A Simplification of Stanley's Young Tableau

Teddy Finkelstein, Sarah Linden, Michael Fagundo

Young Tableaus and Hook Lengths

To understand Stanley's paper, we must first understand the basics of a **Young Tableau**. A Young Tableau is a shape made up of squares, in which each row is ascending left to right, and each column is ascending top to bottom.

Here is an example:

1	3	5
2	6	
4		

Notice how every row and column are in descending order.

Now, every Young Tableau has what are called **hook lengths**. The hook length of a cell is equal to the number of squares one must pass through in order to leave the Young Tableau walking to right plus the number of squares one must pass through to leave moving downwards, plus the square of origin.

Here are the hook lengths for our above example:

5	3	1
3	1	
1		

Say we name the cell in the top left corner A:

A		

If someone is standing in A, they must pass through 2 cells to leave to the left, or 2 cells to leave from the bottom. We include our origin cell, so we find a hook length of 2 + 2 + 1 = 5:

A	→	→
↓		
↓		

Now that we understand hook length, we can now move onto the **hook length formula**. The hook length formula gives us a way to find out how many different ways we could arrange the different numbers in a Young Tableau. The formula is:

# of ways to fill in a Young Tableau = (number of cells)! / product of the hook lengths of each cell

Returning to our example above, we have the following:

Young Tableau

1	3	5
2	6	
4		

Hook Length

5	3	1
3	1	
1		

Therefore, we get the formula:

$$\frac{6 \times 5 \times 4 \times 3 \times 2 \times 1}{5 \times 3 \times 3 \times 1 \times 1 \times 1} = 16$$

Using the formula, we find that there are 16 ways to organize the numbers inside the Young Tableau, and still have the row and columns in ascending order.

Below, we will examine the most simple applications of the hook length formula.

1

number of cells = 1
hook lengths = 1

So, 1!/1 = 1 = # of ways to arrange numbers -> this means that you can only order numbers in a one cell cube one way so that they are ascending as you move to the right and down

2	1

number of cells = 2
hook lengths = 1, 2

2!/(2*1) = 1 -> there is one way to order numbers in a 1 x 2 set of cells such that they are ascending as you move to the right and down

3	1
1	

number of cells = 3
hook lengths = 3, 1, 1

3!/(3*1*1) = 2 -> there are two ways to order numbers in these cells, which are demonstrated below

1	2
3	

1	3
2	

4	2	1
1		

number of cells = 4
hook lengths = 4,2,1,1

4!/(4*2*1*1) = 3 -> there are three ways to order numbers in these cells, which are demonstrated below

1	2	3
4		

1	2	4
3		

1	2	3
4		

Now that we have a strong understanding of young tableau, hook length, and the hook length formula, we can dive into Stanley's paper.

Stanley's Paper

The foundational premise of Stanley's paper is that the number of ways to draw a certain number of cuts in a polygon is equal to the number of ways to fill in a corresponding young tableau. In fact, there is a corresponding organization of numbers within each young tableau that match up to a specific pattern of cuts in the polygon.

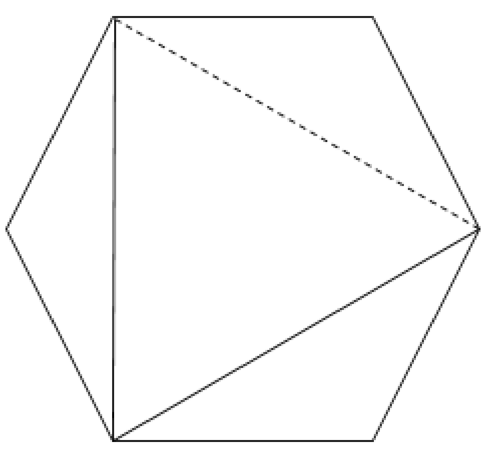
The formula for determining the number of cells and rows is simple. For an *n*-gon with *k* cuts, the number of L cells (L) is equal to:

$$L = n + k - 1$$

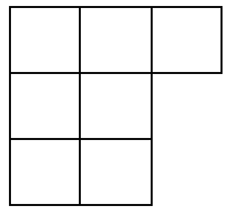
And the number of R rows is equal to:

$$R = k + 1$$

Let's use a hexagon with two cuts as an example. A hexagon has six sides (n = 6), and it has two cuts (k = 2). Therefore, our young tableau should have 7 cells (6 + 2 -1) in 3 rows (2 + 1). Now, in order to determine how the young tableau is shaped, we must look at a hexagon:



