5-2022

Studying Extended Sets from Young Tableaux

Eric S. Nofziger

BUTLER UNIVERSITY HONORS PROGRAM

Honors Thesis Certification

Please type all information in this section:

Applicant Eric S Nofziger
(Name as it is to appear on diploma)

Thesis title Studying Extended Sets from Young Tableaux

Intended date of commencement May 7, 2022

Read, approved, and signed by:

Thesis adviser(s) Amber Russell 5/5/22
Date

Date

Reader(s) Jonathan Webster 5/5/22
Date

Date

Certified by
Director, Honors Program Date

Studying Extended Sets from Young Tableaux

A Thesis

Presented to the Department of Mathematics

College of Liberal Arts and Sciences

and

The Honors Program

of

Butler University

In Partial Fulfillment

of the Requirements for Graduation Honors

Eric S Nofziger

May 5, 2022

1 Introduction

Young tableaux (singular "tableau") are fairly simple combinatorial objects that have widespread applicability within the field of representation theory. The most well-known application of Young tableaux is in the study of representations of the symmetric group S_n—the group containing all possible permutations of the set $\{1, \ldots, n\}$. See [Ful96] for a thorough treatment of this. The topic using Young tableaux that is most pertinent to this project is the study of Springer fibers.

The Springer correspondence is a way of relating different objects in the area of algebra known as Lie theory. The key to understanding this correspondence was the study of the map called the Springer resolution. The preimages of points from this map are known as Springer fibers. Young tableaux have been helpful in studying these fibers, which in turn has led to broader discoveries in Lie theory. See [CM93] and [Hum95] for overviews of these. In recent work, William Graham, Martha Precup, and Amber Russell have been working on an extended version of the Springer resolution. See [Gra19], [Rus20], and [GPR20] for these results. As part of this, they have developed a connection with Young tableaux that allows them to better understand the extended Springer fibers. In this paper, we will explore the combinatorics related to this connection.

1.1 Definitions and Background

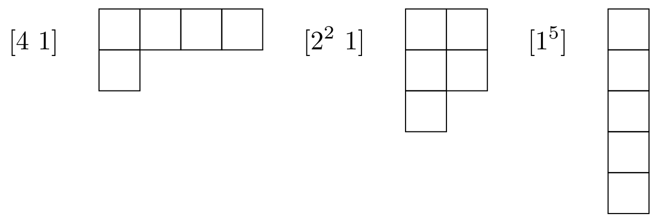
We first introduce some definitions and examples regarding Young tableaux, starting with the partition of a natural number n. Here our primary references for these terms will be [Ful96], [CM93], and [PT17].

Definition 1. *Let $n \in \mathbb{N}$. We call $[p_1\ p_2 \ldots p_m]$ a **partition of** n if $p_1 + p_2 + \cdots + p_m = n$ and $p_1, p_2, \ldots p_m \in \mathbb{N}$. For convenience, we will often write entries that appear with multiplicity greater than one in $[p_1\ p_2 \ldots p_m]$ using exponents.*

Partitions are considered distinct up to the order of the summands. For example, $1+2$ and $2+1$ are equivalent partitions of 3. Thus, all the partitions of 3 are [3], [2 1], and $[1^3]$. For any positive integer n, we can define a function $p(n)$ that is the number of distinct partitions of n. Determining the values of $p(n)$ can be done by listing these as done here for $n = 3$, and while there is no closed form expression for this function, it is given asymptotically as $p(n) = \dfrac{1}{4n\sqrt{3}} e^{\left(\pi\sqrt{\frac{2n}{3}}\right)}$ [Joh12]. See [Gri85] for more of the techniques used to study $p(n)$.

Definition 2. *A **Young diagram** is a visual representation of a partition of an integer n. The diagram is a left-justified set of n boxes organized in rows. Each row corresponds to a summand in* *the partition of n, and the length of the rows weakly decreases down the diagram.*

Example 1. Some Young diagrams for $n = 5$ are shown here with their corresponding partitions:

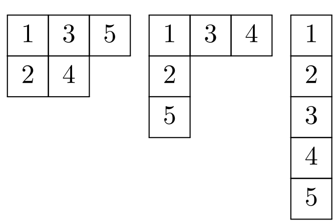


Definition 3. *A* ***Young tableau*** *is a Young diagram with an integer from* 1 *to* n *assigned to each box. For our purposes, there will be no repeated integers.*

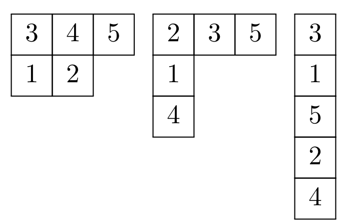
There are several types of Young tableaux, but we will mainly focus on standard and row-strict tableaux.

Definition 4. *A* ***standard tableau*** *is a Young tableau with integers that increase across rows and down columns. A* ***row-strict tableau*** *is a Young tableau in which integers increase across rows, but not necessarily down columns. Note that all standard tableaux are also row-strict.*

Example 2.



Some standard tableaux for $n = 5$



Some row-strict tableaux for $n = 5$ that are not standard

1.2 Extended Sets

In the work of Graham–Precup–Russell mentioned earlier, three sets I, J and K that can be read from a tableau have been established.[1] We will call these **extended sets**. Before we can define

[1] The definitions of these sets is in a portion of this research not publicly available but relayed to the author by Russell.

them, we will need to establish a bit of notation. Let T be a given Young tableau of size n. We will impose a labelling on the rows and columns, so that the leftmost column is Column 1 and the numbers increase by 1 from left to right, and the topmost row is Row 1, and the numbers increase by 1 from top to bottom. Then for any $x \in \{1, ..., n\}$, we use $\text{row}(x)$ to denote the number of the row containing the integer x in T, and we use $\text{col}(x)$ to denote the number of the row containing the integer x in T. We are now able to use this notation to define the extended sets for a tableau T.

Definition 5. *Let X_n be the set of integers $\{1, \ldots, n-1\}$. Define the extended sets I, J, and K for a given Young tableau as the following:*

$$I = \left\{ i \in X_n \middle| \begin{array}{l} \text{row}(i+1) < \text{row}(i) \textit{ and } \text{col}(i+1) = \text{col}(i) + 1; \\ \textit{or } \text{col}(i+1) > \text{col}(i) + 1 \end{array} \right\}$$

$$J = \{ j \in X_n | \text{row}(j+1) = \text{row}(j) \textit{ and } \text{col}(j+1) = \text{col}(j) + 1 \}$$

$$K = \left\{ k \in X_n \middle| \begin{array}{l} \text{col}(k+1) = \text{col}(k) + 1 \textit{ and } \text{row}(k+1) > \text{row}(k); \\ \textit{or } \text{col}(k+1) \leq \text{col}(k) \end{array} \right\}$$

Example 3. Here we see the extended sets for a given row-strict tableau of size $n = 10$.

