Tromino Tiling

Liu Hang
Mathialagan, Surya
Pek Yu-Xuan Sean
NUS High School of Mathematics and Science
Mentor: Mr Goh Shu Liang

June 22, 2014

1 Introduction

The problem of tiling originated from antiquity, with the Ostomachion attributed to Archimedes. The basic premise of the problem is where a completed board is partitioned into smaller pieces. The specific problem of tiling a rectangular board with polyominoes is more recent, with earliest occurences in Japanese Tatami mats covering.

More recently, the advent of the computer has changed the way the problem has been tackled. The Dancing Links algorithm has been invented by Hiroshi Hitotsumatsu and Kohei Noshita in 1979 [1]. It has been used to tackle the ways to tile a finite alphabet of blocks into a rectangular grid. More generally, when the alphabet is not restricted to a finite set, a zero-suppressed binary decision diagram by Shin-ichi Minato [2] can be used to solve the problem.

However, these algorithms are both exponential in the size of the input. Hence, the need of a fast algorithm is required if we wish to solve the problem efficiently.

2 Definitions

Definition 0 A **polyomino** is a plane figure consisting of squares connected edgewise. A n-omino is a polyomino with n squares. Refer to Figure 1 for examples of polyominoes.
Definition 1 Alphabet refers to a non-empty set of polyominoes which may be infinite.
Definition 2 A **board** is a polyomino which we attempt to partition into smaller polyominoes from a given alphabet.
Definition 3 A **tiling** of a board refers to a way to partition the board for some alphabet.

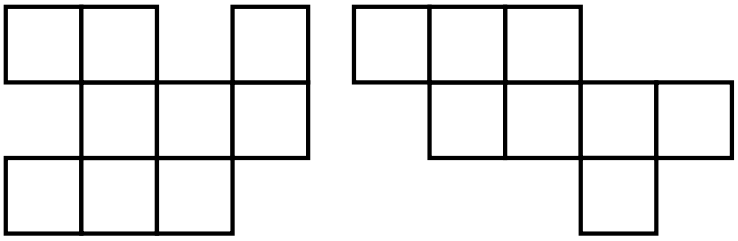


Figure 1: Examples of Polyominoes

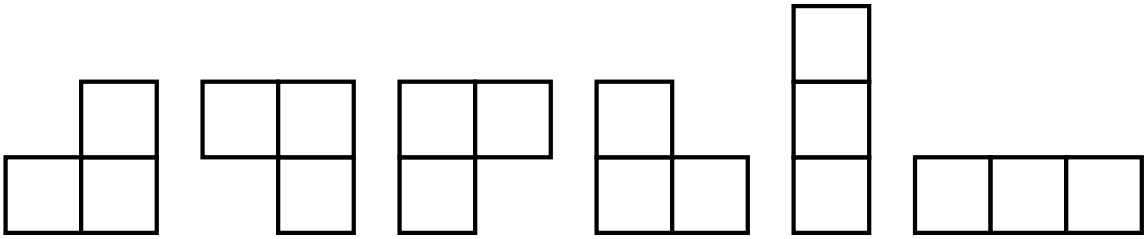


Figure 2: An alphabet of Trominoes

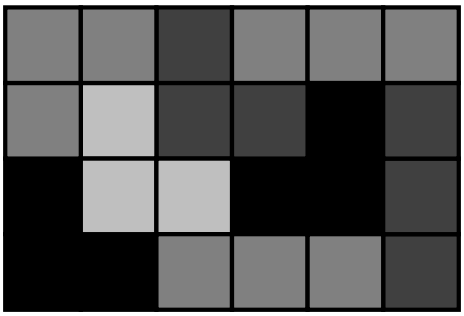


Figure 3: A tiling of a 4×6 board using an alphabet of trominoes.

3 Known Results and Problem Statement

3.1 Domino Tiling

The number of ways to tile a $m \times n$ rectangle with dominoes is given by:

$$\prod_{j=1}^{m}\prod_{k=1}^{n}(4\cos^2\frac{\pi j}{m+1}+4\cos^2\frac{\pi k}{n+1})^{\frac{1}{4}} \tag{1}$$

3.2 Tromino Tiling

[5] The closed form for the tromino tiling of a $2 \times 3k$ is given by:

$$T(2,3k)=\frac{1}{6}(3+\sqrt{3})\cdot(2+\sqrt{3})^k+\frac{1}{6}(3-\sqrt{3})\cdot(2-\sqrt{3})^k \tag{2}$$

Moreover, the generating function for the tromino tiling of a $3 \times k$ is given by:

$$G(k)=\frac{k^3-1}{-1+k+3k^2+6k^3+k^4-k^6} \tag{3}$$

A generating function is a formal power series where the coefficient of the term x^i is equivalent to the i^{th} element of a sequence $a_0, a_1, a_2, ...$

3.3 Problem Statement

In this project, we aim to find a closed form for the number of ways to tile a $4 \times 3k$ given an alphabet consisting of only trominos.

4 Methods

In order to properly explain the methodology, a few more definitions must be made.
Definition 4 A board is **right-aligned, horizontally-convex (RAHC)** if the following two conditions hold:

(a) All rows of squares either terminate on the same column or are empty.
(b) All rows of squares are continuous and have no gaps in between them.

An example of a RAHC board and a non-RAHC board is shown in Figure 4. A RAHC board can be expressed as a row vector $\{a_1, a_2, ..., a_n\}$ where a_i denotes the number of squares in row i. Moreover, let $T(a_1, a_2, ..., a_n)$ denote the number of ways to tile $\{a_1, a_2, ..., a_n\}$. We define $T(a_1, a_2, ..., a_n) = 0$ if any of $a_1, a_2, ..., a_n$ are negative.

Definition 5 The **maximum point** of a RAHC board denotes the value p such that

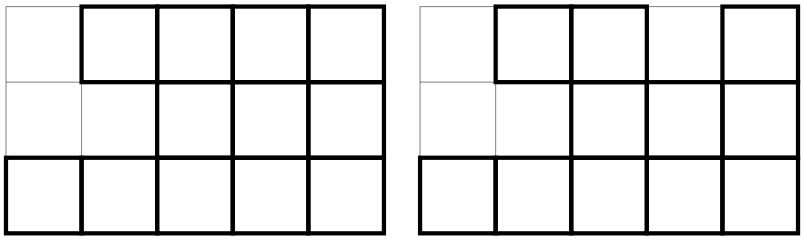


Figure 4: An example of a RAHC board and a non-RAHC board.

$a_p > a_i$ for all $i < p$ and $a_p \geq a_i$ for $i > p$. In other words, this denotes the top-most row with the most number of cells.

Definition 6 For a RAHC board with maximum point p, the left-most square on row p is the **top-left cell** of the RAHC board.

Definition 7 For a RAHC board given by the row vector $\{a_1, a_2, ..., a_n\}$ with maximum point p, the **effective value** of the RAHC board denotes the largest value e such that $a_p = a_{p+1} = ... = a_{p+e}$. In other words, the effective value denotes the number of consecutive rows after row p that have a_p squares in them.

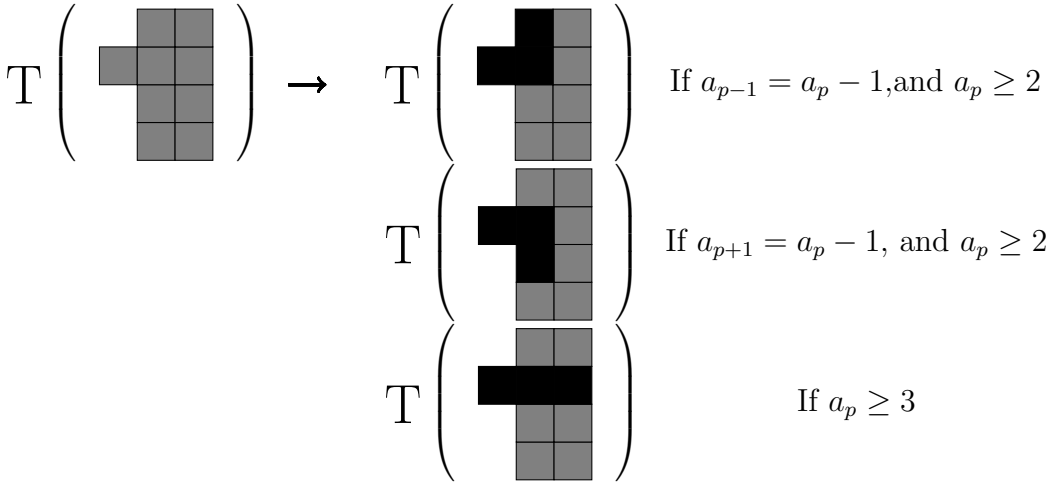
4.1 Initial Methods

Originally, we attempted to extend the method used to find the generating of tromino tiling $3 \times k$ rectangles as shown in [5]. However, finding the number of basic blocks of a $4 \times 3k$ rectangle is definitely not a simple task. We also tried various ways to colour the board hoping it would give us interesting properties regarding the tilings of such a board. However, these methods did not aid much in solving the problem. Hence, we looked in to the idea of recursion.