If we use the solid line as our cuts, we can place the dotted line to show the full triangulation of the hexagon. The corresponding young tableau shape is as follows

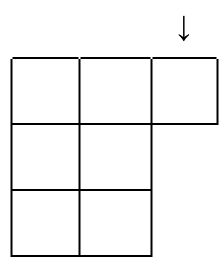


The bottom two rows correspond to cuts we made, and the third column from the left corresponds to the number of additional cuts needed to fully triangulate the polygon, like this:

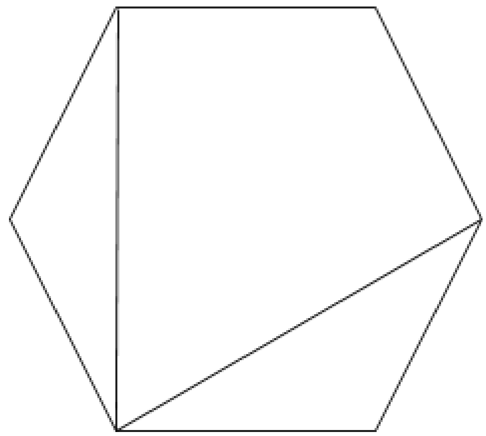
This cell corresponds to the number of additional cuts needed to fully triangulate the polygon.
↓

This row correspond to one made cut →

This row corresponds to the other made cut →

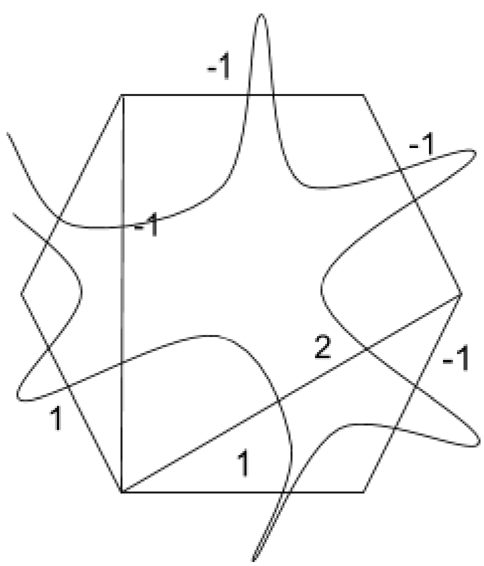


Let’s look at two cutting a hexagon, like this:



A two cut hexagon matches up to a specifc

Example:



1	3	4	○
2	6		
5	7		
□	●		

rows = 3

observe that these are all equivalent

rooms = 3
positive walls = 3
□ = 3

1	2	3	4	5	6	7
2	1	-1	-1	1	-1	-1
□	□	●	○	□	●	●

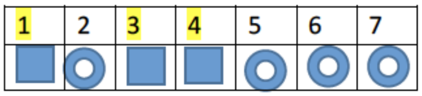
How to Derive Cuts from A Young Tableau

1	2	6
3	5	
4	7	

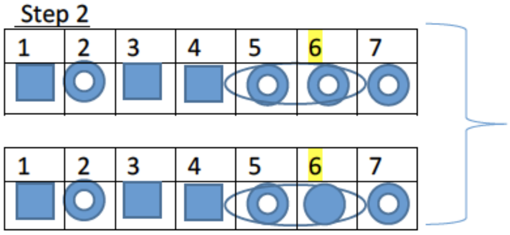
Step 1

-The numbers in the first column will take on the positive values, denoted by a square

- The rest of the numbers take a negative value, denoted by O's

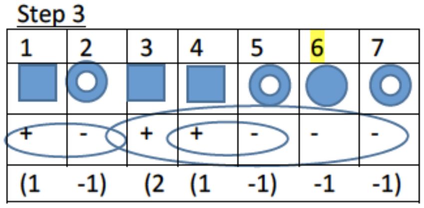


Step 2



In this step, we determine which number belongs in the position of the top right box. As you can see, you take the first pair of O's, placing the second one in the position of the top right box, which is six in this example.

