then the removal of the dotted lines.



The critical point is this: even with the dotted lines removed, the vertex labeling uniquely identifies the polygon mosaic. This is true even if there are polygons surrounding other polygons. This is because the vertex labeling of an individual cell, which we will call a *cell-labeling*, uniquely corresponds with a cell  $T_i$ . This is shown below.

We can then enumerate  $p_{m,n}$  by enumerating the number of vertex labelings that correspond with a valid polygon mosaic.

Consider the collection of vertex labelings for a  $m \times n$  grid of cells. To correspond with a polygon mosaic, each boundary vertex is necessarily labelled 0, and each interior vertex is labelled 0 or 1. Additionally, of these  $2^{(m-1)(n-1)}$  labelings, the labelings that correspond with valid polygon mosaics are ones that do not contain the following two cell-labelings.

These two cell-labelings aren't associated with any of the cells  $T_1, \ldots, T_7$ , and so cannot correspond with a polygon mosaic. We seek a recursive solution to enumerate the number of vertex labelings for a  $m \times n$  grid with these conditions: the labeling has a boundary of all 0's and does not contain the above cell-labelings.

Our solution accomplishes this by building  $m \times n$  vertex labelings for fixed m using vertex labelings of  $m \times 1$  columns of cells. These columns have vertex labelings such that the top and bottom edge all labelled 0. One of these columns for m=3 is below.

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

It is useful to define an index for vertex labelings of these  $m \times 1$  columns. To do this, consider reading a column of vertex labelings from bottom to top, ignoring the first and last 0's. If we interpret these sequences for the left and right column as binary numbers  $b_{left}$  and  $b_{right}$ , the index in base ten is the pair  $(b_{left}, b_{right})$ . For example, for the above m = 3 column, we have sequences 00 for the left column and 10 for the right column, so the index is (0,2).

Next consider a column with index  $(0, b_1)$  for some  $b_1 \in [0, 2^{m-1} - 1]$ . For reasons we will see later, let's call this the *starting column*. Now consider appending a column with index  $(b_1, b_2)$  for some  $b_2 \in [0, 2^{m-1} - 1]$ . For example, below is a diagram for appending our starting column (0, 2) with column (2, 3).

Notice that in the above example we have created a vertex labeling for an  $m \times 2$  grid of cells, in which the left-most vertex column labels are all 0 (left index of 0). Also note that

we did not create either illegal cell-labelings with column (2,3). However, we could append the column (2,1), which would create an illegal cell-labeling.

Finally, if we further appended a column with label  $(b_2, 0)$ , we would have created a vertex labeling with a boundary of all 0's. For this reason, let's call a column with index (b,0) for  $b \in [0,2^{m-1}-1]$  an ending column. This vertex labeling would correspond with 1 polygon mosaic if the middle column had index (2,3), but not if the middle column had index (2,1).

This motivates the creation of a matrix A(m) to every column index (i, j), where  $A(m)_{i,j} = 1$  if the column labeling corresponds with a legal polygon mosaic, and 0 otherwise. For example, the A(3) matrix corresponds with the following labeled columns.

			0 — 0 		
		$ \begin{array}{c c} 0 & - & 0 \\  & &   \\ 1 & - & 0 \\  & &   \\ 0 & - & 1 \\  & &   \\ 0 & - & 0 \end{array} $	$ \begin{array}{c c} 0 & - & 0 \\  & &   \\ 1 & - & 1 \\  & &   \\ 0 & - & 1 \\  & &   \\ 0 & - & 0 \end{array} $		$ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} $
$ \begin{array}{c c} 0 & - & 0 \\  & &   \\ 0 & - & 0 \\  & &   \\ 1 & - & 0 \\  & &   \\ 0 & - & 0 \end{array} $	 1 — 0 		$ \begin{array}{c c} 0 & - & 0 \\  & &   \\ 0 & - & 1 \\  & &   \\ 1 & - & 1 \\  & &   \\ 0 & - & 0 \end{array} $	7	$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$ $A(3)$
$ \begin{array}{c c} 0 & - & 0 \\  & &   \\ 1 & - & 0 \\  & &   \\ 1 & - & 0 \\  & &   \\ 0 & - & 0 \end{array} $			$ \begin{array}{c c} 0 & - & 0 \\  & &   \\ 1 & - & 1 \\  & &   \\ 1 & - & 1 \\  & &   \\ 0 & - & 0 \end{array} $		

A(m) has the property that the 0-th row represets all starting columns, and the 0-th column represents all ending columns. Even more importantly, notice that  $A(m)_{i,j}^2$  represents the number of  $m \times 2$  grids with left-most index i and right-most index j that correspond

with a legal polygon mosaic. In general,  $(A(m)^n)_{i,j}$  represents this quantity for an  $m \times n$  grid of cells, and so if we know A(m) for some m, then  $(A(m)^n)_{0,0} = p_{m,n}$ .

