A REPORT ON THE ROBINSON SCHENSTED ALGORITHM, TABLEAUX, KNUTH RELATIONS, AND VIENNOT'S GEOMETRIC CONSTRUCTION

WILLIAM WEHRING

Contents

A note on notation, we use the symbol ■ to denote the end of an example, the symbol ▨ to indicate the end of a definition, and finally the symbol □ to denote the end of a proof.

1. Introduction

In this report, we discuss the fundamentals of the Robinson Schensted (RS) Algorithm and its relationship to increasing subsequences of a finite permutation. We first discuss how to create a tableau, $P(\pi)$, using the RS-algorithm, given a permutation π on the set $[n] = \{1, 2, 3, \cdots, n\}$. From there, we discuss how we can use the shape of the tableau $P(\pi)$ to understand the length of increasing and decreasing subsequences of π. Next, we introduce a few equivalence relations on the set of permutations of the set $[n]$. These concepts lead to the notions of *Knuth relations* and *Knuth equivalence* given a permutation π, which in turn raise some interesting questions about the types of Knuth relations possible, given the shape of the tableau $P(\pi)$. All of these notes were written following Bruce Sagan's wonderful text [Sag01].

Date: July 22, 2025.

2. The Robinson Schensted (RS) Algorithm

The Robinson-Schensted (RS) Algorithm is a set of steps that takes a finite permutation of numbers and outputs a *tableau*. To describe the RS Algorithm, we first need some foundational terminology.

Definition 2.1. (Partial Tableau) A partial tableau is an array with distinct entries whose rows and columns increase. When a number n is in a partial tableau, we say that $n \in P$ in cell (i, j) where i is the row and j is the column that n is located in. ▨

Example 2.2. For some examples, consider the following:

$$P = \begin{array}{ccc} 1 & 3 & 6 \\ 2 & 5 & \\ 4 & 7 & \end{array}, \quad P' = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & \end{array}, \quad P'' = \begin{array}{ccc} 1 & 3 & 5 \\ 2 & & \\ 4 & & \\ 6 & & \\ 7 & & \end{array}, \quad P''' = \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & \\ 6 & & \\ 7 & & \end{array}$$

. Note that in all the above examples, both the rows and columns are increasing. Furthermore, note $4 \in P$ in cell $(3, 1)$. Similarly, $7 \in P'$ in cell $(2, 3)$. ■

Definition 2.3. (Partition of n) A partition of $n \in \mathbb{N}$ is a sequence,

$$\lambda := (\lambda_1, \lambda_2, \cdots, \lambda_l),$$

where $\lambda_{i+1} \leq \lambda_i$ and $\sum_{i=1}^{l} \lambda_i = n$. If λ partitions n then we write $\lambda \vdash n$. ▨

Example 2.4. Consider $n = 10$. Then some partitions of 7 are,

$$\lambda = (3, 2, 2), \qquad \lambda' = (4, 3), \qquad \lambda'' = (3, 1, 1, 1, 1), \qquad \lambda''' = (3, 2, 1, 1)$$

We can use a partition of n to describe the shape of a tableau with n cells filled in. To do so, set λ_i equal to the number of elements in the i^{th} row. In this way, the partitions above describe the shape of the tableaux in example 1.2, respectively. ■

Algorithm 1 (Robinson Schensted (RS) Algorithm)**.** Let P be a partial tableau and let $x \notin P$. Denote the element in cell (i, j) of P by $P_{i,j}$. We now describe the process of *row-inserting* x *into* P.

(1) Compare x to all the elements in the first row of P. This can result in two situations:
  - (1a) The number x is greater than all the elements in the first row. When this happens, place x in the first open cell in the first row. We are done with row insertion of x.
  - (1b) There is some $P_{1,j}$ larger than x, say $P_{1,\,j}$ is the *smallest element larger* than x in the first row so that $x < P_{1,\,j}$.
    When this happens, we say that x *bumps out* $P_{1,j}$ of cell $(1, j)$ by replacing $P_{1,j}$ with x in cell $(1, j)$. We end (1b) with x in cell $(1, j)$ and $P_{i,\,j}$ outside of the tableau. Continue to step 2.