1	5	6
2	3	10
4	8	
7		
9		

$I = \{4, 7, 9\}$ $\quad J = \{2, 5\}$ $\quad K = \{1, 3, 6, 8\}$

While it is a fairly straightforward task to construct the corresponding extended sets given a tableau, it is more challenging to determine whether a possible collection of extended sets corresponds to a valid tableau. Furthermore, if it does, can the tableau be built? In this paper, we explore this question, and in doing so obtain some bounds on the number of extended sets for a fixed n, put forth some general results, and outline solutions to our question for some special cases.

2 Number of Possible Extended Sets

Our first question regarding these extended sets is: How many unique extended sets correspond to all possible row-strict tableaux of size n? We start with an obvious upper bound for this value, which is the total number of possible extended sets. This bound is achieved by finding the number of unique divisions of the elements of X_n into 3 subsets. Therefore, our first upper bound is 3^{n-1}.

We now introduce a theorem that will help us set a lower bound on this number of extended sets.

Theorem 1. *Any collection of disjoint sets I, J, and K which partition $\{1, 2, \ldots, n-1\}$ and for which I is the empty set will be the extended sets for some valid row-strict tableau.*

Proof. Given disjoint sets, let I be empty. We will prove this corresponds to a valid row-strict tableau by construction. For any run $a, a+1, \ldots, b$ in J with $b+1 \in K$, we construct a row of the tableau of length such that the labels in the row are $a, a+1, \ldots, b, b+1$. Then we order these rows such that the length of the rows weakly decreases down the tableau. Any remaining elements of $\{1, \ldots, n-1\}$ should all be in K. Place each of these in a row of length one below the already-constructed rows. The order of these rows will not matter. Then every element x at the end of a row, including those elements in rows of length one, will be in K, since $\text{col}(x+1) = 1$ by construction, and therefore $\text{col}(x+1) \leq \text{col}(x)$. Also, every other element will be in J by construction and the definition of J. Then we have a valid row-strict tableau with no elements in I. □

Since we now know any possible extended sets with an empty I will always produce a valid tableau, the minimum number of extended sets for size n results from placing the elements $\{1, 2, \ldots, n-1\}$ into either J or K. Therefore, a lower bound on our value is 2^{n-1}.

We introduce another theorem to further narrow our bounds.

Theorem 2. *In any row-strict or standard tableau, the smallest element not in J must be in K.*

Proof. Let $1, 2, \ldots, a$ be in J and $x = a+1$, $x \geq 1$, $x \notin J$. Then by the definition of row-strict and the set J, $\text{row}(1) = \text{row}(2) = \ldots = \text{row}(a) = \text{row}(x)$, and $\text{col}(1) = 1, \text{col}(2) = 2, \ldots, \text{col}(a) = a, \text{col}(x) = x$. However, since $x \notin J$, $\text{row}(x+1) \neq \text{row}(x)$, so $\text{col}(x+1) = 1$ by the definition of row-strict. Then $\text{col}(x+1) \leq \text{col}(x)$, and therefore $x \in K$. □

This theorem allows us to exclude more possible extended sets and improve our upper bound. In order to count the number of possible extended sets to be excluded from our estimate, fix the elements $\{1, 2, \ldots, m\}$ in J, and then suppose $m+1 \in I$. When $m = n-2$, then $n-1 \in I$, so that accounts for $3^0 = 1$ possible extended set that can be excluded. When $m = n-3$, then $n-2 \in I$, and so $n-1$ can be in I, J, or K, which accounts for $3^1 = 3$ more possible extended sets to exclude. This continues until $m = 1$, which corresponds to 3^{n-3} excluded possible extended sets. Then lastly, we have the case when J is empty and $1 \in I$. This accounts for 3^{n-2} possibilities. Then the number of sets excluded due to Theorem 2 is

$$3^0 + 3^1 + \cdots + 3^{n-3} + 3^{n-2} = \sum_{i=0}^{n-2} 3^i.$$

All of the above discussion gives us a new upper bound.

Theorem 3. *Let n be an integer such that $n \geq 2$. Then there are at most*

$$\frac{3^{n-1}+1}{2}$$

extended sets corresponding to row-strict tableau of size n.

Proof. From counting the number of collections of three disjoint sets partitioning $\{1, \ldots, n-1\}$, we started with the bound 3^{n-1}. We can then remove those that we identified as impossible from Theorem 2 to get

$$3^{n-1} - \sum_{i=0}^{n-2} 3^i.$$

Since this is a geometric series, we have the compact formula of

$$\sum_{i=0}^{n-2} 3^i = \frac{3^{n-1}-1}{3-1} = \frac{3^{n-1}-1}{2}.$$

So we have

$$3^{n-1} - \frac{3^{n-1}-1}{2} = \frac{3^{n-1}+1}{2}.$$

□

We now present a table containing our lower bound, simple upper bound, improved upper bound, and actual number of unique extended sets for some relatively small values of n.

At this time, those are the only actual values for the number of valid extended sets known. We paused our computations here because it would take 90 hand computations of possible extended sets to calculate for $n = 6$ based on our bounds. Alternatively, we could use a program to compute all of the extended sets for all of the row-strict tableaux of size 6. However, there are 1,602 such row-strict tableaux that would need to be generated for such a computation, and this number would

n	Lower Bound	First UB	Improved UB	Actual Number of Unique Extended Sets
2	2	3	2	2
3	4	9	5	5
4	8	27	13	12
5	16	81	41	29

Figure 1: Bounds Versus Actual Number of Unique Extended Sets of Size n

grow quickly with n. Because of the connection to the partition function $p(n)$, the question of the asymptotic behaviour of the number of extended sets would be interesting to determine, but we did not explore that here.

3 Results for Row-strict and Standard Tableaux

Now that we have made some investigation into the number of these extended sets, we turn to our overall question of identifying properties that guarantee given extended sets correspond to a valid row-strict or standard tableaux. We were able to produce some general results in Theorem 1 and Theorem 2. Before we move on and present further results, let us first note some obvious properties of tableaux.

Property 1. *Let T be a row-strict tableau of size n and let $x \in X_n$. If* $\text{row}(x+1) = \text{row}(x)$*, then* $\text{col}(x+1) = \text{col}(x) + 1$.

This follows from the fact that in any row-strict tableau, the entries must increase across each row, so consecutive numbers in the same row must be adjacent.

Property 2. *Let T be a row-strict tableau. Then* $\text{col}(1) = 1$.