The final component of the proof is constructing A(m) for any m. Begin by calculating A(2) by identifying the number of legal polygon mosaics that correspond with each vertex coloring index  $\{(0,0),(0,1),(1,0),(1,1)\}$ , like so.

Next consider an arbitrary value  $A(k)_{i,j}$  for any  $k \geq 2$ . This value is 1 if the  $k \times 1$  column with index (i,j) can be part of a polygon mosaic, and 0 otherwise. We can determine that specific values of A(k+1) are multiples of  $A(k)_{i,j}$  by considering the following operation on an arbitrary column with index (i,j). Copy the column four times and replace the top two 0's of each column with the bottom row of labels from one of the four cell-labelings below.

For example, for the m=2 column with index (0,1), this operation looks like the following.

This operation results in 4 new columns that are represented in A(k+1). In our example, specifically we get the following.

$$\begin{bmatrix} A(3)_{0,2} & A(3)_{0,3} \\ A(3)_{1,2} & A(3)_{1,3} \end{bmatrix} = \begin{bmatrix} 1A(2)_{i,j} & 1A(2)_{i,j} \\ 0A(2)_{i,j} & 1A(2)_{i,j} \end{bmatrix},$$

Critically, this transformation *only* changes the identity of the top two tiles. This implies that the same value coefficients computed by comparing A(2) and A(3) can be used for any  $m \times 1$  column, as long as both column indices (i, j) are congruent mod 2. Furthermore, if one writes A(k) as the block matrix

$$A(k) = \begin{bmatrix} A_{0,0} & A_{0,1} \\ A_{1,0} & A_{1,1} \end{bmatrix},$$

where  $A_{\hat{i},\hat{j}} \in \mathbb{R}^{2^{k-2} \times 2^{k-2}}$ , then all column's represented in  $A_{\hat{i},\hat{j}}$  have indices  $(i,j) \equiv (\hat{i},\hat{j}) \mod 2$ . This allows us to write that in general, if  $A(k) = \begin{bmatrix} A_{0,0} & A_{0,1} \\ A_{1,0} & A_{1,1} \end{bmatrix}$ , then

$$A(k+1) = \begin{bmatrix} V_{0,0}A_{0,0} & V_{0,1}A_{0,0} & V_{0,2}A_{0,1} & V_{0,3}A_{0,1} \\ V_{1,0}A_{0,0} & V_{1,1}A_{0,0} & V_{1,2}A_{0,1} & V_{1,3}A_{0,1} \\ V_{2,0}A_{1,0} & V_{2,1}A_{1,0} & V_{2,2}A_{1,1} & V_{2,3}A_{1,1} \\ V_{3,0}A_{1,0} & V_{3,1}A_{1,0} & V_{3,2}A_{1,1} & V_{3,3}A_{1,1} \end{bmatrix}.$$

where  $V \in \mathbb{R}^{4\times 4}$  can be found after directly computing A(2) and A(3), then solving the following equation.

$$\begin{bmatrix} A(3)_{0,0} & A(3)_{0,1} & A(3)_{0,2} & A(3)_{0,3} \\ A(3)_{1,0} & A(3)_{1,1} & A(3)_{1,2} & A(3)_{1,3} \\ A(3)_{2,0} & A(3)_{2,1} & A(3)_{2,2} & A(3)_{2,3} \\ A(3)_{3,0} & A(3)_{3,1} & A(3)_{3,2} & A(3)_{3,3} \end{bmatrix} = \begin{bmatrix} V_{0,0}A(2)_{0,0} & V_{0,1}A(2)_{0,0} & V_{0,2}A(2)_{0,1} & V_{0,3}A(2)_{0,1} \\ V_{1,0}A(2)_{0,0} & V_{1,1}A(2)_{0,0} & V_{1,2}A(2)_{0,1} & V_{1,3}A(2)_{0,1} \\ V_{2,0}A(2)_{1,0} & V_{2,1}A(2)_{1,0} & V_{2,2}A(2)_{1,1} & V_{2,3}A(2)_{1,1} \\ V_{3,0}A(2)_{1,0} & V_{3,1}A(2)_{1,0} & V_{3,2}A(2)_{1,1} & V_{3,3}A(2)_{1,1} \end{bmatrix}$$

This is easily solved to give

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

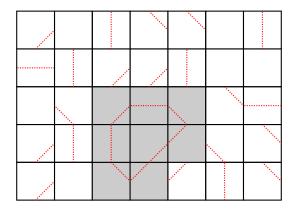
which completes the proof.