(2) Since we made it to step (2), it must have been the case that (1)(ii) occurred. We are therefore starting with x in cell $(1, j)$ and $P_{1,j}$ outside of the tableau P. Since $P_{1,j}$ is not in the tableau, we are in the position to row insert $P_{1,j}$ into the *second row* of the tableau P. To do this, we treat row-inserting $P_{1,j}$ into row 2 the same way we row-inserted x into the first row. That is, we compare $P_{1,j}$ to all the elements in the second row of P. We will end up with situations similar to (1a) and (1b).
  - (2a) If $P_{1,j}$ is larger than all the elements in the second row (or if the second row is empty), then put $P_{1,j}$ in the first open cell of row 2 (like in (1a)). When this occurs, we have completed the row insertion process.
  - (2b) Or, if there is some element in the second row larger than $P_{i,j}$ then we locate the smallest element larger than $P_{i,j}$. Let the element in cell $(2, k)$ be the smallest element larger than $P_{i,j}$ in the second row by $P_{2,k}$, that is, we have $P_{1,j} < P_{2,\,k}$. We then bump $P_{2,k}$ out of cell $(2, k)$ by replacing $P_{2,\,k}$ with $P_{1,j}$ and thereby placing $P_{1,j}$ into cell $(2, k)$ (like in (1b)). Continue to step 3.

(3) We now have $P_{1,j}$ in cell $(2, k)$ and the element $P_{2,k}$ ready to be row-inserted into row 3 in a similar way to the previous two steps. Continue this process until you either reach a situation similar to (1a) or (2a).

We use $r_x(P) = P'$ to denote that the result of row-inserting x into P yields P'.

Example 2.5. Consider the partial tableau,

$$P \;=\; \begin{matrix} 1 & 2 & 3 \\ 5 & & \\ 6 & & \end{matrix}$$

We aim to determine $r_4(P)$. Note that 4 is larger than every element in the first row. This is situation (1a). Thus, we place 4 into cell $(1, 4)$ yielding,

$$r_4(P) \;=\; \begin{matrix} 1 & 2 & 3 & 4 \\ 5 & & & \\ 6 & & & \end{matrix},$$

Let's consider a row insertion that takes a few more steps. ■

Example 2.6. Consider the partial tableau,

$$P = \begin{array}{ccc} 1 & 5 & 6 \\ 2 & 7 & \\ 4 & & \end{array}$$

Note that $3 \notin P$, so we can row-insert 3 into P to determine $r_3(P)$. First, observe that $1 < 3 < 5$ and therefore 5 is the smallest element in the first row that is larger than 3. Therefore, step (1b) tells us that 3 bumps 5 from the first row to the second row, leaving 3 in the cell that 5 was located, that is, cell $(2,1)$. From there, step 2 tells us to compare 5 to all the numbers in the second row. We find that $2 < 5 < 7$, and hence 5 bumps 7 out of the second row and into the third (step (2b)). Finally, comparing 7 to all the numbers in the third row gives, $7 > 4$, so 7 ends up in cell $(3,2)$ (step (3a)).

$\begin{array}{cccl} 1 & \mathbf{5} & 6 & \leftarrow \mathbf{3} \\ 2 & 7 & & \\ 4 & & & \end{array}$	$\begin{array}{cccl} 1 & 3 & 6 & \\ 2 & \mathbf{7} & & \leftarrow \mathbf{5} \\ 4 & & & \end{array}$	$\begin{array}{cccl} 1 & 3 & 6 & \\ 2 & 5 & & \\ 4 & & & \leftarrow \mathbf{7} \end{array}$	$\begin{array}{ccc} 1 & 3 & 6 \\ 2 & 5 & \\ 4 & 7 & \end{array}$