This again follows from the fact that in any row-strict tableau, entries increase across the rows, and 1 is the least element in our set X_n.

Property 3. *Let T be a standard tableau of size n and let $x \in X_n$. Then* $\text{col}(x+1) > \text{col}(x)$ *if and only if* $\text{row}(x+1) \leq \text{row}(x)$.

This property is a bit less obvious, so we will include a proof.

Proof. In order to see that this property holds, first let $\text{col}(x+1) > \text{col}(x)$ for an element $x \in X_n$ in a standard tableau T. If $\text{row}(x+1) > \text{row}(x)$, then for the elements of the tableau to increase across each row, there must be some element $y > x+1$ such that $\text{row}(y) = \text{row}(x)$, $\text{col}(y) = \text{col}(x+1)$, and row$(y) <$ row$(x+1)$. This implies that the elements of the tableau decrease down a column, which is a contradiction since the tableau is standard. Thus, row$(x+1) \leq$ row(x) as claimed.

Now take row$(x+1) \leq$ row(x), and again for a contradiction, suppose col$(x+1) \leq$ col(x). One possibility for this case is row$(x+1) <$ row(x) and col$(x+1) \leq$ col(x), in which case the elements of the tableau decrease down a column, a contradiction. Another possibility is row$(x+1) =$ row(x) and col$(x+1) <$ col(x), in which case the elements of the tableau decrease across a row, another contradiction. The final possibility is row$(x+1) =$ row(x) and col$(x+1) =$ col(x). But then x and $x+1$ occupy the same box in the tableau, a contradiction. □

With these properties established, we are now able to state some general results regarding extended sets corresponding to standard and row-strict tableau. Our first result posits a relationship between an element in the first row of a standard tableau and its placement in the corresponding extended sets.

Theorem 4. *Let x be an element in a standard tableau such that* row$(x) = 1$. *Then $x \notin I$.*

Proof. For a contradiction, let $x \in I$. In either condition for $x \in I$, col$(x+1) >$ col(x). However, according to Property 3 for standard tableaux, if col$(x+1) >$ col(x), then row$(x+1) \leq$ row(x). Since there is no row smaller than 1, the only option is row$(x+1) =$ row(x), in which case $x \in J$ by Property 1 and the definition of J. Then we have a contradiction, and $x \notin I$ if x is an element in Row 1 of a standard tableau. □

Our next theorem states the existence of a standard tableau of any shape and any size n such that I is empty in the corresponding extended sets and gives a way to build such a tableau.

Theorem 5. *There exists a specific labelling of a tableau of any shape that produces an empty I in the extended sets.*

Proof. Let T be a tableau of some fixed shape and let b be the length of the top row of T. Then fill the top row with $1, 2, \ldots, b$ in order. Place $b+1, b+2, \ldots$ in order in the second row, and continue filling that row in the same manner as Row 1. Continue this until the tableau is full. Then $1, 2, \ldots, b-1 \in J$ by definition, and $b \in K$ since col$(b+1) \leq$ col(b). By similar logic, every element before the last spot in each row is in J, and every element at the end of each row is in K. Then I is empty for every such labelling. □

Note that this theorem is similar to Theorem 1, but Theorem 1 begins with a given collection of possible extended sets with I empty and proves the existence of a corresponding row-strict tableau, while Theorem 5 concerns the specific labelling of a given Young diagram in order to produce a standard tableau with a corresponding empty I set.

The next theorem and a corollary that follows are concerned with "runs" of consecutive integers in I and how they relate to the shape and size of the corresponding Young tableau. Before we state and prove these, we must introduce some notation. Define $\langle i, j \rangle$ as a list of consecutive integers from i to j inclusive, where $i \leq j$. Also, define $|\langle i, j \rangle|$ as the number of consecutive integers in the run. In other words, $|\langle i, j \rangle| = j - i + 1$. We may now proceed with our results.

Theorem 6. *Let $\langle a, a+j \rangle$ be the longest run of consecutive integers in I for a row-strict tableau. Then the tableau must have at least $|\langle a, a+j \rangle| + 1$ rows and $|\langle a, a+j \rangle| + 1$ columns.*

Proof. To prove our statement, we will directly argue that if $\langle a, a+j \rangle$ is a run in I, then the labels $a, \ldots, a+j+1$ must be in distinct rows and columns of any row-strict tableau. We will use induction on the size of a run in I. We first consider the case of a run of integers in I that is size 1. Suppose this run is $\langle a \rangle$. If $a \in I$ by the first condition, then $\text{row}(a+1) < \text{row}(a)$ and $\text{col}(a+1) = \text{col}(a)+1$, in which case a and $a+1$ are in distinct rows and columns as needed. If $a \in I$ by the second condition, then $\text{col}(a+1) > \text{col}(a)+1$. Let $\text{row}(a+1) = \text{row}(a)$. Then by Property 1 of row-strict and standard tableaux, $\text{col}(a+1) = \text{col}(a)+1$, in which case $a \in J$, a contradiction. So $\text{row}(a+1) \neq \text{row}(a)$, and again a and $a+1$ are in distinct rows and columns.

Now, for our induction hypothesis, suppose if $\langle a, a+j-1 \rangle$ is a run in I for some $a \in X_n$ and $j \geq 1$ then $a, \ldots, a+j$ are in distinct rows and columns of any row-strict tableau. Consider now a run in I of length $j+1$. We will suppose then it is of the form $\langle a, a+j \rangle$ for some $a \in X_n$. By our induction hypothesis, $a, \ldots, a+j$ are in distinct rows and columns of the tableau, so we need only argue that $a+j+1$ is in a distinct row and column. First, since $a, \ldots, a+j \in I$, we know $\text{col}(a+j+1) > \text{col}(a+j) > \cdots > \text{col}(a)$. So we know quickly that $a+j+1$ is in a distinct column. Now, suppose $\text{row}(a+j+1) = \text{row}(a+k)$ for some $0 \leq k \leq j-1$. (We will consider the case of $k = j$ separately.) Then $\text{col}(a+j+1) = \text{col}(a+k)+1$ because the tableau is row-strict and all possible y such that $a+k < y < a+j+1$ are in other rows according to our induction hypothesis. However, we know $\text{col}(a+j+1) > \text{col}(a+j)$ since $a+j \in I$. Together $\text{col}(a+j+1) \geq \text{col}(a+j)+1$ and $\text{col}(a+j+1) = \text{col}(a+k)+1$ for $k \leq j-1$ means $\text{col}(a+k) \geq \text{col}(a+j)$. This contradicts $\text{col}(a+j+1) > \text{col}(a+j) > \cdots > \text{col}(a)$, so we know it cannot be true. This only leaves the possibility that $\text{row}(a+j) = \text{row}(a+j+1)$. If this is true, we must have $\text{col}(a+j+1) > \text{col}(a+j)+1$ since $a+j \in I$. However, this contradicts Property 1. Thus, each element $a, \ldots, a+j+1$ is in a distinct row and column, and the claim follows from induction.