The method detailed in Theorem 2 generalizes to other tile sets, which we tabularize in Section 6 without proof. Interestingly, the method not only generalizes to other tile sets, but can also be augmented to enumerate the more complicated "messy" polygon mosaics.

#### Messy Polygon Mosaics 3

A messy polygon mosaic is a mosaic that contains at least one polygon, with no restriction on other shared edges.

**Example 3.1.** Below is a messy polygon mosaic of size (5, 7) that contains 1 polygon, with the squares that make it up hilighted in gray.



It turns out that it is easier to enumerate the number of messy mosaics that do not contain a polygon. Therefore, let  $t_{m,n}$  be the number of mosaics that do not contain a polygon. Also from the fact that the smallest polygon is



we have that  $t_{n,1} = 7^n$ , and  $t_{2,2} = 7^4 - 1$ . It turns out that using similar techniques to Theorem 2 we can exactly enumerate messy mosaics. We define  $A(m) \in \mathbb{Z}^{2^{m-1} \times 2^{m-1}}$ , which we use to enumerate  $t_{m,n}$ .

**Definition 3.** First define  $A(2) = \begin{bmatrix} 7^2 & 1 \\ -1 & 1 \end{bmatrix}$ . We recursively define  $A(k+1) \in \mathbb{Z}^{2^{n-1} \times 2^{n-1}}$  given A(k). Begin by writing A(k) = 1 $\begin{bmatrix} A_{0,0} & A_{0,1} \\ A_{1,0} & A_{1,1} \end{bmatrix}$ , where the block matrices  $A_{i,j}$  are square block matrices of size  $2^{k-2} \times 2^{k-2}$ . We then have

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

Construct A(m) by starting with k=2 and recursing until k=m.

**Theorem 4.** The number of messy mosaics that do not contain a polygon is the (0,0)-th entry of  $A(m)^n$ .

8

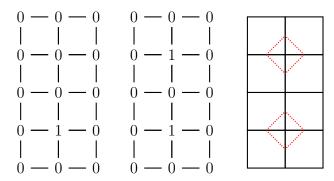
## 4 Proof of Theorem 4

*Proof.* We consider the same vertex labeling in the proof of Theorem 2, namely labeling a vertex by the number of polygons that contain it mod 2. However, there is not a bijection between messy polygon mosaics and vertex labelings as there is with polygon mosaics. This is because the following cell-labellings configurations are no longer uniquely determined.

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

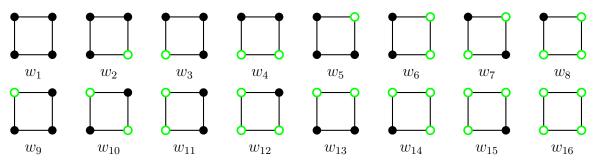
Additionally, the sub-grid vertex labelings shown below

are now ambiguous as to whether or not they represent a polygon. For example, both of the following vertex labelings could include the right-most messy polygon mosaic.



Critically, the number of polygons that appear in a given messy polygon mosaic is exactly the number of parity configurations that map to that mosaic. Therefore, if we are to count these mosaics, we need to account for this double-counting with the inclusion-exclusion principle.

To create an analogous result to Theorem 2 we need to find a weight assignment to the 16 parity configurations below that fulfill a set of conditions.



First  $w_7 = w_{10} = 0$ , as again these are impossible parity configurations for our tile set. Next notice that  $w_1 = w_{16} = b$ , as these cells do not indicate a specific tile. The remaining weights uniquely specify a tile, and so are equal to 1 or -1. But how do we find these assignments?

First notice that we want a weight assignment so that the parity configurations for a given polygon multiply to -1. This is due to the inclusion-exclusion principle. This means that if there are multiple polygons in a mosaic, then the product will be positive if there is an even number of polygons specified, and negative if there is an odd number specified.