Step 3



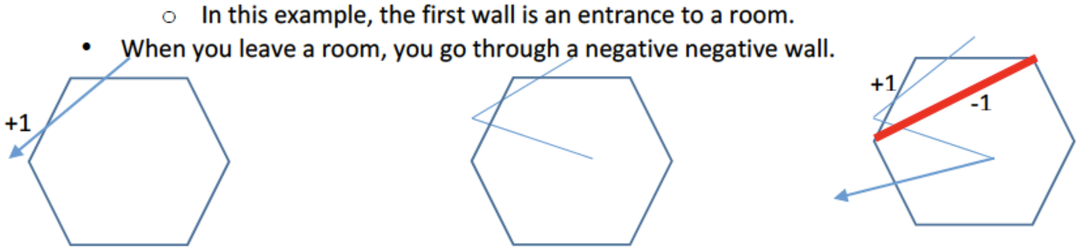
Goal: To make pairs of positives and negatives that add out to 0.

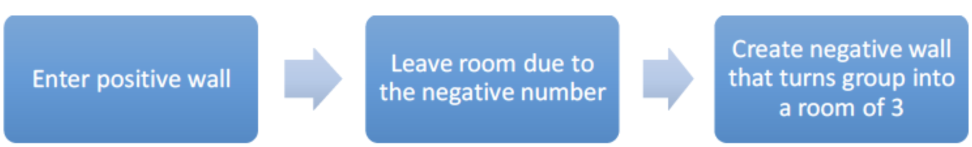
-By doing this, we can identify given "rooms" within the hexagon, as well as their size and order.

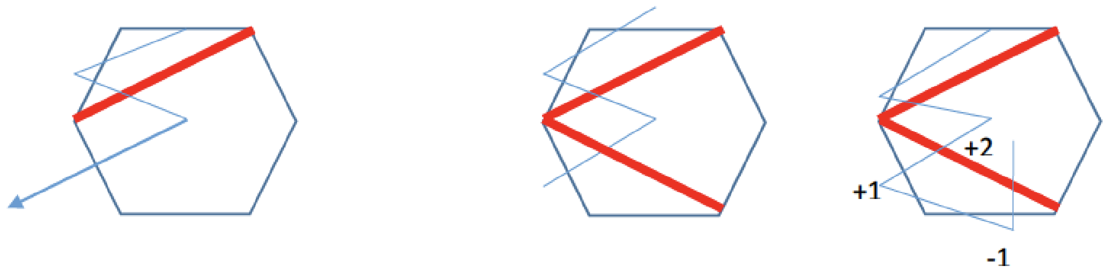
-One of these combinations will be a 2, -1 & -1 another -1; This tells us the location of a cut

*We now have a subset of groups, which allows us to think about the hexagon.

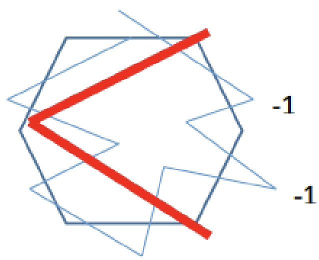
- Each group starts with a positive, when you enter a "room" of the hexagon, you enter a positive wall, which is one of the three positive numbers.
  - In this example, the first wall is an entrance to a room.
- When you leave a room, you go through a negative negative wall.







You enter the second room through positive +2 wall → However, there is another group within this, thus enter another room → Leave through negative back into "2 Room"



Following the last part, you left the subgroup. Now, you must get out of the “2 Room.” Therefore, follow two more “-1” walls to exit.