□

While Theorem 6 is concerned with the shape of the resulting Young tableau given a certain condition, Corollary 1 puts forth a more general result regarding the size n of the tableau given the same condition.

Corollary 1. *If there are k consecutive integers in I, then $n \geq \dfrac{(k+1)(k+2)}{2}$.*

Proof. If there are k consecutive integers in I, there are $k+1$ elements each in a distinct column and distinct row, as shown by Theorem 6. Then a new consecutive integer in I will add at least one new column and one new row to the tableau. The first condition for $x \in I$, which is $\text{row}(i+1) < \text{row}(i)$ and $\text{col}(i+1) = \text{col}(i)+1$, will produce the shape with the smallest n since a new consecutive integer will only add 1 new row and 1 new column. Adding new consecutive integers $a, a+1, a+2, \ldots \in I$ in this fashion gives a "stair-step" shape that corresponds to the smallest possible n. This leads to a tableau with a top row of length $k+1$, second row of k, and so on for $k+1$ rows. Because this is optimal, $n \geq 1+2+3+\ldots+k+1 = \dfrac{(k+1)(k+2)}{2}$.

□

Using Corollary 1, we are able to immediately dismiss certain possible extended sets as not corresponding to any valid standard or row-strict tableau given their I set. More specifically, if there are k consecutive integers in I for possible extended sets, but $n < \dfrac{(k+1)(k+2)}{2}$, we know these do not correspond to a valid tableau, no matter the shape or labelling.

4 Hook Shape Tableaux

We now move into discussion on hook shape tableaux and answer our overarching question of which extended sets correspond to valid row-strict and standard tableaux in this specific case. In order to do so, we must first define hook shape.

Definition 6. *A tableau T is a **hook shape** tableau if the partition of n corresponding to the tableau is of the form $[a\ 1^b]$, where $a \geq 1$, $b \geq 0$. In other words, every element x in a tableau of hook shape is such that either $\text{row}(x) = 1$ and $\text{col}(x) \geq 1$ or $\text{col}(x) = 1$ and $\text{row}(x) \geq 1$.*

Before we are able to define the structure of extended sets that give us a valid hook shape tableau, we must introduce a new theorem.

Theorem 7. *For a row-strict tableau of hook shape, $x \in K$ if and only if $\text{col}(x+1) = 1$.*

Proof. Suppose first that an element $x \in X_n$ in in K for a row-strict tableau of hook shape. Then since there is only one column and one row with more than one element in a hook shape tableau, the condition for $x \in K$ must be the second, which is $\text{col}(x+1) \leq \text{col}(x)$. If $\text{col}(x+1) = \text{col}(x)$, then both $\text{col}(x) = 1$ and $\text{col}(x+1) = 1$ because Column 1 is the only column with more than one element. If $\text{col}(x+1) < \text{col}(x)$, then $\text{col}(x) \neq 1$ because, if it was, then there would be no column such that $\text{col}(x+1) < \text{col}(x)$ since columns are labeled with positive integers. Since $\text{col}(x) \neq 1$, then $\text{row}(x) = 1$, $\text{col}(x) \geq 2$. Suppose $\text{row}(x+1) = 1$. Then, for the tableau to be row-strict, $\text{col}(x+1) = \text{col}(x) + 1$ by Property 1, which puts $x \in J$, a contradiction to the fact that $x \in K$. Then $\text{row}(x+1) \neq 1$, and therefore $\text{row}(x) > 1$, which means $\text{col}(x) = 1$ since it's the only column with more than one row.

Now we can prove the converse statement. Suppose $x \in X_n$ and $\text{col}(x+1) = 1$ for a row-strict tableau. Then either $\text{col}(x) = 1$ as well, or $\text{col}(x) > 1$. In either case, $\text{col}(x+1) \leq \text{col}(x)$, so $x \in K$ by definition. □

We also introduce one last bit of notation helpful for our full theorem.

Definition 7. *Let T be a tableau of size n. We will say an element $x \in \{1, 2, \ldots, n\}$ in this tableau T is in **standard position** if $\text{row}(x) = 1$ and $\text{col}(x) = 1$.*

We are now finally able to discuss which extended sets produce a valid standard or row-strict tableau of hook shape, and in turn how to build such a tableau from given extended sets. Two cases should be considered. The first is standard tableaux and row-strict tableaux with 1 in standard position. The second is row-strict tableaux with 1 not in standard position. We introduce the following forms of the possible extended sets for these two cases:

Form 1. $K = \{\langle a_0, b_1\rangle, \langle a_1, b_2\rangle, \langle a_2, b_3\rangle, \ldots, \langle a_{i-1}, b_i\rangle\}$ *where* $a_i > b_i + 1$ *and* $a_j \leq b_{j+1}$

$I = \{b_1 + 1, b_2 + 1, b_3 + 1, \ldots, b_i + 1\}$

$J = X_n \backslash (K \cup I)$

Form 2. $K = \{\langle 1, b_1\rangle, \langle a_1, b_2\rangle, \langle a_2, b_3\rangle, \ldots, \langle a_{i-1}, b_i\rangle\}$ *where* $a_i > b_i + 1$ *and* $a_j \leq b_{j+1}$

$I = \{b_2 + 1, b_3 + 1, \ldots, b_i + 1\}$

$J = X_n \backslash (K \cup I)$

Theorem 8. *Let T be a row-strict tableau of hook shape with 1 in standard position. Then the structure of the extended sets is of Form 1.*

Proof. Let $K = \{\langle a_0, b_1\rangle, \langle a_1, b_2\rangle, \langle a_2, b_3\rangle, \ldots, \langle a_{i-1}, b_i\rangle\}$ in a standard or row-strict tableau of hook shape, with $a_i > b_i+1$. Any extended sets will have a K set of this form. Then $\text{col}(\{b_1+1, b_2+1, b_3+1, \ldots, b_i+1\}) = 1$ by Theorem 7 and $\text{row}(\{b_1+1, b_2+1, b_3+1, \ldots, b_i+1\}) > 1$ since 1 is in standard position. Since $\{b_1+1, b_2+1, b_3+1, \ldots, b_i+1\} \notin K$, $\{b_1+2, b_2+2, b_3+2, \ldots, b_i+2\}$ are not in Column 1 and therefore $\text{row}(\{b_1+2, b_2+2, b_3+2, \ldots, b_i+2\}) = 1$ and $\text{col}(\{b_1+2, b_2+2, b_3+2, \ldots, b_i+2\}) > 1$. It follows that for each b_i, either $\text{row}(b_i+2) < \text{row}(b_i+1)$ and $\text{col}(b_i+2) = \text{col}(b_i+1)+1$ or $\text{col}(b_i+2) > \text{col}(b_i+1)+1$, so by definition, $\{b_1+1, b_2+1, b_3+1, \ldots, b_i+1\} \in I$.