Next note the following lemma.

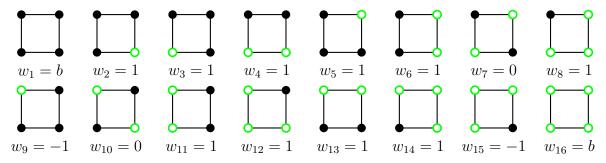
**Lemma 5.** One can construct all larger polygons from the smallest polygon using a finite set of transformations S.

This is because one can find  $w_1, \ldots, w_{16}$  so that the following two constraints hold:

Constraint 6. The weights associated with the smallest polygon multiply to -1, ie.  $w_2w_3w_5w_9 = -1$ .

Constraint 7. All transformations in S preserve the weight product of a changed polygon.

Constraint 6 and Constraint 7 amount to a series of constraints on the values of  $w_i$ . The derivation for these constraints can be found in the Appendix. Choosing a solution set from these constraints gives the following weights.



This immediately gives us a way to construct an analogous definition for our transition matrixes  $A(k) = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}$ . As the appending of a cell parity configuration can possibly change the number of admitted cells, we define  $v_1, \ldots, v_{16}$  for the associated change in the number of mosaics for the recursive definition of A(k+1). A simple way to obtain  $v_1, \ldots, v_{16}$  is to directly compute A(2) and A(3) and compare their values like so.

$$A(2) = \begin{bmatrix} b^2 & 1 \\ -1 & 1 \end{bmatrix}, A(3) = \begin{bmatrix} b^3 & b & b & 1 \\ -b & b & 0 & 1 \\ -b & 0 & b & 1 \\ -1 & 1 & 1 & b \end{bmatrix}.$$

This gives

$$A(k+1) = \begin{bmatrix} v_1 A_1 & v_2 A_2 & v_3 A_1 & v_4 A_2 \\ v_5 A_3 & v_6 A_4 & v_7 A_3 & v_8 A_4 \\ v_9 A_1 & v_{10} A_2 & v_{11} A_1 & v_{12} A_2 \\ v_{13} A_3 & v_{14} A_4 & v_{15} A_3 & v_{16} A_4 \end{bmatrix} = \begin{bmatrix} bA_1 & bA_2 & \frac{1}{b}A_1 & A_2 \\ bA_3 & bA_4 & 0A_3 & A_4 \\ -\frac{1}{b}A_1 & 0A_2 & \frac{1}{b}A_1 & A_2 \\ A_3 & A_4 & -A_3 & bA_4 \end{bmatrix}.$$

An analogous argument to Theorem 2 then gives the result.

## 5 Summary of Results

Below is a tabulated summary of the results, where the matrices under the "Polygon Mosaics" and "Messy Polygon Mosaics" column are A(2), W. One can identify the number of the number of (messy)

$$A(k+1) = W \cdot \begin{bmatrix} A_1(k) & A_2(k) \\ A_3(k) & A_4(k) \end{bmatrix}$$

$$\begin{bmatrix} v_1 A_1 & v_2 A_2 & v_3 A_1 & v_4 A_2 \\ v_5 A_3 & v_6 A_4 & v_7 A_3 & v_8 A_4 \\ v_9 A_1 & v_{10} A_2 & v_{11} A_1 & v_{12} A_2 \\ v_{13} A_3 & v_{14} A_4 & v_{15} A_3 & v_{16} A_4 \end{bmatrix}$$

Then the (0,0)-th entry of  $A(m)^n$  is  $p_{m,n}$  (polygon mosaics) and  $t_{n,m}$  (mosaics that do not contain polygons).

TODO finish explan of following table

Tile Set	Polygon Mosaics	Messy Polygon Mosaics				
	$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 7^2 & 1 \\ -1 & 1 \end{bmatrix}, \begin{bmatrix} 7 & 7 & \frac{1}{7} & 1 \\ 7 & 7 & 0 & 1 \\ -\frac{1}{7} & 0 & 1 & 1 \\ 1 & 1 & -1 & 7 \end{bmatrix}$				
		$\begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{7} & 0 & 1 & 1 \\ 1 & 1 & -1 & 7 \end{bmatrix}$				
	TODO	$\begin{bmatrix} 6^2 & 1 \\ -1 & 1 \end{bmatrix}, \begin{bmatrix} 6 & 6 & \frac{1}{6} & 1 \\ 6 & 6 & 0 & 1 \\ -\frac{1}{6} & 0 & 1 & 1 \\ 1 & 1 & -1 & 6 \end{bmatrix}$				
	1020	$\begin{bmatrix} -1 & 1 \end{bmatrix}, \begin{bmatrix} -\frac{1}{6} & 0 & 1 & 1 \\ 1 & 1 & -1 & 6 \end{bmatrix}$				
	$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ , TODO	TODO				
		1000				
	TODO	TODO				
	TODO	TODO				