Thus, we end up with the final tableau

$$r_3(P) = \begin{array}{ccc} 1 & 3 & 6 \\ 2 & 5 & \\ 4 & 7 & \end{array}$$

concluding our example. ■

Suppose that you had the finite permutation in two-line notation,[1]

$$\pi = \begin{pmatrix} 1 & 2 & \cdots & n \\ \pi(1) & \pi(2) & \cdots & \pi(n) \end{pmatrix} \in S_n,$$

and an empty tableau $P_0 = \emptyset$. We can now attempt to row-insert every $\pi(i)$ into P_0 using the RS algorithm. First, we row-insert $\pi(1)$ into P_0 giving $P_1 = r_{\pi(1)}(P_0)$. From there, we row insert $\pi(2)$ into P_1 giving $P_2 = r_{\pi(2)}(P_1)$, and so on, until we row insert $\pi(n)$ into P_{n-1} to get P_n. By undergoing this process, we create a tableau P_n from the permutation π which we will call the *insertion tableau*. Moreover, for each row-insertion step, we can *record* where the final insertion ended, see Example 2.7 below. In so doing, we create another tableau, Q_n, which we will call the *recording tableau*.

Algorithm 2. [Insertion and Recording tableaux (P_n, Q_n)] *We define (P_n, Q_n) recursively. Start with P_0 and Q_0 being empty tableaux, and let*

$$(P_k, Q_k) = \Big(r_x(P_{k-1}) \ , \ \text{Place k into } Q_{k-1} \text{ where the insertion terminated} \Big)$$

[1]We let S_n be the set of all permutations of the set $\{1, 2, 3, \cdots, n\}$.

Example 2.7. Let's take the permutation $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 7 & 2 & 3 & 6 & 1 \end{pmatrix} \in S_7$.

$P_0 = \emptyset \qquad Q_0 = \emptyset$

$P_1 = r_4(P_0) = 4 \qquad Q_1 = 1$ because 4 ended in cell $(1,1)$ in P_1

$P_2 = r_5(P_1) = \begin{matrix} 4 & \boxed{5} \end{matrix} \qquad Q_2 = \begin{matrix} 1 & 2 \end{matrix}$ because 5 ended in cell $(1,2)$ in P_2

$P_3 = r_7(P_2) = \begin{matrix} 4 & 5 & \boxed{7} \end{matrix} \qquad Q_3 = \begin{matrix} 1 & 2 & 3 \end{matrix}$ because 7 ended in cell $(1,3)$ in P_3

$P_4 = r_2(P_3) = \begin{matrix} 2 & 5 & 7 \\ \boxed{4} \end{matrix} \qquad Q_4 = \begin{matrix} 1 & 2 & 3 \\ 4 \end{matrix}$ because 4 ended in cell $(2,1)$ in P_4

$P_5 = r_3(P_4) = \begin{matrix} 2 & 3 & 7 \\ 4 & \boxed{5} \end{matrix} \qquad Q_5 = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 \end{matrix}$ because 5 ended in cell $(2,2)$ in P_5

$P_6 = r_6(P_5) = \begin{matrix} 2 & 3 & 6 \\ 4 & 5 & \boxed{7} \end{matrix} \qquad Q_6 = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix}$ because 7 ended in cell $(2,4)$ in P_6

$P = r_1(P_6) = \begin{matrix} 1 & 3 & 6 \\ 2 & 5 & 7 \\ \boxed{4} \end{matrix} \qquad Q = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 \end{matrix}$ because 4 ended in cell $(3,1)$ in P

■

We could show that the RS algorithm produces a bijection between S_n and (P, Q) where P and Q are the insertion and recording tableaux. The easiest way to do this is to show that this is a bijection is to show that the RS algorithm has an *inverse*. That is, there is a way to go from a pair of tableaux (P_n, Q_n) to a permutation $\pi \in S_n$. We will not detail this inverse RS algorithm; instead, we will go through an example that highlights the process.

We will use the P and Q tableaux we just found in Example 2.7 in our example, going from (P_n, Q_n) to a permutation $\pi \in S_n$. We will reverse-engineer the permutation that produced P and Q.

Example 2.8. Take the two tableaux from Example 2.7,

$$P = \begin{matrix} 1 & 3 & 6 \\ 2 & 5 & 7 \\ 4 \end{matrix} \qquad Q = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6. \\ 7 \end{matrix}$$

First, locate the largest number in Q. In this case, 7 is located in cell $(3,1)$. In general, let's say the largest number in Q is located in cell (i,j). Then, find the number in P in the corresponding cell, i.e., $P_{3,1}$ or in general $P_{i,j}$.

$$P = \begin{matrix} 1 & 3 & 6 \\ 2 & 5 & 7 \\ \boxed{4} & & \end{matrix} \qquad Q = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ \boxed{7} & & \end{matrix}$$

Since the only way for an element to have ended up in a row below row 1, it must have been bumped from the previous row by the largest element that is smaller than $P_{i,j}$. Say this element is in cell $(i-1,l)$. Replace this element $P_{i-1,l}$ by $P_{i,j}$. If $i-1=1$ (i.e. $P_{i-1,l}$ was in the first row), then you are finished. However, if $i-1>1$, then continue this process by finding the largest element in row $i-2$ that is smaller than $P_{i-1,k}$. Continue this process until you reach the first row.