4.2 Method X

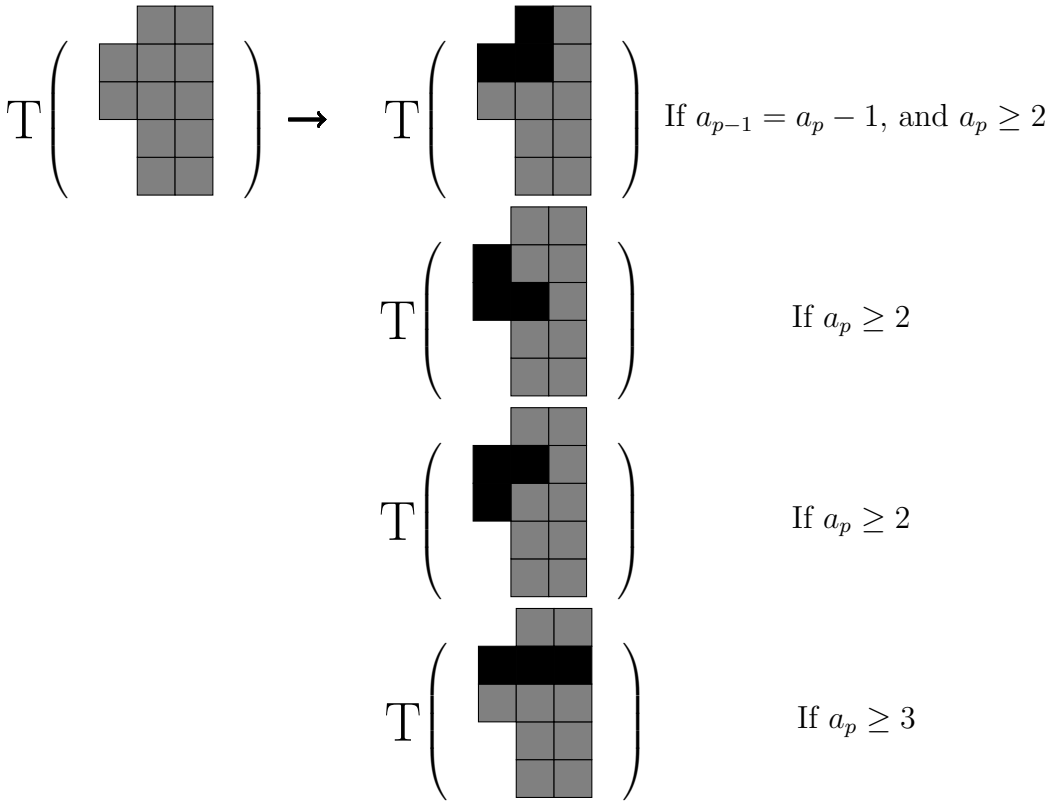
Consider a RAHC board with row vector $\{a_1, a_2, ..., a_n\}$ and maximum point p. Note that in order to obtain a tiling of the RAHC board, a few trominoes can be chosen to be removed at each stage such that the top-left cell of the RAHC board is removed, while maintaining the property that the resulting board is RAHC. Hence, by considering the possible effective values of the board, a set of rules to remove the top-left cell of a RAHC board.

EV = 0
$T(a_1, a_2, ..., a_{p-1}, a_p, a_{p+1}, ..., a_n)$
$= T(a_1, a_2, ..., a_{p-1} - 1, a_p - 2, a_{p+1}, ..., a_n)$ if $a_{p-1} = a_p - 1$ and $a_p \geq 2$
$+T(a_1, a_2, ..., a_{p-1}, a_p - 2, a_{p+1} - 1, ..., a_n)$ if $a_{p+1} = a_p - 1$ and $a_p \geq 2$
$+T(a_1, a_2, ..., a_{p-1}, a_p - 3, a_{p+1}, ..., a_n)$ if $a_p \geq 3$



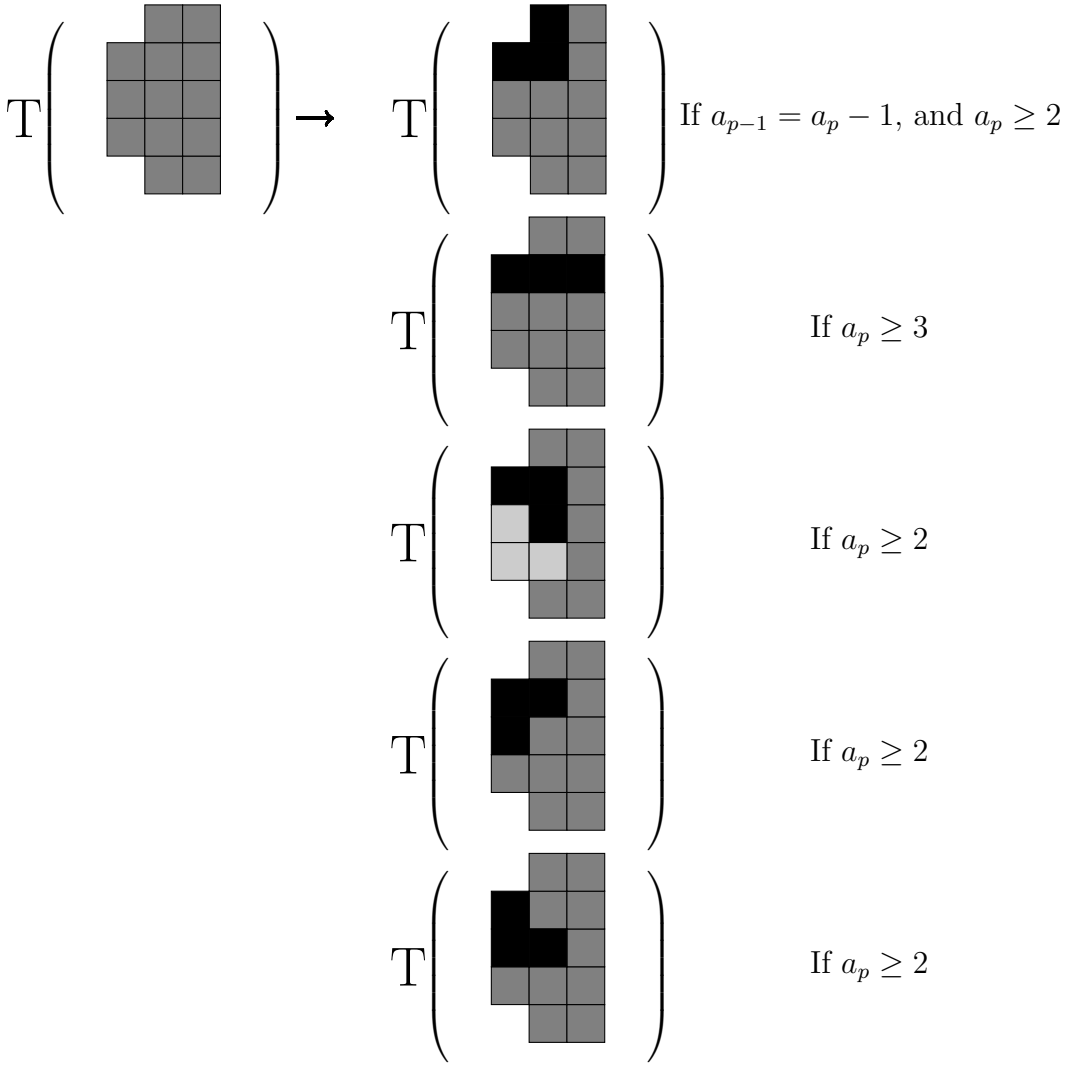
EV = 1
$T(a_1, a_2, ..., a_{p-1}, a_p, a_{p+1}, ..., a_n)$
$= T(a_1, a_2, ..., a_{p-1} - 1, a_p - 2, a_{p+1}, ..., a_n)$ if $a_{p-1} = a_p - 1$ and $a_p \geq 2$
$+T(a_1, a_2, ..., a_{p-1}, a_p - 1, a_{p+1} - 2, ..., a_n)$ if $a_p \geq 2$