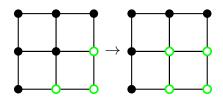
# References

- [1] 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.
- [2] OEIS Foundation Inc. *The On-Line Encyclopedia of Integer Sequences*. Published electronically at http://oeis.org.

## 6 Appendix

Flipping the parity of a single vertex in a parity configuration changes the 4 surrounding cells. This creates a constraint on a subset of  $w_1, \ldots, w_{16}$ .

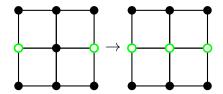
The flipping of parity of a single vertex can result in 2 distinct types of constraints. Let a constraint of *Type 1* be a parity flip that does not change the number of polygons in the parity configuration. For example, consider the following flip of the center vertex in the following portion of a parity configuration.



As this does not change the associated number of polygons in the larger parity configuration, we want this to preserve the sign of the weight product. This gives the following associated constraint.

$$sign(w_1w_2w_5w_9) = sign(w_2w_4w_6w_{16}).$$

Now let a constraint of *Type 2* be a parity flip that does change the number of polygons. For example, consider flipping the center vertex of the following portion of a parity configuration.



The above 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 switch. This corresponds with the following constraint.

$$sign(w_3w_2w_9w_5) = -sign(w_4w_4w_{13}w_{13}).$$

All Type 1 constraints are as follows. For the following set of equations, assume the equals sign (=) means *only* equal in sign.

$w_1 w_1 w_2 w_3 = w_2 w_3 w_6 w_{11}$	$w_1 w_1 w_2 w_4 = w_2 w_3 w_6 w_{12}$	$w_1 w_1 w_4 w_3 = w_2 w_3 w_8 w_{11}$
$w_1 w_1 w_4 w_4 = w_2 w_3 w_8 w_{12}$	$w_1 w_2 w_1 w_5 = w_2 w_4 w_5 w_{13}$	$w_1 w_2 w_1 w_6 = w_2 w_4 w_5 w_{14}$
$w_1 w_2 w_2 w_8 = w_2 w_4 w_6 w_{16}$	$w_1 w_2 w_4 w_8 = w_2 w_4 w_8 w_{16}$	$w_3w_1w_9w_1 = w_4w_3w_{13}w_9$
$w_3 w_1 w_{11} w_1 = w_4 w_3 w_{15} w_9$	$w_3 w_1 w_{12} w_3 = w_4 w_3 w_{16} w_{11}$	$w_3 w_1 w_{12} w_4 = w_4 w_3 w_{16} w_{12}$
$w_3 w_2 w_{12} w_8 = w_4 w_4 w_{16} w_{16}$	$w_1 w_6 w_1 w_5 = w_2 w_8 w_5 w_{13}$	$w_1 w_6 w_1 w_6 = w_2 w_8 w_5 w_{14}$
$w_1 w_6 w_2 w_8 = w_2 w_8 w_6 w_{16}$	$w_1 w_6 w_4 w_8 = w_2 w_8 w_8 w_{16}$	$w_3 w_6 w_{12} w_8 = w_4 w_8 w_{16} w_{16}$
$w_5 w_9 w_1 w_1 = w_6 w_{11} w_5 w_9$	$w_5w_{13}w_1w_1 = w_6w_{15}w_5w_9$	$w_5 w_{14} w_1 w_5 = w_6 w_{16} w_5 w_{13}$
$w_5 w_{14} w_1 w_6 = w_6 w_{16} w_5 w_{14}$	$w_5 w_{14} w_2 w_8 = w_6 w_{16} w_6 w_{16}$	$w_5 w_{14} w_4 w_8 = w_6 w_{16} w_8 w_{16}$
$w_{11}w_1w_9w_1 = w_{12}w_3w_{13}w_9$	$w_{11}w_1w_{11}w_1 = w_{12}w_3w_{15}w_9$	$w_{11}w_1w_{12}w_3 = w_{12}w_3w_{16}w_{11}$
$w_{11}w_1w_{12}w_4 = w_{12}w_3w_{16}w_{12}$	$w_{11}w_2w_{12}w_8 = w_{12}w_4w_{16}w_{16}$	$w_{11}w_6w_{12}w_8 = w_{12}w_8w_{16}w_{16}$
$w_{13}w_9w_1w_1 = w_{14}w_{11}w_5w_9$	$w_{15}w_9w_9w_1 = w_{16}w_{11}w_{13}w_9$	$w_{15}w_9w_{11}w_1 = w_{16}w_{11}w_{15}w_9$
$w_{15}w_9w_{12}w_3 = w_{16}w_{11}w_{16}w_{11}$	$w_{15}w_9w_{12}w_4 = w_{16}w_{11}w_{16}w_{12}$	$w_{13}w_{13}w_1w_1 = w_{14}w_{15}w_5w_9$
$w_{13}w_{14}w_1w_5 = w_{14}w_{16}w_5w_{13}$	$w_{13}w_{14}w_1w_6 = w_{14}w_{16}w_5w_{14}$	$w_{13}w_{14}w_2w_8 = w_{14}w_{16}w_6w_{16}$
$w_{13}w_{14}w_4w_8 = w_{14}w_{16}w_8w_{16}$	$w_{15}w_{13}w_9w_1 = w_{16}w_{15}w_{13}w_9$	$w_{15}w_{13}w_{11}w_1 = w_{16}w_{15}w_{15}w_9$
$w_{15}w_{13}w_{12}w_3 = w_{16}w_{15}w_{16}w_{11}$	$w_{15}w_{13}w_{12}w_4 = w_{16}w_{15}w_{16}w_{12}$	$w_{15}w_{14}w_9w_5 = w_{16}w_{16}w_{13}w_{13}$
$w_{15}w_{14}w_9w_6 = w_{16}w_{16}w_{13}w_{14}$	$w_{15}w_{14}w_{11}w_5 = w_{16}w_{16}w_{15}w_{13}$	$w_{15}w_{14}w_{11}w_6 = w_{16}w_{16}w_{15}w_{14}$