Now choose some element x such that $x \notin K$ and x is not in the I set outlined above, and let $\text{row}(x) > 1$, so $\text{col}(x) = 1$. In this case, one option is $\text{col}(x+1) = 1$, in which case $\text{col}(x+1) = \text{col}(x)$, and $x \in K$ by definition, which is contradiction. The other option for $x+1$ is $\text{col}(x+1) > 1$, which means $\text{row}(x+1) = 1$ by definition of hook shape. Then $x \in I$ by definition, another contradiction. So $\text{row}(x) = 1$. If $\text{row}(x+1) > 1$ and therefore $\text{col}(x+1) = 1$, then $x \in K$ by 7, another contradiction. Then $\text{row}(x+1) = 1$, and by 1, $\text{col}(x+1) = \text{col}(x)+1$, so $x \in J$ by definition.

□

Theorem 9. *The extended sets that correspond to a row-strict tableau of hook shape with 1 not in standard position can take either Form 1 or Form 2.*

Proof. Note that $1 \notin I$ by Theorem 2. If 1 is not in standard position, then $1 \notin J$ as well. Thus, $1 \in K$. Then the only other difference between these two forms is the location of b_1+1 in the extended sets. The first form is the case where $b_1+1 \notin J$, which is of the same form as Theorem 8 and follows a similar argument for its proof. The second case occurs when $b_1+1 \in J$, In this case, b_1+1 is in Column 1 by Theorem 7. Also, the condition for $b_1+1 \in J$ is $\text{row}(b_1+2) = \text{row}(b_1+1)$ and $\text{col}(b_1+2) = \text{col}(b_1+1)+1$, and the only element in Column 1 and also in a row with length greater than 1 is in Row 1. Therefore, for Case 2, $b_2+1, b_3+1, \ldots, b_i+1 \in I$ follows the same logic as outlined in the proof for Theorem 8, except b_1+1 is in standard position instead of 1. □

To build a hook shape tableau of Theorem 8 type with valid extended sets of the form above, place 1 in standard position. Then place the elements $x+1$ such that $x \in K$ in Column 1, in ascending order down the column if the tableau is standard, or in any order if the tableau is row-strict. Then place the remaining elements in ascending order in Row 1.

To build a tableau of the form outlined in Theorem 9, first place the element b_1+1 in standard position if the extended sets follow the second case, i.e. if $b_1+1 \in J$. If $b_1+1 \in I$, place another element $x+1$ such that $x \in K$ in standard position. Then build the tableau in the same manner as above, placing the elements in Column 1 in any order since a tableau of hook shape with 1 not in standard position will never be standard.

Example 1. The following is a hook shape, row-strict tableau of size $n = 12$ and Theorem 9, Case 2 type and its corresponding extended sets.

5	6	8	10	11	12
3					
1					
4					
7					
9					
2					

$I = \{7, 9\}$ $\quad J = \{5, 10, 11\}$ $\quad K = \{1, 2, 3, 4, 6, 8\}$

5 Two-Row Tableaux

We now answer our questions regarding valid extended sets and construction from these extended sets for another special type of tableau: two-row, standard tableaux. We begin by defining these tableaux.

Definition 8. *A tableau T is a **two-row** tableau if the partition of n corresponding to the tableau is of the form $[a\ b]$, where $a, b \geq 1$.*

We now restate an old theorem and put forth a new theorem that relates an element's position in a two-row, standard tableau and its position in the corresponding extended sets. While these theorems are concerned mainly with construction of a tableau, they will be useful later in proving theorems regarding the format of the extended sets for two-row, standard tableaux. First, recall the statement of Theorem 4:

Let x be an element in T, a standard tableau. If $\text{row}(x) = 1$, then $x \notin I$.

The contrapositive of this theorem applied to the two-row case produces a perhaps more useful result for our goal. We'll state this as a corollary to Theorem 4.

Corollary 2. *If $x \in I$ in a two-row, standard tableau, then $\text{row}(x) \neq 1$, so $\text{row}(x) = 2$.*

The next theorem is of a similar form to Theorem 4.

Theorem 10. *Let x be an element in T, a standard tableau with two rows. If* $\text{row}(x) = 2$*, then* $x \notin K$.

Proof. Let $\text{row}(x) = 2$. The first condition for $x \in K$ is $\text{col}(x+1) = \text{col}(x) + 1$ and $\text{row}(x+1) > \text{row}(x)$, but there is no row greater than 2, so this condition fails. Furthermore, this means $\text{row}(x+1) \leq \text{row}(x)$ The second condition for $x \in K$ is $\text{col}(x+1) \leq \text{col}(x)$. However, by Property 3 of standard tableaux, $\text{row}(x+1) \leq \text{row}(x)$ if and only if $\text{col}(x+1) > \text{col}(x)$, so this condition fails as well. Then $x \notin K$. □

Again, the contrapositive of this theorem gives another useful result: If $x \in K$, then $\text{row}(x) \neq 2$, so $\text{row}(x) = 1$.

Using these results, we can now introduce three new theorems that define the format that the extended sets for a valid two-row, standard tableau must take.

Theorem 11. *A two-row, standard tableau has no consecutive integers in I or K.*

Proof. Let x be an element in I. Then by the contrapositive of Theorem 4 above, $\text{row}(x) \neq 1$, so $\text{row}(x) = 2$. If $\text{row}(x+1) = \text{row}(x)$, then $\text{col}(x+1) = \text{col}(x)+1$ by Property 1 of standard tableaux, in which case $x \in J$, which is a contradiction. Then $\text{row}(x+1) \neq \text{row}(x)$, so $\text{row}(x+1) = 1$. Then by Theorem 4, $x+1 \notin I$.

Let x be an element in K. Then by the contrapositive of Theorem 10 above, $\text{row}(x) \neq 2$, so $\text{row}(x) = 1$. If $\text{row}(x+1) = \text{row}(x)$, then $\text{col}(x+1) = \text{col}(x)+1$ by Property 1 of standard tableaux, in which case $x \in J$, which is a contradiction. Then $\text{row}(x+1) \neq \text{row}(x)$, so $\text{row}(x+1) = 2$. Then by Theorem 10, $x+1 \notin K$. □

The next two theorems further specify the format of the extended sets in this case.