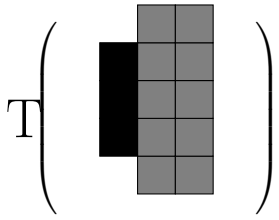
$+T(a_1, a_2, ..., a_{p-1}, a_p - 2, a_{p+1} - 1, ..., a_n)$ if $a_p \geq 2$
$+T(a_1, a_2, ..., a_{p-1}, a_p - 3, a_{p+1}, ..., a_n)$ if $a_p \geq 3$



EV = 2

$T(a_1, a_2, ..., a_{p-1}, a_p, a_{p+1}, ..., a_n)$
$= T(a_1, a_2, ..., a_{p-1} - 1, a_p - 2, a_{p+1}, ..., a_n)$ if $a_{p-1} = a_p - 1$ and $a_p \geq 2$
$+T(a_1, a_2, ..., a_{p-1}, a_p - 3, a_{p+1}, ..., a_n)$ if $a_p \geq 3$
$+T(a_1, a_2, ..., a_p - 2, a_{p+1} - 2, a_{p+2} - 2, ..., a_n)$ if $a_p \geq 2$
$+T(a_1, a_2, ..., a_{p-1}, a_p - 2, a_{p+1} - 1, ..., a_n)$ if $a_p \geq 2$
$+T(a_1, a_2, ..., a_{p-1}, a_p - 1, a_{p+1} - 2, ..., a_n)$ if $a_p \geq 2$
$+T(a_1, a_2, ..., a_p - 1, a_{p+1} - 1, a_{p+2} - 1, ..., a_n)$ $a_p \geq 1$

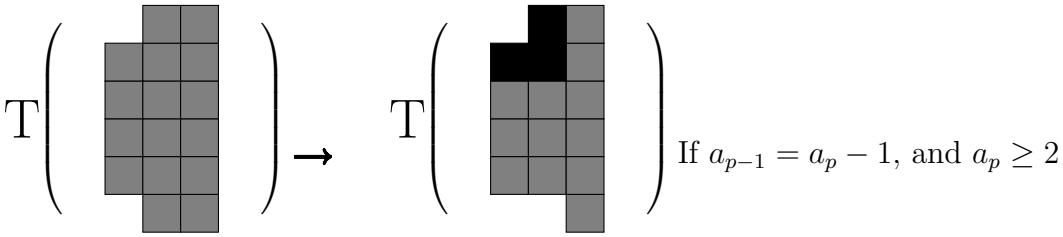




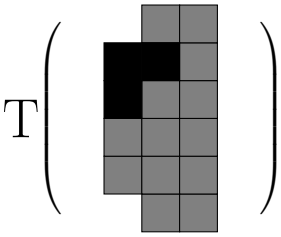
If $a_p \geq 1$

EV $\geq$ 3

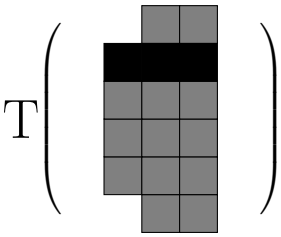
$T(a_1, a_2, ..., a_{p-1}, a_p, a_{p+1}, ..., a_n)$
$= T(a_1, a_2, ..., a_{p-1} - 1, a_p - 2, a_{p+1}, ..., a_n)$ if $a_{p-1} = a_p - 1$ and $a_p \geq 2$
$+T(a_1, a_2, ..., a_{p-1}, a_p - 2, a_{p+1} - 1, ..., a_n)$ if $a_p \geq 2$
$+T(a_1, a_2, ..., a_{p-1}, a_p - 3, a_{p+1}, ..., a_n)$ if $a_p \geq 3$
$+T(a_1, a_2, ..., a_{p-1}, a_p - 1, a_{p+1} - 2, ..., a_n)$ if $a_p \geq 2$
$+T(a_1, a_2, ..., a_p - 1, a_{p+1} - 1, a_{p+2} - 1, ..., a_n)$ if $a_p \geq 1$
$+T(a_1, a_2, ..., a_p - 2, a_{p+1} - 2, a_{p+2} - 2, ..., a_n)$ if $a_p \geq 2$
$+T(a_1, a_2, ..., a_p - 2, a_{p+1} - 1, a_{p+2} - 1, a_{p+3} - 1..., a_n)$ $a_p \geq 2$



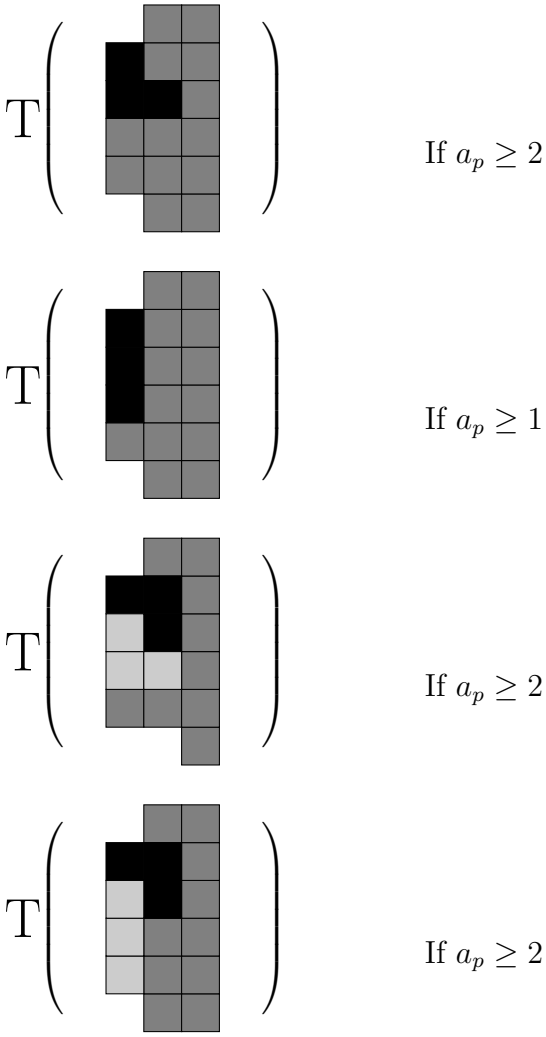
If $a_{p-1} = a_p - 1$, and $a_p \geq 2$