Similarly, all Type 2 constraints are as follows. Again, for the following set of equations, assume the equals sign (=) means *only* equal in sign.

```
-w_3w_2w_9w_5 = w_4w_4w_{13}w_{13}
                                                                                     -w_3w_2w_{11}w_5 = w_4w_4w_{15}w_{13}
                                            -w_3w_2w_9w_6=w_4w_4w_{13}w_{14}
  -w_3w_2w_{11}w_6 = w_4w_4w_{15}w_{14}
                                            -w_3w_6w_9w_5 = w_4w_8w_{13}w_{13}
                                                                                     -w_3w_6w_9w_6 = w_4w_8w_{13}w_{14}
  -w_3w_6w_{11}w_5 = w_4w_8w_{15}w_{13}
                                                                                     -w_5w_9w_2w_3 = w_6w_{11}w_6w_{11}
                                           -w_3w_6w_{11}w_6 = w_4w_8w_{15}w_{14}
   -w_5w_9w_2w_4 = w_6w_{11}w_6w_{12}
                                            -w_5w_9w_4w_3 = w_6w_{11}w_8w_{11}
                                                                                     -w_5w_9w_4w_4 = w_6w_{11}w_8w_{12}
  -w_5w_{13}w_2w_3 = w_6w_{15}w_6w_{11}
                                           -w_5w_{13}w_2w_4 = w_6w_{15}w_6w_{12}
                                                                                     -w_5w_{13}w_4w_3 = w_6w_{15}w_8w_{11}
  -w_5w_{13}w_4w_4 = w_6w_{15}w_8w_{12}
                                           -w_{11}w_2w_9w_5 = w_{12}w_4w_{13}w_{13}
                                                                                     -w_{11}w_2w_9w_6 = w_{12}w_4w_{13}w_{14}
-w_{11}w_2w_{11}w_5 = w_{12}w_4w_{15}w_{13}
                                          -w_{11}w_2w_{11}w_6 = w_{12}w_4w_{15}w_{14}
                                                                                     -w_{11}w_6w_9w_5 = w_{12}w_8w_{13}w_{13}
-w_{11}w_6w_9w_6 = w_{12}w_8w_{13}w_{14}
                                          -w_{11}w_6w_{11}w_5 = w_{12}w_8w_{15}w_{13}
                                                                                     -w_{11}w_6w_{11}w_6 = w_{12}w_8w_{15}w_{14}
-w_{13}w_9w_2w_3 = w_{14}w_{11}w_6w_{11}
                                           -w_{13}w_9w_2w_4 = w_{14}w_{11}w_6w_{12}
                                                                                     -w_{13}w_9w_4w_3 = w_{14}w_{11}w_8w_{11}
-w_{13}w_9w_4w_4 = w_{14}w_{11}w_8w_{12}
                                          -w_{13}w_{13}w_2w_3 = w_{14}w_{15}w_6w_{11}
                                                                                     -w_{13}w_{13}w_2w_4 = w_{14}w_{15}w_6w_{12}
-w_{13}w_{13}w_4w_3 = w_{14}w_{15}w_8w_{11} - w_{13}w_{13}w_4w_4 = w_{14}w_{15}w_8w_{12}
```