Theorem 12. *Let T be a 2-row, standard tableau and x be an element in I. If there is a run $\langle x+1, b\rangle \in J$, where $x+1 \leq b$, then $b+1 \in K$. Otherwise, $x+1 \in K$.*

Proof. Let $x \in I$. Then by the contrapositive of Theorem 4, $\text{row}(x) \neq 1$, so $\text{row}(x) = 2$. If $\text{row}(\langle x+1, b\rangle) = 2$, then $x \in J$ because $\text{row}(x+1) = \text{row}(x)$ and $\text{col}(x+1) = \text{col}(x) + 1$ since the tableau is standard, which is a contradiction, so $\text{row}(\langle x+1, b\rangle) = 1$. Then $\text{row}(b+1) = 1$ by the definition of J. Thus, by Theorem 4 above, $b+1 \notin I$, and $b+1 \notin J$, so $b+1 \in K$.

If no such run $\langle x+1, b\rangle \in J$ exists, then $x+1 \notin J$. Additionally, $x+1 \notin I$ by Theorem 11 above. Therefore, $x+1 \in K$. □

Theorem 13 and its proof are of a similar form to Theorem 12 and its proof.

Theorem 13. *Let T be a 2-row, standard tableau and x be an element in K. If there is a run $\langle x+1, b\rangle \in J$, where $x+1 \leq b$, then $b+1 \in I$. Otherwise, $x+1 \in I$.*

Proof. Let $x \in K$. Then by the contrapositive of Theorem 10, $\text{row}(x) \neq 2$, so $\text{row}(x) = 1$. If $\text{row}(\langle x+1, b\rangle) = 1$, then $x \in J$ because $\text{row}(x+1) = \text{row}(x)$ and $\text{col}(x+1) = \text{col}(x)+1$ since the tableau is standard, which is a contradiction, so $\text{row}(\langle x+1, b\rangle) = 2$. Then $\text{row}(b+1) = 2$ by the definition of J. Thus, by Theorem 10 above, $b+1 \notin K$, and $b+1 \notin J$, so $b+1 \in I$.

If no such run $\langle x+1, b\rangle \in J$ exists, then $x+1 \notin J$. Additionally, $x+1 \notin K$ by Theorem 11 above. Therefore, $x+1 \in I$. □

Our last theorem gives a result regarding the construction of a two-row, standard tableau. By construction, we mean creating a Young tableau by placing the numbers $1, 2, \ldots, n$ into a Young diagram in increasing order.

Theorem 14. *The bottom row of a two-row standard tableau will never exceed the length of the top row during construction of the tableau.*

Proof. Take a standard, two-row tableau partway through its construction in which the bottom row is longer than the top row. Then there exists an element x in the top row such that $x+1$ is in the bottom row, and $\text{col}(x+1) > \text{col}(x)$. Since a valid, completed tableau must have a top row with a length greater than or equal to that of the bottom row, at some point during construction, an element y such that $y > x+1$ must be placed in the top row such that $\text{col}(y) = \text{col}(x+1)$. Then the tableau decreases down columns, which means it's standard, and we have a contradiction, so the bottom row of a two-row standard tableau can never exceed the length of the top row during construction. □

If the conditions given in Theorems 11-13 are met, we may begin to try to construct a two-row, standard tableau. Start by placing 1 in Column 1, Row 1 since the tableau is standard. Continue across each row with the elements $2, 3, \ldots, n$ by placing the element in Row 2 if the element is in I in the extended sets and in Row 1 if the element is in K in the extended sets (by Theorems 4 and 10). Continue with this pattern. If ever the length of the bottom row exceeds the length of the top row, then the given extended sets are invalid and no standard tableau can be constructed by 14.

Example 1. The following is a two-row, standard tableau of size $n = 15$, along with its corresponding extended sets. Note that the format of the extended sets follows the conditions set by Theorems 11-13 and that the positions of the elements in the tableau follow from Theorems 4 and 10. Also note that Theorem 14 holds as the tableau is being constructed.

1	2	5	6	9	10	12	14	15
3	4	7	8	11	13			

$I = \{4, 8, 11, 13\}$ $J = \{1, 3, 5, 7, 9, 14\}$ $K = \{2, 6, 10, 12\}$

6 Programs

For efficiency purposes, as part of this research we've built several Java programs that deal with Young tableau and extended sets.

setsijk.java: This program first prompts the user to enter n and also the name of a text file containing a Young tableau. Arrays for the row value and column value of each number $1, 2, \ldots, n$ are initialized, as are counts for the column and row numbers. The program then parses through the text file, first reading in each row and then reading in each number in that row. For each number, the corresponding value for row and column number are updated at that position in the two arrays. Column count is increased when a new number is read in and then reset back to 0 when the end of the row is reached, at which point row count is increased. The program then uses the row and column value arrays and the conditions outlined in Definition 5 in a simple if-else statement in order to place each element of the tableau into arrays that correspond to the extended sets. Finally, the contents of these arrays that represent the extended sets are printed out to the screen. This program has an $O(n)$ run time.

```
import java.io.*;
import java.util.*;

class setsijk
{
    public static void main(String [] args)
    throws IOException
    {
            int nextEntry=0;
            int n;
            String filename;
```

```java
String readrow="";

//read in text file with Young tableau
Scanner cin = new Scanner(System.in);
System.out.print("What is the name of your text file? ");
filename = cin.nextLine();
Scanner f = new Scanner(new FileReader(filename));

System.out.print("Enter n: ");
n=cin.nextInt();

//create arrays that will hold which row and column each element is in
int [] rowVal = new int[n];
int [] colVal = new int[n];
int colCount=0;
int rowCount=0;

int a;
//begin parsing through text file
while(f.hasNextLine())
{
        readrow=f.nextLine();

        Scanner s = new Scanner(readrow);
        colCount=0;
        //assign row and column values to each element based on
        //positions in tableau
        while(s.hasNext())
        {
                a=s.nextInt();
                rowVal[a-1]=rowCount;
                colVal[a-1]=colCount;
                colCount++;
```

```
        }
        rowCount++;
}

//create arrays that will hold extended sets
int [] iSet=new int[n];
int [] jSet=new int[n];
int [] kSet=new int[n];
int iCount=0;
int jCount=0;
int kCount=0;

//put each element x in extended sets based on position of x+1
//Note that if-else statements correspond exactly to Definition 5
for(int j=0; j<(n-1); j++)
{
        if((colVal[j+1]==(colVal[j]+1) && rowVal[j+1]<rowVal[j]) ||
        (colVal[j+1] > colVal[j]+1))
        {
                iSet[iCount]=(j+1);
                iCount++;
        }
        else if((rowVal[j+1]==rowVal[j]) && (colVal[j+1]==(colVal[j]+1)))
        {
                jSet[jCount]=(j+1);
                jCount++;
        }
        else
        {
                kSet[kCount]=(j+1);
                kCount++;
        }
}
```

```
            //print elements in extended sets out to screen
            System.out.print("I: ");
            for(int i=0; i<iCount; i++)
            {
                    System.out.print(iSet[i]+" ");
            }
            System.out.print("\n"+"J: ");
            for(int j=0; j<jCount; j++)
            {
                    System.out.print(jSet[j]+" ");
            }
            System.out.print("\n"+"K: ");
            for(int k=0; k<kCount; k++)
            {
                    System.out.print(kSet[k]+" ");
            }
        System.out.println("");
    }
}
```