Back to our example, in this case 4 is in the third row of P, so it must have been bumped out of the second row by 2, since 2 is the only number in the second row that is smaller than 4. Similarly, 2 must have been bumped out of the first row by 1. Therefore, we move 4 to where 2 is and 2 to where 1 is, giving:

$$P_6 = \begin{matrix} 2 & 3 & 6 \\ 4 & 5 & 7 \end{matrix} \qquad Q_6 = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix}$$

And since 1 is the last number we deleted from P we deduce that $\pi(7)=1$.

Next, we find where 6 is located (since 6 is now the largest element in Q_6) in Q_6 and find the corresponding element in P_6.

$$P_6 = \begin{matrix} 2 & 3 & 6 \\ 4 & 5 & \boxed{7} \end{matrix} \qquad Q_6 = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & \boxed{6} \end{matrix}$$

Just like before, since we are looking at the second row, we know that 7 was bumped out of the first row by 6. We now replace 6, in the first row, with 7, from the second row.

$$P_5 = \begin{matrix} 2 & 3 & 7 \\ 4 & 5 & \end{matrix} \qquad Q_5 = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & \end{matrix}$$

Since 6 is the number we deleted from P, we deduce that $\pi(6)=6$.

Now, 5 is located in cell $(2,2)$ in both P_5 and Q_5.

$$P_5 = \begin{matrix} 2 & 3 & 7 \\ 4 & \boxed{5} & \end{matrix} \qquad Q_5 = \begin{matrix} 1 & 2 & 3 \\ 4 & \boxed{5} & \end{matrix}$$

Again, because 5 is in the second row, we deduce that 5 was bumped out by 3 in the first row. Thus, we replace the 3 with 5, giving,

$$P_4 = \begin{matrix} 2 & 5 & 7 \\ 4 & & \end{matrix} \qquad Q_4 = \begin{matrix} 1 & 2 & 3 \\ 4 & & \end{matrix}$$

Continue until we get back to two empty tableaux P_0 and Q_0. ■

2.1. **Row-Insertion Properties and Column Insertion.** So far, we have learned how to take a permutation and use the RS algorithm to output the *insertion* tableau, while keeping track of the row insertions at each step to create the recording tableau. We can also define *column-insertion* in a similar way to that of row-insertion. That is, everywhere the word 'row' was used, swap it for the word 'column'.

We now aim to describe some of the properties that row insertion and column insertion enjoy.

Lemma 2.9. Let P be a partial tableau with $x \notin P$. Suppose that during the insertion $r_x(P) = P'$, the elements $x_1, x_2, x_3, \cdots$ are bumped from the cells $(1, j_1), (2, j_2), (3, j_3), \cdots$ respectively. Then,

(i) $x < x_1 < x_2 < x_3 < \cdots$
(ii) $j_1 \geq j_2 \geq j_3 \geq \cdots$
(iii) $P'_{i,j} \leq P_{i,j}$ for all i, j.

Proof. Let P be a partial tableau with $x \notin P$ and suppose that during the row-insertion $r_x(P) = P'$, the elements $x_1, x_2, x_3, \cdots$ are bumped from the cells $(1, j_1), (2, j_2), (3, j_3), \cdots$ respectively.

Part *(i)* follows from the fact that the algorithm tells us that x_i only bumps out x_{i+1} when $x_i < x_{i+1}$.