If $a_p \geq 2$



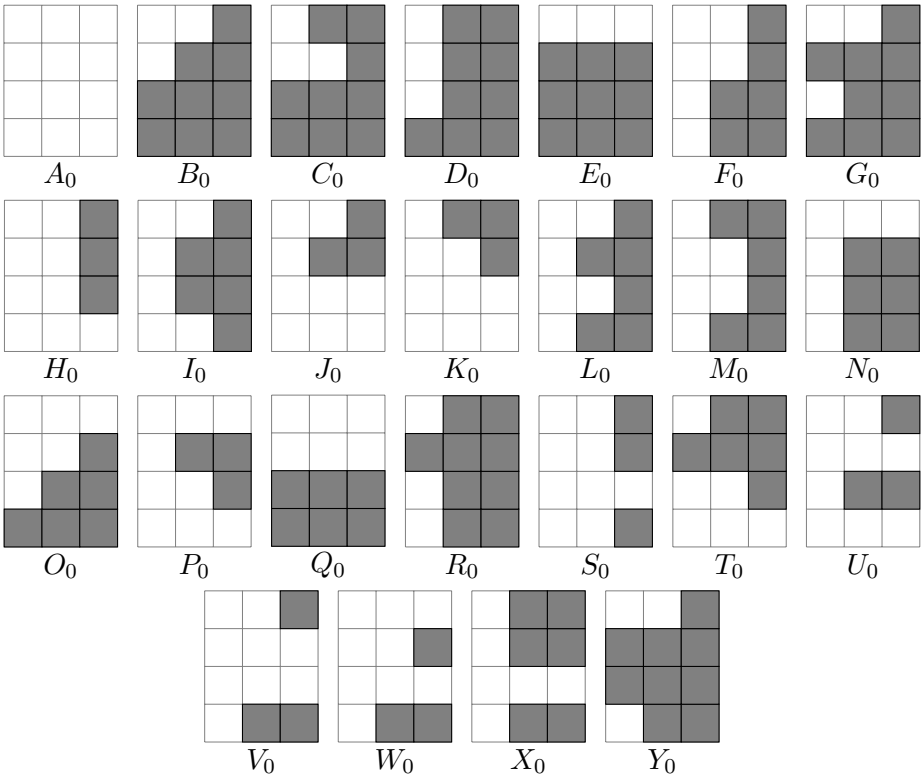
If $a_p \geq 3$



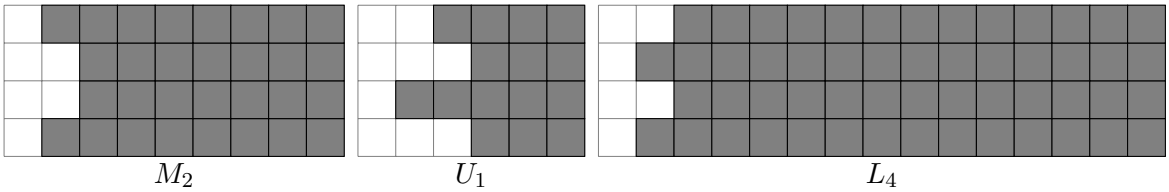
This algorithm was then implemented on a computer program to aid in finding the number of ways to tile a $4 \times 3n$ board for small values of n. These results will be shown in the **Results** section of this report.

4.3 Method M

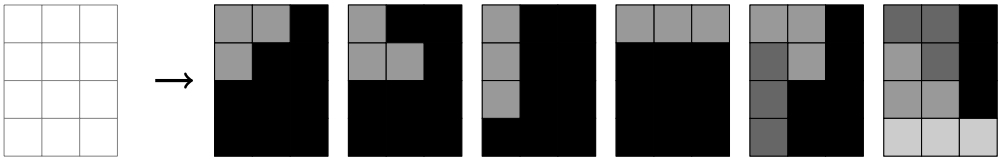
Method M uses the rules of Method X to partition a RAHC board, but in this method we aim to come up with a recursive relationship by considering all possible **states** of a RAHC Board. First, we define 25 states as follows:

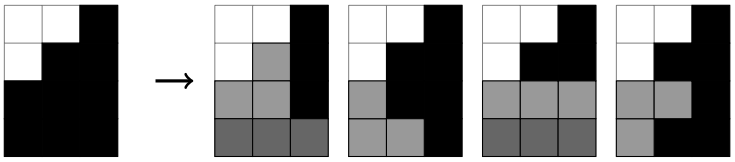


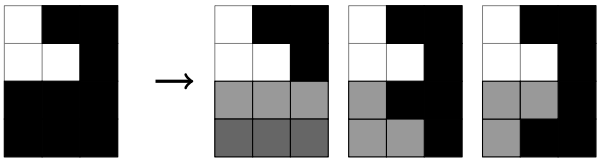
We define $A_n - Y_n$ as $A_0 - Y_0$ with a $4 \times 3n$ rectangular board to its right. Examples of some boards are shown below.

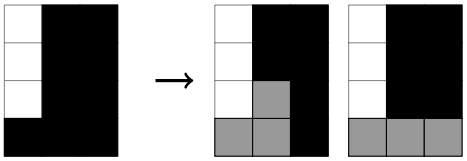


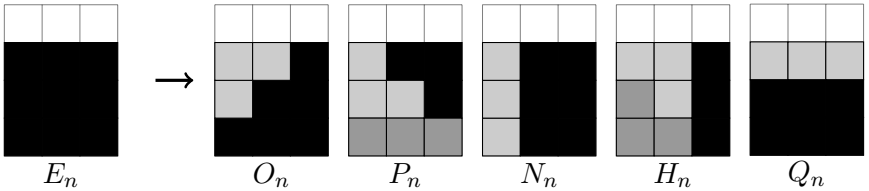
Now, note that each state has a specific group of states that they can move on to. The diagrams on the next page shows all possible states that each state can reach, and how one can tile the RAHC board at one state to reach another state.