Solving all Type 1 and Type 2 constraints gives the following solution set.

$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$w_7$	$w_8$	$w_9$	$w_{10}$	$w_{11}$	$w_{12}$	$w_{13}$	$w_{14}$	$w_{15}$	$w_{16}$
b	-1	-1	-1	-1	-1	0	-1	1	0	-1	-1	-1	-1	1	b
b	-1	-1	-1	-1	1	0	1	1	0	1	1	-1	1	-1	b
b	-1	-1	-1	1	-1	0	-1	-1	0	-1	-1	-1	1	-1	b
b	-1	-1	-1	1	1	0	1	-1	0	1	1	-1	-1	1	b
b	-1	-1	1	-1	-1	0	1	1	0	-1	1	1	1	-1	b
b	-1	-1	1	-1	1	0	-1	1	0	1	-1	1	-1	1	b
b	-1	-1	1	1	-1	0	1	-1	0	-1	1	1	-1	1	b
b	-1	-1	1	1	1	0	-1	-1	0	1	-1	1	1	-1	b
b	-1	1	-1	-1	-1	0	-1	-1	0	-1	1	-1	-1	-1	b
b	-1	1	-1	-1	1	0	1	-1	0	1	-1	-1	1	1	b
b	-1	1	-1	1	-1	0	-1	1	0	-1	1	-1	1	1	b
b	-1	1	-1	1	1	0	1	1	0	1	-1	-1	-1	-1	b
b	-1	1	1	-1	-1	0	1	-1	0	-1	-1	1	1	1	b
b	-1	1	1	-1	1	0	-1	-1	0	1	1	1	-1	-1	b
b	-1	1	1	1	-1	0	1	1	0	-1	-1	1	-1	-1	b
b	-1	1	1	1	1	0	-1	1	0	1	1	1	1	1	b
b	1	-1	-1	-1	-1	0	1	-1	0	-1	-1	-1	-1	-1	b
b	1	-1	-1	-1	1	0	-1	-1	0	1	1	-1	1	1	b
b	1	-1	-1	1	-1	0	1	1	0	-1	-1	-1	1	1	b
b	1	-1	-1	1	1	0	-1	1	0	1	1	-1	-1	-1	b
b	1	-1	1	-1	-1	0	-1	-1	0	-1	1	1	1	1	b
b	1	-1	1	-1	1	0	1	-1	0	1	-1	1	-1	-1	b
b	1	-1	1	1	-1	0	-1	1	0	-1	1	1	-1	-1	b
b	1	-1	1	1	1	0	1	1	0	1	-1	1	1	1	b
b	1	1	-1	-1	-1	0	1	1	0	-1	1	-1	-1	1	b
b	1	1	-1	-1	1	0	-1	1	0	1	-1	-1	1	-1	b
b	1	1	-1	1	-1	0	1	-1	0	-1	1	-1	1	-1	b
b	1	1	-1	1	1	0	-1	-1	0	1	-1	-1	-1	1	b
b	1	1	1	-1	-1	0	-1	1	0	-1	-1	1	1	-1	b
b	1	1	1	-1	1	0	1	1	0	1	1	1	-1	1	b
b	1	1	1	1	-1	0	-1	-1	0	-1	-1	1	-1	1	b
b	1	1	1	1	1	0	1	-1	0	1	1	1	1	-1	b

Any of these assignments are sufficient for calculating  $t_{m,n}$ .