Regarding part *(ii)*, we will show $\cdots \geq j_{k-2} \geq j_{k-1} \geq j_k$ for all $k \in \mathbb{N}$ where j_k exists. To this end, assume, for the hope of a contradiction, that there exists some i where $j_i < j_{i+1}$. Since x_i is the i^{th} element to be bumped out, it must originate in the i^{th} row and, by assumption, x_i was located in the j_i^{th} column. Therefore $x_i = P_{i,j_i}$ and,

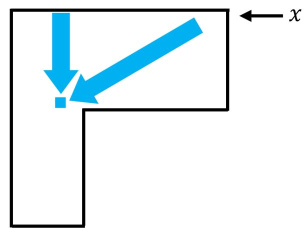
$$x_i := P_{i,j_i} < P_{(i+1),\, j_i}$$
$$P_{(i+1),\, j_i} < x_i < x_{i+1} := P_{(i+1),\, (j_{i+1})}.$$

Where we have $P_{i,j_i} < P_{(i+1),\,(j_i)}$ since P is assumed to be a partial tableau (so that its columns are increasing as you go down), and $P_{(i+1),\, j_i} < x_i < P_{(i+1),\,(j_{i+1})}$ follows from the fact that x_i bumped out x_{i+1}. These give the contradiction,

$$x_i < P_{(i+1),\, j_i} < x_i < x_{i+1}$$

Lastly, part *(iii)*. In the situation that no number was bumped out in cell (i, j), it must be that $P'_{i,j} = P_{i,j}$. On the other hand, if a number was pumped out of (i, j), then the number bumped out would have been replaced with a smaller number since $x_i < x_{i+1}$. Therefore, $P'_{i,j} < P_{i,j}$, concluding the proof. □

Remark 2.10. Schematically, lemma 2.9 tells us that the *row-insertion path* defined to be $(1, j_1), (2, j_2), (3, j_3), \cdots$, is down and weakly left.



So far, we have discussed only row-inserting an element into a tableau; however, we can define a similar process for column insertion. As it turns out, row insertion and column insertion commute with each other.

Proposition 2.11. *For any partial tableau P and distinct elements $x, y \notin P$ we have,*

$$c_y r_x(P) = r_x c_y(P).$$

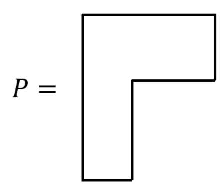
I.e., row and column insertion commute!

Proof. Let P be a partial tableau and let the distinct elements $x, y \notin P$. The key idea in the proof given in [Sag01] is that we want to follow the maximum element in $(x, y) \cup P$. From there, our proof breaks up into two cases (with multiple subcases!).

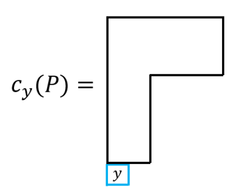
First, let $m = \max\{n : n \in (x, y) \cup P\}$. The proof breaks up into whether or not $m \in P$ or $m \notin P$.

Case 1: $y = m$

Let P be represented schematically by,

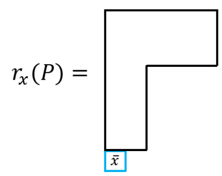


Consider column inserting y first. Since $y = m$ and m is maximal, when y is column inserted into P it won't bump any element out of the first column. Therefore, we end up with:

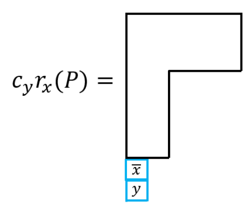


On the other hand, if we insert x first, let $\overline{x}$ denote the last element that is bumped out during the row-insertion of x. Furthermore, we will say that $\overline{x}$ ends up in cell u. Now, consider whether or not u is located in the first column.

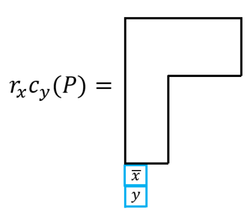
Subcase 1a: When $\overline{x}$ ends up in the first column
In the case that $\overline{x}$ ends up in the first column, we have:



Since we have $\overline{x} < y$, when we column insert y we need only to put y at the bottom of the first row, giving the tableau,



Similarly, when we row insert x into $c_y(P)$, there would be no difference in the steps until we reached the final row. Since y is in cell u, the element $\overline{x}$ would bump out y giving the same tableau as above,

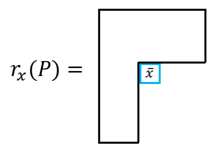


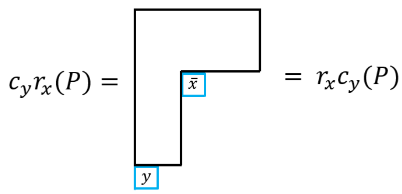
completing subcase 1a.