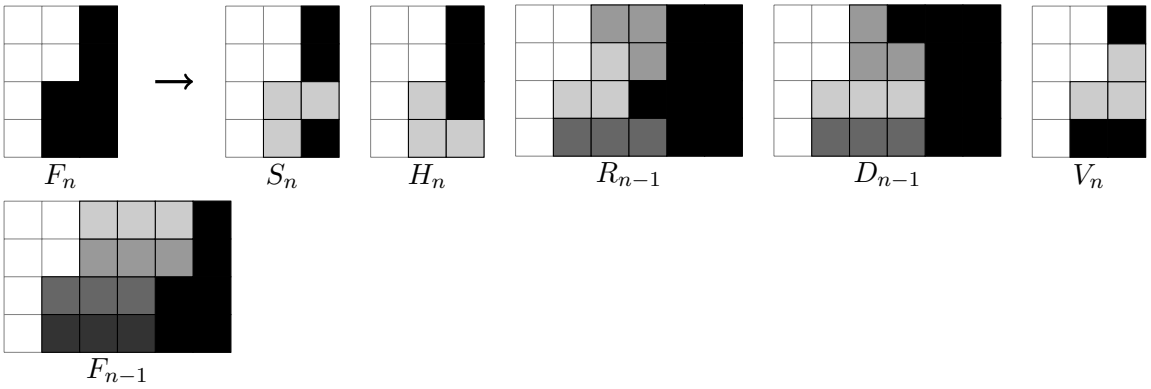

A_n
B_n
C_n
D_n
E_n
F_n
H_n

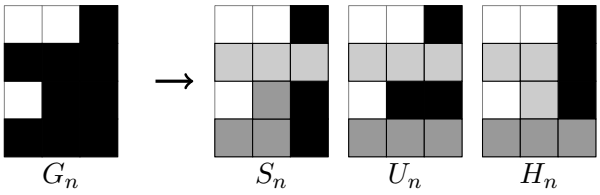

B_n
H_n
I_n
J_n
L_n

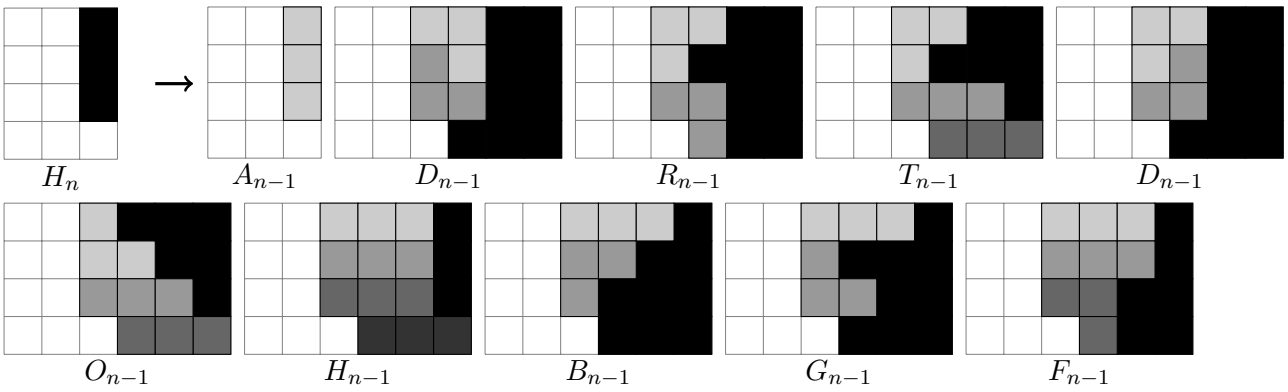

C_n
K_n
L_n
M_n

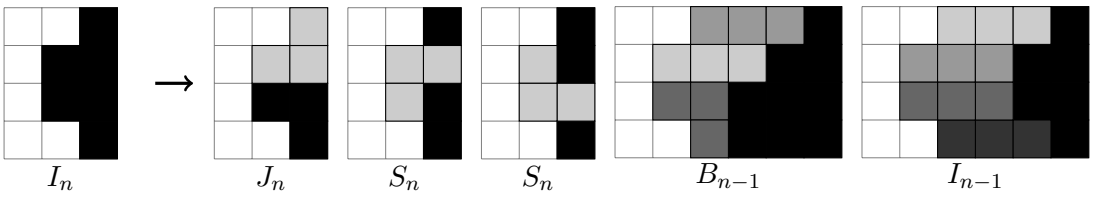

D_n
F_n
N_n

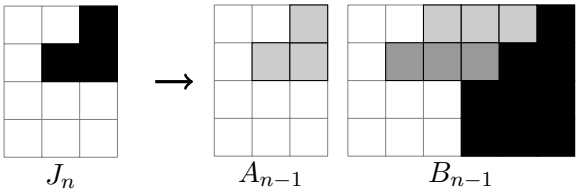

E_n
O_n
P_n
N_n
H_n
Q_n

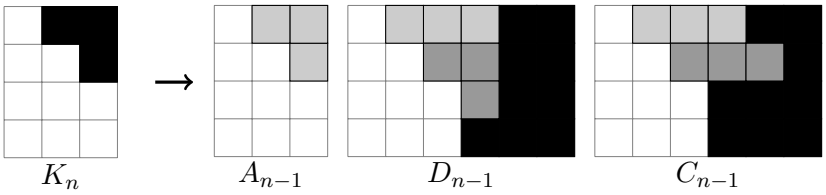

F_n
S_n
H_n
R_{n-1}
D_{n-1}
V_n
F_{n-1}

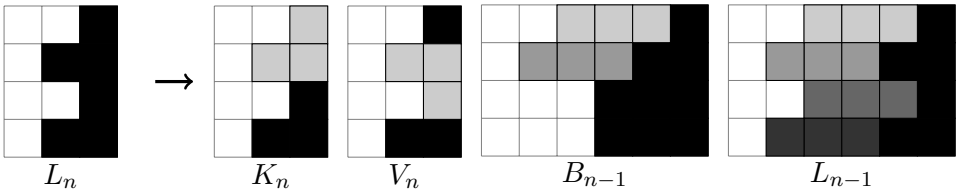

G_n
S_n
U_n
H_n

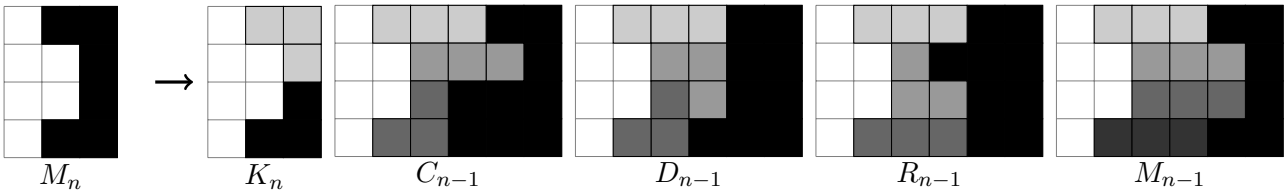

H_n
A_{n-1}
D_{n-1}
R_{n-1}
T_{n-1}
D_{n-1}
O_{n-1}
H_{n-1}
B_{n-1}
G_{n-1}
F_{n-1}