hookbuilder.java: This program again prompts the user to enter n and the name of a text file, this time containing extended sets. The program creates a set array and then goes through the text file and assigns each element a value in the array based on if it's in I, J, or K. Then each condition for extended sets corresponding to hook shape tableaux is checked. First, if the first element of the set array is in I, then the program quits and an error message is printed out. Then, we cycle through the start of the set array and use a simple variable to check if the first element not in J is in K, as per Theorem 2, and print an error message and quit if it's not. We use a similar system in a serious of if-else statements in order to check the conditions of forms 1 and 2 and if they don't match, an error message is printed and the program quits. Finally, arrays for the top row and first column are initialized, and again using forms 1 and 2, along with Theorem 7, the elements are placed in the correct array. Finally, using these arrays, the tableau is printed out to the screen along with if the tableau is standard or row-strict. This program also has an $O(n)$ run time.

```java
import java.io.*;
import java.util.*;

class hookbuilder
{
	public static void main(String [] args)
	throws IOException
	{
		String filename;
		String readrow;
		int setCount;
		int n;
		boolean loopVar=true; //set to false to break loop if theorems
		//broken

		//read in text file with extended sets
		Scanner cin = new Scanner(System.in);
		System.out.print("What is the name of your text file? ");
		filename = cin.nextLine();
		Scanner f = new Scanner(new FileReader(filename));

		System.out.print("Enter n: ");
		n=cin.nextInt();

		int [] set = new int[n-1];

		int a;
		setCount=1;
		//parse through text file
		while(f.hasNextLine())
		{
			readrow=f.nextLine();
```

```
        Scanner s = new Scanner(readrow);

        //each element's value in set array is which of extended sets
        //it's in 1 for I, 2 for J, 3 for K
        while(s.hasNext())
        {
                a=s.nextInt();
                set[a-1]=setCount;
        }
        setCount++;
}

//This comes from a Corollary of Theorem 2
if(set[0]==1)
{
        System.out.println("This is not a valid tableau because 1
        is in I.");
        loopVar=false;
}

int placeholder=0;
//this loop finds smallest element not in J
while(set[placeholder]==2 && loopVar==true)
{
        placeholder++;
}
//This comes from Theorem 2
if(set[placeholder]!=3)
{
        loopVar=false;
        System.out.println("This is not a valid tableau because the
        smallest element not in J is not in K.");
}
```

```
int counter=0; //keeps track of set array index
int var=0; //tracks which set previous element was in
int RSvar=0; //switches "on" after firstJ is found
int firstJ=0; //if row-strict, keeps track of element in standard
//position

while(counter<(n-1) && loopVar==true)
{
        if(set[counter]==1) //if element in I
        {
                if(var==3) //if previous element in K, update previous
                //and switch RSvar on
                {
                        var=1;
                        RSvar=1;
                }
                else
                {
                        //follows from Forms 1 and 2
                        loopVar=false;
                        System.out.println("This is not a valid
                        tableau because the next element after a run
                        in K is not in I.");
                }
        }
        else if(set[counter]==2) //if element in J
        {
                if(var!=3) //if previous element not in K, update
                //previous and switch RSvar on
                {
                        var=2;
                        RSvar=1;
```

```
            }
            else if(var==3 && RSvar==0) //if previous element in
            //K and RSvar hasn't been switched on, save firstJ
            {
                firstJ=counter+1;
                RSvar=1;
                var=2;
            }
            else
            {
                //follows from Forms 1 and 2
                loopVar=false;
                System.out.println("This is not a valid
                tableau because the next element after a
                run in K is not in I.");
            }
        }
        else if(set[counter]==3) //if element in K, update previous
        {
            var=3;
        }
        counter++;
    }

    //create arrays that will help us print
    int [] toprow = new int[n-1];
    int [] column = new int[n-1];
    int colLength=0;
    int rowLength=1;

    if(loopVar==true) //if theorems have held up
    {
        if(firstJ!=0) //Form 2: set element in standard position
```

```
//to firstJ
{
        toprow[0]=firstJ;
        column[0]=1;
        colLength=1;
}
else //if not, Form 1
{
        toprow[0]=1;
}
for(int i=0; i<(n-1); i++)
{
        if(set[i]==3) //if element is in K
        {
                if((i+2)!=firstJ) //besides firstJ
                {
                        column[colLength]=(i+2); //add to
                        //column: corresponds to Theorem 7
                        colLength++;
                }
        }
        else //if element is not in K
        {
                toprow[rowLength]=(i+2); //add to top row
                rowLength++;
        }
}

//prints out Row 1, then Column 1
for(int j=0; j<rowLength; j++)
{
        System.out.print(toprow[j]+" ");
}
```

```
                System.out.println("");
                for(int k=0; k<colLength; k++)
                {
                        System.out.println(column[k]);
                }
                //tell if tableau is standard or row-strict based on firstJ
                if(firstJ!=0)
                        System.out.println("This tableau is row-strict, so
                        all the elements in Column 1 besides " + firstJ +
                        " are permutable.");
                else
                        System.out.println("This is a standard tableau.");
            }
        }
}
```