Subcase 1 b: When $\overline{x}$ does not end up in the first column
When $\overline{x}$ doesn't end up in the first column, we have a situation represented below,
In this situation, we can see that column-inserting y into $r_x(P)$ gives the same as row-inserting x into $c_y(P)$.

In either of the subcases 1a and 1b, we have seen that the row and column insertion commute.

Case 2: $x = m$





This situation is very similar to case 1.

Case 3: When $m \in P$

Now that the maximum element is in P prior to the row and column insertions, we choose to use induction on the number of entries that P has. Our base case is when P only has m as an element:

$$P = m.$$

This breaks down into two situations: when $x < y$ and when $x > y$.

Suppose first that $x < y$, when we row insert x before we column insert y we end up with the following tableau,

$$r_x(P) = m \leftarrow x = \begin{array}{c} x \\ m \end{array} \qquad \text{and then} \qquad c_y r_x(P) = \begin{array}{cc} x & m \\ \uparrow & \\ y & \end{array} = \begin{array}{cc} x & m \\ y & \end{array}$$

Likewise, when we column insert y and then row insert x we arrive at the same tableau,

$$c_y(P) = \begin{array}{c} m \\ \uparrow \\ y \end{array} = y \quad m \qquad \text{and then} \qquad r_x c_y(P) = y \quad m \leftarrow x = \begin{array}{cc} x & m \\ y & \end{array}$$

Now consider when $x > y$. We have a very similar situation:

$$r_x(P) = m \leftarrow x = \begin{array}{c} x \\ m \end{array} \qquad \text{and then} \qquad c_y r_x(P) = \begin{array}{c} x \\ m \\ \uparrow \\ y \end{array} = \begin{array}{cc} y & \\ m & \\ & \uparrow \\ & x \end{array} = \begin{array}{cc} y & x \\ m & \end{array}$$

and

$$c_y(P) = \begin{array}{c} m \\ \uparrow \\ y \end{array} = y \quad m \qquad \text{and then} \qquad r_x c_y(P) = y \quad m \leftarrow x = \begin{array}{cc} y & x \\ m & \end{array}.$$

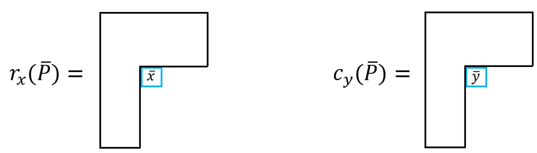
This concludes our base case.

Assume that any partial tableau with n entries has the property that row and column insertions commute. Consider some partial tableau P with $n+1$ entries and denote the partial tableau P with m erased by $\overline{P}$. Then $\overline{P}$ has n entries and by induction we have $c_y r_x(\overline{P}) = r_x c_y(\overline{P})$. Thus, $c_y r_x(P)$ and $r_x c_y(P)$ must agree everywhere except possibly where m is located!

Our goal is reduced to showing that m is located in the same cell for both $c_y r_x(P)$ and $r_x c_y(P)$. We accomplish this by considering different cases once again. For each subcase, let $\overline{x}$ be the last element bumped into cell u during the insertion $r_x(\overline{P})$. Likewise, denote the last element bumped during $c_y(\overline{P})$ by $\overline{y}$, and let $\overline{y}$ end up in cell v.

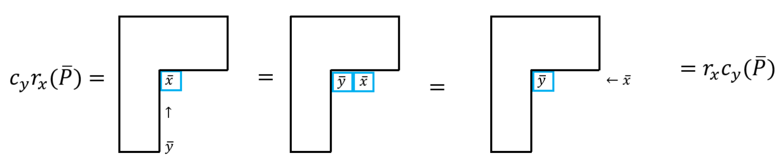
Subcase 3a: $u = v$**.**

Let $u = v$. We begin by representing the partial tableaux after row-inserting x and column-inserting y as follows,



Recall that the previous lemma 2.9 tells us that every cell affected by the row insertion of x will be located in a row weakly to the right of cell $u = v$. This means that when we row insert x into $\overline{P}$ and $c_y(\overline{P})$ the same cells are affected up to the the last step when $\overline{x}$ would have ended up in cell $u = v$ in $r_x(\overline{P})$. Two things can happen in this last step, depending on whether $\overline{x} > \overline{y}$ or $\overline{x} < \overline{y}$.