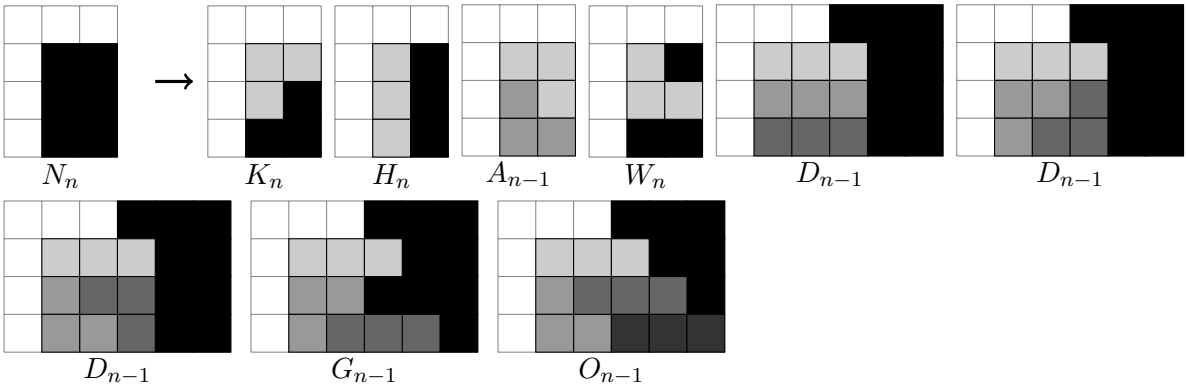

I_n
J_n
S_n
S_n
B_{n-1}
I_{n-1}

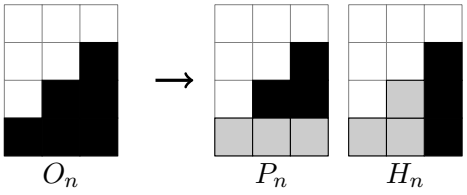

J_n
A_{n-1}
B_{n-1}

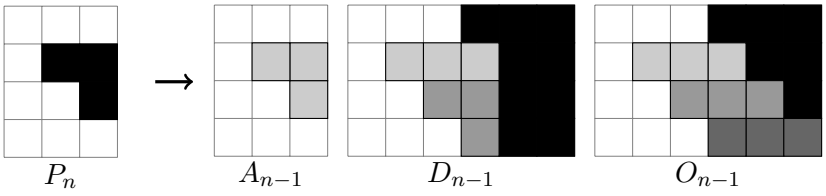

K_n
A_{n-1}
D_{n-1}
C_{n-1}

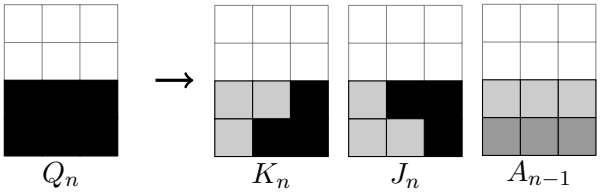

L_n
K_n
V_n
B_{n-1}
L_{n-1}

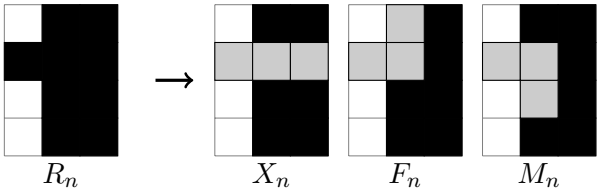

M_n
K_n
C_{n-1}
D_{n-1}
R_{n-1}
M_{n-1}

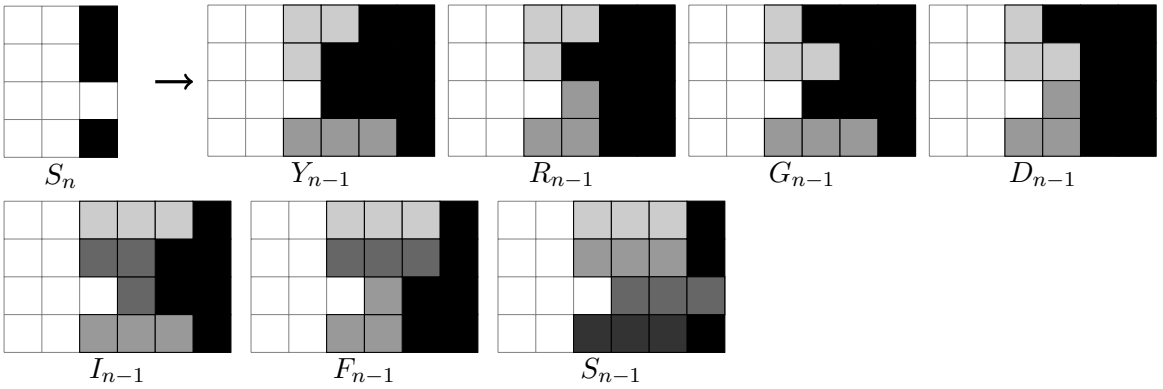

N_n
K_n
H_n
A_{n-1}
W_n
D_{n-1}
D_{n-1}
D_{n-1}
G_{n-1}
O_{n-1}

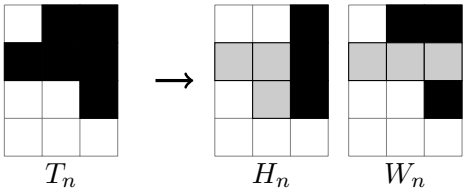

O_n
P_n
H_n

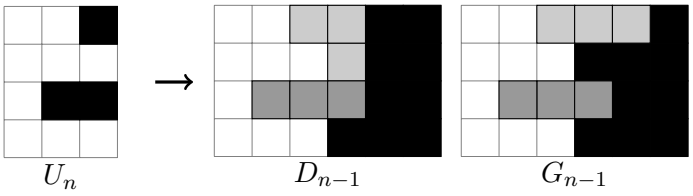

P_n
A_{n-1}
D_{n-1}
O_{n-1}

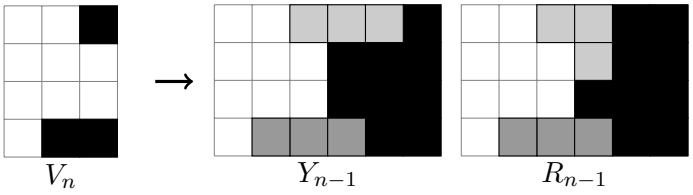

Q_n
K_n
J_n
A_{n-1}

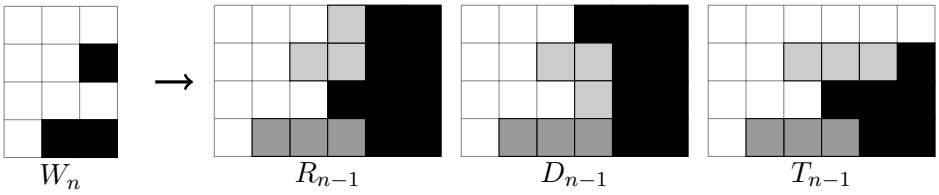

R_n
X_n
F_n
M_n

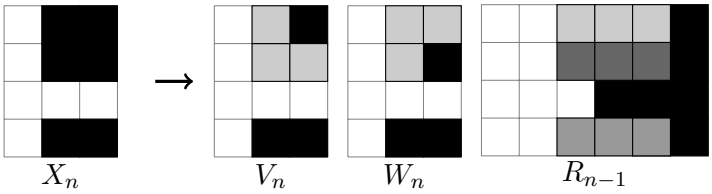

S_n
Y_{n-1}
R_{n-1}
G_{n-1}
D_{n-1}
I_{n-1}
F_{n-1}
S_{n-1}

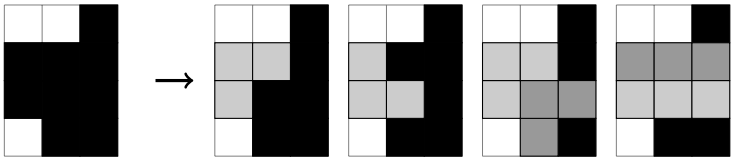

T_n
H_n
W_n


U_n
D_{n-1}
G_{n-1}


V_n
Y_{n-1}
R_{n-1}


W_n
R_{n-1}
D_{n-1}
T_{n-1}


X_n
V_n
W_n
R_{n-1}



Therefore, based on this, we will be able to obtain a 25 degree recurrence relationship as shown:

$A_n = B_{n-1} + C_{n-1} + D_{n-1} + E_{n-1} + F_{n-1} + H_{n-1}$
$B_n = H_n + I_n + J_n + L_n$
$C_n = K_n + L_n + M_n$
$D_n = F_n + N_n$
$E_n = H_n + N_n + O_n + P_n + Q_n$
$F_n = D_{n-1} + F_{n-1} + H_n + R_{n-1} + S_n + V_n$
$G_n = H_n + S_n + U_n$
$H_n = A_{n-1} + B_{n-1} + 2D_{n-1} + F_{n-1} + G_{n-1} + H_{n-1} + O_{n-1} + R_{n-1} + T_{n-1}$
$I_n = B_{n-1} + I_{n-1} + J_n + 2S_n$
$J_n = A_{n-1} + B_{n-1}$
$K_n = A_{n-1} + C_{n-1} + D_{n-1}$
$L_n = B_{n-1} + K_n + L_{n-1} + V_n$
$M_n = C_{n-1} + D_{n-1} + K_n + M_{n-1} + R_{n-1}$
$N_n = A_{n-1} + 3D_{n-1} + G_{n-1} + H_n + K_n + O_{n-1} + W_n$
$O_n = H_n + P_n$
$P_n = A_{n-1} + D_{n-1} + O_{n-1}$
$Q_n = A_{n-1} + J_n + K_n$
$R_n = F_n + M_n + X_n$
$S_n = D_{n-1} + F_{n-1} + G_{n-1} + I_{n-1} + R_{n-1} + S_{n-1} + Y_{n-1}$
$T_n = H_n + W_n$
$U_n = D_{n-1} + G_{n-1}$
$V_n = R_{n-1} + Y_{n-1}$
$W_n = D_{n-1} + R_{n-1} + T_{n-1}$
$X_n = R_{n-1} + V_n + W_n$
$Y_n = F_n + L_n + S_n + V_n$

5 Results

After implementing the algorithm described in Method X on a computer program, we obtain the following values.

n	Number of ways to Tile $4 \times 3n$
0	1
1	23
2	939
3	41813
4	1895145
5	86208957
6	3924499731
7	178682349823
8	8135650498647
9	370429531112741
10	16866286184557689
11	767950873073579951
12	34966119230441665595
13	1592067343861413081837
14	72489555274710984629691
15	3300573714050654978094583

Now let's look at the equations obtained in Method M. Note that it is possible to eliminate some of the functions easily, for example, consider the function B_n. It can be easily substituted with $H_n + I_n + J_n + L_n$, giving us the following set of equations:

$A_n = (H_{n-1} + I_{n-1} + J_{n-1} + L_{n-1}) + C_{n-1} + D_{n-1} + E_{n-1} + F_{n-1} + H_{n-1}$
$C_n = K_n + L_n + M_n$
$D_n = F_n + N_n$
$E_n = H_n + N_n + O_n + P_n + Q_n$
$F_n = D_{n-1} + F_{n-1} + H_n + R_{n-1} + S_n + V_n$
$G_n = H_n + S_n + U_n$
$H_n = A_{n-1}+(H_{n-1}+I_{n-1}+J_{n-1}+L_{n-1})+2D_{n-1}+F_{n-1}+G_{n-1}+H_{n-1}+O_{n-1}+R_{n-1}+T_{n-1}$
$I_n = (H_{n-1} + I_{n-1} + J_{n-1} + L_{n-1}) + I_{n-1} + J_n + 2S_n$
$J_n = A_{n-1} + (H_{n-1} + I_{n-1} + J_{n-1} + L_{n-1})$
$K_n = A_{n-1} + C_{n-1} + D_{n-1}$
$L_n = (H_{n-1} + I_{n-1} + J_{n-1} + L_{n-1}) + K_n + L_{n-1} + V_n$
$M_n = C_{n-1} + D_{n-1} + K_n + M_{n-1} + R_{n-1}$
$N_n = A_{n-1} + 3D_{n-1} + G_{n-1} + H_n + K_n + O_{n-1} + W_n$
$O_n = H_n + P_n$
$P_n = A_{n-1} + D_{n-1} + O_{n-1}$