tworowbuilder.java: This final program, like **hookbuilder.java**, takes in a filename containing I, J, K sets of the same form and n from the user. Again, we check to see if 1 is in I and if the smallest element not in J is not in K, and if either is true we print out a message and quit. Then the conditions outlined in Theorems 11-13 are checked using a while loop and if-else statements, and if the condition fails, an error is printed and we quit. We then begin construction of the tableau and use more if-else statements and the earlier set array to check whether the bottom row ever gets longer than the top row during construction, and if it does, we again print an error and quit. Finally, if we haven't yet quit, the program creates top and bottom row arrays and fills them based on the results of the previous construction loop. The arrays are printed out to the screen in the form of a tableau. This program has an $O(n)$ run time as well.

```
import java.io.*;
import java.util.*;

class tworowbuilder
{
        public static void main(String [] args)
        throws IOException
```

```java
{
        String filename;
        String readrow;
        int setCount;
        int n;
        boolean loopVar=true; //set to false to break loop if theorems broken

        //read in text file with extended sets
        Scanner cin = new Scanner(System.in);
        System.out.print("What is the name of your text file? ");
        filename = cin.nextLine();
        Scanner f = new Scanner(new FileReader(filename));

        System.out.print("Enter n: ");
        n=cin.nextInt();

        int [] set = new int[n-1];

        int a;
        setCount=1;
        //parse through text file
        while(f.hasNextLine())
        {
                readrow=f.nextLine();

                Scanner s = new Scanner(readrow);

                //each element's value in set array is which of extended sets
                //it's in 1 for I, 2 for J, 3 for K
                while(s.hasNext())
                {
                        a=s.nextInt();
                        set[a-1]=setCount;
```

```
        }
        setCount++;
}

//This comes from a Corollary of Theorem 2
if(set[0]==1)
{
        System.out.println("This is not a valid tableau because 1
        is in I.");
        loopVar=false;
}

int placeholder=0;
//this loop finds smallest element not in J
while(set[placeholder]==2 && loopVar==true)
{
        placeholder++;
}
//This comes from Theorem 2
if(set[placeholder]!=3)
{
        loopVar=false;
        System.out.println("This is not a valid tableau because the
        smallest element not in J is not in K.");
}

int counter=0; //keeps track of set array index
int var=0; //tracks index of pattern array
int [] pattern = new int[n-1]; //

while(counter<(n-1) && loopVar==true)
{
        if(set[counter]==1) //if element in I, enter 1 in array
```

```
		{
			pattern[var]=1;
			var++;
		}
		else if(set[counter]==3) //if element in K, enter 2 in array
		{
			pattern[var]=2;
			var++;
		}
		if(var>1 && pattern[var-2]==pattern[var-1]) //if consecutive
		//non-J elements are in same set, breaks Theorems 12-13
		{
			loopVar=false;
			System.out.println("This is not a valid tableau
			because it does not follow the alternating pattern.");
		}
		counter++;
	}

	//create array that tracks if element is in top or bottom row
	int [] tob = new int[n];
	int topCount=0;
	int bottomCount=0;
	counter=0;
	int tracker=1; //tracks which row previous non-J element was in

	while(counter<(n-1) && loopVar==true)
	{
		if(set[counter]==1) //if element in I
		{
			tob[counter]=2; //element is in bottom row: Corollary 2
			bottomCount++;
			tracker=1;
```

```
        }
        else if(set[counter]==2) //if element in J: use parts of
        //Theorems 12-13
        {
                if(tracker==1) //and if previous non-J element was in I
                {
                        tob[counter]=1; //element is in top row
                        topCount++;
                }
                else if(tracker==2) //or if previous non-J element was
                //in K
                {
                        tob[counter]=2; //element is in bottom row
                        bottomCount++;
                }
        }
        else if(set[counter]==3) //if element in K
        {
                tob[counter]=1; //element is in top row: Theorem 10
                topCount++;
                tracker=2;
        }
        if(bottomCount>topCount) //corresponds to Theorem 14
        {
                loopVar=false;
                System.out.println("This is not a valid tableau because
                the bottom row becomes longer than the top row.");
        }
        counter++;
}

//does previous loop but for last element
//because last element isn't in original extended sets
```

```
if(loopVar==true)
{
        if(tracker==1)
        {
                tob[counter]=1;
                topCount++;
        }
        else if(tracker==2)
        {
                tob[counter]=2;
                bottomCount++;
        }
        if(bottomCount>topCount)
        {
                loopVar=false;
                System.out.println("This is not a valid tableau because
                the bottom row becomes longer than the top row.");
        }
}

//create arrays for top and bottom rows
int [] topRow = new int[topCount];
int [] bottomRow = new int[bottomCount];
int topPlace=0;
int bottomPlace=0;
if(loopVar==true) //if all theorems have held up
{
        //this loop separates elements into two arrays based on
        //previous tob array
        for(int i=0; i<n; i++)
        {
                if(tob[i]==1)
                {
```

```
                                topRow[topPlace]=(i+1);
                                topPlace++;
                        }
                        else if(tob[i]==2)
                        {
                                bottomRow[bottomPlace]=(i+1);
                                bottomPlace++;
                        }
                }

                //print out top and bottom row to screen
                for(int j=0; j<topCount; j++)
                {
                        System.out.print(topRow[j]+" ");
                }
                System.out.println("");
                for(int k=0; k<bottomCount; k++)
                {
                        System.out.print(bottomRow[k]+" ");
                }
                System.out.println("");
        }
    }
}
```

7 Conclusion

In this research regarding Young tableaux and their extended sets, we've been able to produce some general results for standard and row-strict tableaux and made counting arguments for the number of possible extended sets for a tableau size n. We've also been able to answer our overarching question in some special cases by specifying the format of extended sets for valid hook shape and two-row tableaux, and in turn discussed how to build such tableaux.

The main question of being able to tell if extended sets correspond to any valid row-strict tableaux is still unanswered, a lot of which has to do with the large number of possible unique sets, along with the number of possible configurations of a tableau of a given size. Some progress has been made in solving the two-row, row-strict case, and an algorithm has been proposed for building a standard tableau given extended sets we know are valid. However, our overarching problem remains unsolved and requires further research.

References

[CM93] David H. Collingwood and William M. McGovern. *Nilpotent Orbits in Semisimple Lie Algebras.* Van Nostrand Reinhold, New York, 1993.

[Ful96] William Fulton. *Young Tableaux: With Applications to Representation Theory and Geometry.* London Mathematical Society Student Texts. Cambridge University Press, 1996.

[GPR20] William Graham, Martha Precup, and Amber Russell. A new approach to the generalized springer correspondence, 2020. arXiv:2002.12480.

[Gra19] William Graham. Toric varieties and a generalization of the Springer resolution, 2019. arXiv:1911.09601.

[Gri85] Ralph P. Grimaldi. *Discrete and Combinatorial Mathematics; An Applied Introduction.* Addison-Wesley Longman Publishing Co., Inc., USA, 1985.

[Hum95] James E. Humphreys. *Conjugacy Classes in Semisimple Algebraic Groups.* American Mathematical Society, USA, 1995.

[Joh12] Fredrik Johansson. Efficient implementation of the hardy–ramanujan–rademacher formula. *LMS Journal of Computation and Mathematics*, 15:341–359, oct 2012.

[PT17] Martha Precup and Julianna Tymoczko. Springer fibers and schubert points. *European Journal of Combinatorics*, 76, 01 2017.

[Rus20] Amber Russell. Graham's variety and perverse sheaves on the nilpotent cone. *Journal of Algebra*, 557, 9 2020.