Suppose $\overline{x} > \overline{y}$. Then when we column insert y into $r_x(\bar{P})$ we deduce that $\overline{y}$ bumps out $\overline{x}$. Also, when we row insert x into $c_y(\bar{P})$, we would end up placing $\bar{x}$ in the cell to the right of u. Thus, giving,

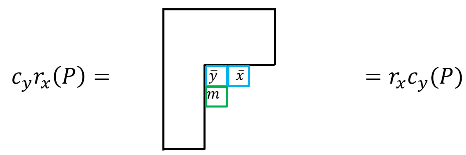


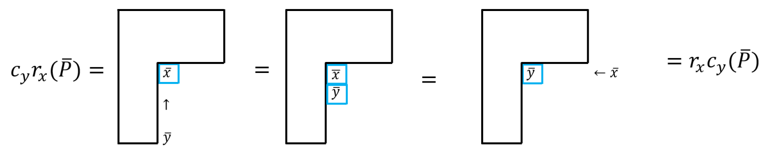
so that $\overline{x}$ and $\overline{y}$ end up in the same row.

If m is not in cell $u = v$, then no insertions will displace it since we know that $\bar{x}$ and $\bar{y}$ ended up in their respective cells without needing m. However, if m was in cell $u = v$, then we have the following situation:

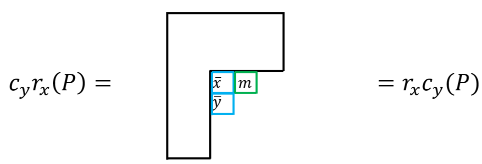
Which you can see by considering what would happen in the final steps of the row and column insertions.

On the other hand, when $\overline{x} < \overline{y}$ we have,





so that $\overline{x}$ and $\overline{y}$ end up in the same column. Again, if m was not in cell $u = v$, no insertion would bump it; however, if m was in cell $u = v$ then we would end up with



In either case, we have row and column insertion commuting.

Subcase 3b: $u \neq v$

Let $\overline{P}$ have shape $\overline{\lambda}$ and let $c_y r_x(\overline{P}) = r_x c_y(\overline{P})$ have shape λ. By comparing $r_x(\overline{P})$ with $c_y r_x(\overline{P})$ we note that $\overline{\lambda} \cup \{u\} \subset \lambda$. This is because when we perform the row and column insertions of x and y, there is no way to delete a cell that's already filled in. This logic also leads us to the conclusion that $\overline{\lambda} \cup \{v\} \subset \lambda$. Since λ must have two more cells than $\overline{\lambda}$ we deduce,

$$\lambda = \overline{\lambda} \cup \{u, v\}.$$

However, this does not mean that $\overline{x}$ ends up in u nor does it mean that $\overline{y}$ will end up in v since the paths that $r_x(\overline{P})$ and $c_y(\overline{P})$ take may *cross*.

When the insertion paths of x and y do not cross, we only need to consider if m is located in cell u or v prior to the insertions. If $m \in P$ in cell u, then we know that $\bar{x}$ will bump out m into the next row. There are two situations that can occur: (1) m is bumped into cell v and (2) m is not bumped into cell v.

Let's consider the situation when m is bumped into cell v. When x is row inserted first, we end up with m in cell v. This means that $\bar{y}$ will then bump m into the next column, into the cell to the right of cell v. On the other hand, when we column insert y, cell u is unaffected since the insertion paths don't cross. Thus, when we row insert x, we know that $\bar{x}$ will bump out m into the next row, to the right of $\bar{y}$. Thus, we get that $c_y r_x(P) = r_x c_y(P)$.

When m is not bumped into cell v, it's clear that $c_y r_x(P) = r_x c_y(P)$ since neither the row insertion nor the column insertion affects the other. A similar argument can be used for the situation where $m \in$ P is in cell v.