$Q_n = A_{n-1} + J_n + K_n$
$R_n = F_n + M_n + X_n$
$S_n = D_{n-1} + F_{n-1} + G_{n-1} + I_{n-1} + R_{n-1} + S_{n-1} + Y_{n-1}$
$T_n = H_n + W_n$
$U_n = D_{n-1} + G_{n-1}$
$V_n = R_{n-1} + Y_{n-1}$
$W_n = D_{n-1} + R_{n-1} + T_{n-1}$
$X_n = R_{n-1} + V_n + W_n$
$Y_n = F_n + L_n + S_n + V_n$

Similarly, we eliminated most of the equations to obtain the following equations:

$A_n = A_{n-1} + 2F_{n-1} + 4H_{n-1} + I_{n-1} + 2J_{n-1} + 2K_{n-1} + 2L_{n-1} + M_{n-1} + 2N_{n-1} + 2P_{n-1}$
$F_n = A_{n-1} + 16F_{n-1} + 10H_{n-1} + 3I_{n-1} + 3J_{n-1} + 2K_{n-1} + 5L_{n-1} + 5M_{n-1} + 7N_{n-1} + 3P_{n-1} + 5S_{n-1} + 2U_{n-1} + 2V_{n-1} + W_{n-1} + 4X_{n-1}$
$H_n = A_{n-1} + 7F_{n-1} + 9H_{n-1} + 2I_{n-1} + 3J_{n-1} + 2K_{n-1} + 3L_{n-1} + 2M_{n-1} + 5N_{n-1} + 3P_{n-1} + S_{n-1} + U_{n-1} + W_{n-1} + X_{n-1}$
$I_n = A_{n-1} + 10F_{n-1} + 8H_{n-1} + 6I_{n-1} + 4J_{n-1} + 2K_{n-1} + 6L_{n-1} + 3M_{n-1} + 4N_{n-1} + 2P_{n-1} + 6S_{n-1} + 2U_{n-1} + 2V_{n-1} + 2X_{n-1}$
$J_n = A_{n-1} + 2F_{n-1} + 5H_{n-1} + 2I_{n-1} + 3J_{n-1} + 2K_{n-1} + 3L_{n-1} + M_{n-1} + 2N_{n-1} + 2P_{n-1}$
$K_n = A_{n-1} + 3F_{n-1} + 4H_{n-1} + I_{n-1} + 2J_{n-1} + 3K_{n-1} + 3L_{n-1} + 2M_{n-1} + 3N_{n-1} + 2P_{n-1}$
$L_n = A_{n-1} + 5F_{n-1} + 5H_{n-1} + 2I_{n-1} + 3J_{n-1} + 3K_{n-1} + 6L_{n-1} + 3M_{n-1} + 3N_{n-1} + 2P_{n-1} + S_{n-1} + V_{n-1} + X_{n-1}$
$M_n = A_{n-1} + 6F_{n-1} + 4H_{n-1} + I_{n-1} + 2J_{n-1} + 4K_{n-1} + 4L_{n-1} + 6M_{n-1} + 4N_{n-1} + 2P_{n-1} + 2X_{n-1}$
$N_n = 3A_{n-1} + 17F_{n-1} + 20H_{n-1} + 4I_{n-1} + 7J_{n-1} + 7K_{n-1} + 8L_{n-1} + 6M_{n-1} + 14N_{n-1} + 8P_{n-1} + 2S_{n-1} + 2U_{n-1} + 2W_{n-1} + 2X_{n-1}$
$P_n = A_{n-1} + 3F_{n-1} + 5H_{n-1} + I_{n-1} + 2J_{n-1} + 2K_{n-1} + 2L_{n-1} + M_{n-1} + 3N_{n-1} + 3P_{n-1}$
$S_n = 4F_{n-1} + H_{n-1} + I_{n-1} + L_{n-1} + M_{n-1} + N_{n-1} + 3S_{n-1} + U_{n-1} + V_{n-1} + X_{n-1}$
$U_n = F_{n-1} + H_{n-1} + N_{n-1} + S_{n-1} + U_{n-1}$
$V_n = 2F_{n-1} + L_{n-1} + M_{n-1} + S_{n-1} + V_{n-1} + X_{n-1}$
$W_n = 2F_{n-1} + H_{n-1} + M_{n-1} + N_{n-1} + W_{n-1} + X_{n-1}$
$X_n = 5F_{n-1} + H_{n-1} + L_{n-1} + 3M_{n-1} + N_{n-1} + S_{n-1} + V_{n-1} + W_{n-1} + 3X_{n-1}$
Note that we can express the above system of linear equations in a matrix multiplication

as shown:

$$\begin{pmatrix} A_n \\ F_n \\ H_n \\ I_n \\ J_n \\ K_n \\ L_n \\ M_n \\ N_n \\ P_n \\ S_n \\ U_n \\ V_n \\ W_n \\ X_n \end{pmatrix} = \begin{pmatrix} 1 & 2 & 4 & 1 & 2 & 2 & 2 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 16 & 10 & 3 & 3 & 2 & 5 & 5 & 7 & 3 & 5 & 2 & 2 & 1 & 4 \\ 1 & 7 & 9 & 2 & 3 & 2 & 3 & 2 & 5 & 3 & 1 & 1 & 0 & 1 & 1 \\ 1 & 10 & 8 & 6 & 4 & 2 & 6 & 3 & 4 & 2 & 6 & 2 & 2 & 0 & 2 \\ 1 & 2 & 5 & 2 & 3 & 2 & 3 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 4 & 1 & 2 & 3 & 3 & 2 & 3 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 5 & 5 & 2 & 3 & 3 & 6 & 3 & 3 & 2 & 1 & 0 & 1 & 0 & 1 \\ 1 & 6 & 4 & 1 & 2 & 4 & 4 & 6 & 4 & 2 & 0 & 0 & 0 & 0 & 2 \\ 3 & 17 & 20 & 4 & 7 & 7 & 8 & 6 & 14 & 8 & 2 & 2 & 0 & 2 & 2 \\ 1 & 3 & 5 & 1 & 2 & 2 & 2 & 1 & 3 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 3 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 5 & 1 & 0 & 0 & 0 & 1 & 3 & 1 & 0 & 1 & 0 & 1 & 1 & 3 \end{pmatrix} \times \begin{pmatrix} A_{n-1} \\ F_{n-1} \\ H_{n-1} \\ I_{n-1} \\ J_{n-1} \\ K_{n-1} \\ L_{n-1} \\ M_{n-1} \\ N_{n-1} \\ P_{n-1} \\ S_{n-1} \\ U_{n-1} \\ V_{n-1} \\ W_{n-1} \\ X_{n-1} \end{pmatrix}$$

From Method X, we are able to obtain that $A_0 = F_0 = H_0 = I_0 = J_0 = K_0 = L_0 = M_0 = P_0 = 1$, $N_0 = 3$ and $S_0 = U_0 = V_0 = W_0 = X_0 = 0$. Hence, we can express the above expression as follows:

$$
\begin{pmatrix} A_n & F_n & H_n & I_n & J_n & K_n & L_n & M_n & N_n & P_n & S_n & U_n & V_n & W_n & X_n \end{pmatrix}^T
$$

$$
= \begin{pmatrix}
1 & 2 & 4 & 1 & 2 & 2 & 2 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\
1 & 16 & 10 & 3 & 3 & 2 & 5 & 5 & 7 & 3 & 5 & 2 & 2 & 1 & 4 \\
1 & 7 & 9 & 2 & 3 & 2 & 3 & 2 & 5 & 3 & 1 & 1 & 0 & 1 & 1 \\
1 & 10 & 8 & 6 & 4 & 2 & 6 & 3 & 4 & 2 & 6 & 2 & 2 & 0 & 2 \\
1 & 2 & 5 & 2 & 3 & 2 & 3 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\
1 & 3 & 4 & 1 & 2 & 3 & 3 & 2 & 3 & 2 & 0 & 0 & 0 & 0 & 0 \\
1 & 5 & 5 & 2 & 3 & 3 & 6 & 3 & 3 & 2 & 1 & 0 & 1 & 0 & 1 \\
1 & 6 & 4 & 1 & 2 & 4 & 4 & 6 & 4 & 2 & 0 & 0 & 0 & 0 & 2 \\
3 & 17 & 20 & 4 & 7 & 7 & 8 & 6 & 14 & 8 & 2 & 2 & 0 & 2 & 2 \\
1 & 3 & 5 & 1 & 2 & 2 & 2 & 1 & 3 & 3 & 0 & 0 & 0 & 0 & 0 \\
0 & 4 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 3 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 5 & 1 & 0 & 0 & 0 & 1 & 3 & 1 & 0 & 1 & 0 & 1 & 1 & 3
\end{pmatrix}
\begin{pmatrix} A_{n-1} \\ F_{n-1} \\ H_{n-1} \\ I_{n-1} \\ J_{n-1} \\ K_{n-1} \\ L_{n-1} \\ M_{n-1} \\ N_{n-1} \\ P_{n-1} \\ S_{n-1} \\ U_{n-1} \\ V_{n-1} \\ W_{n-1} \\ X_{n-1} \end{pmatrix}
$$

$$
= \begin{pmatrix}
1 & 2 & 4 & 1 & 2 & 2 & 2 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\
1 & 16 & 10 & 3 & 3 & 2 & 5 & 5 & 7 & 3 & 5 & 2 & 2 & 1 & 4 \\
1 & 7 & 9 & 2 & 3 & 2 & 3 & 2 & 5 & 3 & 1 & 1 & 0 & 1 & 1 \\
1 & 10 & 8 & 6 & 4 & 2 & 6 & 3 & 4 & 2 & 6 & 2 & 2 & 0 & 2 \\
1 & 2 & 5 & 2 & 3 & 2 & 3 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\
1 & 3 & 4 & 1 & 2 & 3 & 3 & 2 & 3 & 2 & 0 & 0 & 0 & 0 & 0 \\
1 & 5 & 5 & 2 & 3 & 3 & 6 & 3 & 3 & 2 & 1 & 0 & 1 & 0 & 1 \\
1 & 6 & 4 & 1 & 2 & 4 & 4 & 6 & 4 & 2 & 0 & 0 & 0 & 0 & 2 \\
3 & 17 & 20 & 4 & 7 & 7 & 8 & 6 & 14 & 8 & 2 & 2 & 0 & 2 & 2 \\
1 & 3 & 5 & 1 & 2 & 2 & 2 & 1 & 3 & 3 & 0 & 0 & 0 & 0 & 0 \\
0 & 4 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 3 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 5 & 1 & 0 & 0 & 0 & 1 & 3 & 1 & 0 & 1 & 0 & 1 & 1 & 3
\end{pmatrix}^2
\begin{pmatrix} A_{n-2} \\ F_{n-2} \\ H_{n-2} \\ I_{n-2} \\ J_{n-2} \\ K_{n-2} \\ L_{n-2} \\ M_{n-2} \\ N_{n-2} \\ P_{n-2} \\ S_{n-2} \\ U_{n-2} \\ V_{n-2} \\ W_{n-2} \\ X_{n-2} \end{pmatrix}
$$

$$
= \ldots
$$

$$= \begin{pmatrix} 1 & 2 & 4 & 1 & 2 & 2 & 2 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 16 & 10 & 3 & 3 & 2 & 5 & 5 & 7 & 3 & 5 & 2 & 2 & 1 & 4 \\ 1 & 7 & 9 & 2 & 3 & 2 & 3 & 2 & 5 & 3 & 1 & 1 & 0 & 1 & 1 \\ 1 & 10 & 8 & 6 & 4 & 2 & 6 & 3 & 4 & 2 & 6 & 2 & 2 & 0 & 2 \\ 1 & 2 & 5 & 2 & 3 & 2 & 3 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 4 & 1 & 2 & 3 & 3 & 2 & 3 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 5 & 5 & 2 & 3 & 3 & 6 & 3 & 3 & 2 & 1 & 0 & 1 & 0 & 1 \\ 1 & 6 & 4 & 1 & 2 & 4 & 4 & 6 & 4 & 2 & 0 & 0 & 0 & 0 & 2 \\ 3 & 17 & 20 & 4 & 7 & 7 & 8 & 6 & 14 & 8 & 2 & 2 & 0 & 2 & 2 \\ 1 & 3 & 5 & 1 & 2 & 2 & 2 & 1 & 3 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 3 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 5 & 1 & 0 & 0 & 0 & 1 & 3 & 1 & 0 & 1 & 0 & 1 & 1 & 3 \end{pmatrix}^n \times \begin{pmatrix} A_0 \\ F_0 \\ H_0 \\ I_0 \\ J_0 \\ K_0 \\ L_0 \\ M_0 \\ N_0 \\ P_0 \\ S_0 \\ U_0 \\ V_0 \\ W_0 \\ X_0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 2 & 4 & 1 & 2 & 2 & 2 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 16 & 10 & 3 & 3 & 2 & 5 & 5 & 7 & 3 & 5 & 2 & 2 & 1 & 4 \\ 1 & 7 & 9 & 2 & 3 & 2 & 3 & 2 & 5 & 3 & 1 & 1 & 0 & 1 & 1 \\ 1 & 10 & 8 & 6 & 4 & 2 & 6 & 3 & 4 & 2 & 6 & 2 & 2 & 0 & 2 \\ 1 & 2 & 5 & 2 & 3 & 2 & 3 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 4 & 1 & 2 & 3 & 3 & 2 & 3 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 5 & 5 & 2 & 3 & 3 & 6 & 3 & 3 & 2 & 1 & 0 & 1 & 0 & 1 \\ 1 & 6 & 4 & 1 & 2 & 4 & 4 & 6 & 4 & 2 & 0 & 0 & 0 & 0 & 2 \\ 3 & 17 & 20 & 4 & 7 & 7 & 8 & 6 & 14 & 8 & 2 & 2 & 0 & 2 & 2 \\ 1 & 3 & 5 & 1 & 2 & 2 & 2 & 1 & 3 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 3 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 5 & 1 & 0 & 0 & 0 & 1 & 3 & 1 & 0 & 1 & 0 & 1 & 1 & 3 \end{pmatrix}^n \times \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 3 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Hence, using this matrix, we will be able to calculate the value of A_n, and this gives us the number of ways to tile a $4 \times 3n$.

The generating function is:

$$T(x) = \frac{-x^{14} + 45 * x^{13} - 790 * x^{12} + 7195 * x^{11} - 37791 * x^{10} + 120544 * x^9 - 241021 * x^8 + 30738}{x^{15} - 56 * x^{14} + 1223 * x^{13} - 13643 * x^{12} + 87066 * x^{11} - 338409 * x^{10} + 836269 * x^9 - 1345297 * x^8 +}$$

6 Conclusion

In conclusion, this paper illustrates a method in which a closed form for the k^{th} term of the tiling of a $4 \times 3k$ board can be determined. This method can be trivially generalised to higher dimensional boards or different alphabets.

References

[1] Hitotumatu, Hirosi; Noshita, Kohei (1979). "A Technique for Implementing Backtrack Algorithms and its Application". Information Processing Letters 8 (4): 174175. doi:10.1016/0020-0190(79)90016-4

[2] Shin-ichi Minato, "Zero-suppressed BDDs for set manipulation in combinatorial problems", DAC '93: Proceedings of the 30th international conference on Design automation, 1993

[3] Erica Klarreich (1991) "Counting Domino Tilings of Mutilated Grids", OSU REU '91

[4] Bodini, Olivier; Latapy, Matthieu (2003),""Generalized Tilings with Height Functions", Morfismos 7 (1): 4768, ISSN 1870-6525.

[5] Silvia Heubach, Phyllis Chinn, Patrick Callahan. ""Tiling With Trominoes"