A more challenging situation arises when the insertion paths of x and y do cross. We need to consider cases when m is located in one of these cells or not. If m is not located in one of these cells, then $c_y r_x(P) = r_x c_y(P)$ since the two could have only differed by what happened to m. On the other hand, if m is located in one of the cells that is affected by both insertion paths, then we can use a similar argument that we made two paragraphs ago to conclude that $c_y r_x(P) = r_x c_y(P)$, concluding the proof. □

We finish out this section by mentioning a theorem regarding a tableau and its transposition.

Theorem 2.12. *If $P(\pi) = P$, then $P(\pi^r) = P^t$. Where,*

$$\pi = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ x_1 & x_2 & x_3 & \cdots & x_n \end{pmatrix} \quad \text{and} \quad \pi^r = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ x_n & x_{n-1} & x_{n-2} & \cdots & x_1 \end{pmatrix},$$

and P^t is the transposition of P, which is the tableau you get when you swap cell (i, j) of P with cell (j, i) of P.

3. Increasing and Decreasing Subsequences

Definition 3.1. Given a permutation,

$$\pi = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ x_1 & x_2 & x_3 & \cdots & x_n \end{pmatrix} \in S_n$$

we define an **increasing** subsequence of π as:

$$x_{i_1} < x_{i_2} < x_{i_3} < \cdots x_{i_k}.$$

where $i_1 < i_2 < \cdots i_k$.

Similarly, a **decreasing** subsequence of π is defined to be,

$$x_{i_1} > x_{i_2} > x_{i_3} > \cdots x_{i_k}$$

where we still have $i_1 < i_2 < \cdots i_k$.

In both cases, we call k the length of the subsequence. ▨

Example 3.2. Consider the permutation $\pi = \left(\begin{smallmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 7 & 2 & 3 & 6 & 1 \end{smallmatrix}\right)$ from Example 2.7. Some of the increasing and decreasing subsequences are,

$$\begin{array}{rlll} \text{Increasing}: & 4,5,7 & 2,3,6 & 4,5,6. \\ \text{Decreasing}: & 7,6,1 & 7,2,1 & 4,2,1. \end{array}$$

An interesting question that one could ask is, "*Given a permutation, what is the longest increasing (decreasing) subsequence we can find?*" As we will show in the following theorem, the answer can be found using the RS algorithm and the shape of the resulting tableau. More precisely, the answer is found in the length of the first row (column) of the tableau. For example, observe that $P(\pi)$'s first row and column have three elements in them (see Example 2.7) and that the longest increasing and decreasing sequence of π are of length 3. ■

Theorem 3.3. *Consider some $\pi \in S_n$. The length of the longest increasing subsequence of π is the length of the first row of $P(\pi)$. Similarly, the length of the longest decreasing subsequence of π is the length of the first column of $P(\pi)$.*

Proof. Note that the symmetry between a tableau and its transpose allows us to focus on proving the increasing subsequence portion of the theorem, and the decreasing subsequence statement will follow from the former.

Let $\pi \in S_n$ and let P_{k-1} denote the tableau formed after $k-1$ insertions of the RS algorithm. Denote π in two-line notation,

$$\pi = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ x_1 & x_2 & x_3 & \cdots & x_n \end{pmatrix}.$$

We will prove that, if x_k enters P_{k-1} in column j, then the longest increasing subsequence of π ending in x_k has length j.

We induct on k. Our base case is the trivial case when $k = 1$. Now, assume that our result holds for all values up to $k-1$. We will continue this proof in two steps. First, we will show he existence of an increasing subsequence of length j ending in x_k. Then, we will show that there aren't any subsequences of length greater than j.

(1) Let y be the element in P_{k-1} that is located in cell $(1, j-1)$. Recall by assumption x_k enters into the column j, thus $y < x_k$. By induction, we know that there is an increasing subsequence of length $j-1$ ending in y. Adding x_k to this subsequence gives our desired subsequence of length j.

(2) Assume for the hope of a contradiction that there is a subsequence that has a length longer than j and that ends with x_k. Denote the element that precedes x_k in this subsequence by x_i and consider what happens when we row insert x_i into P_i. By induction, when x_i is inserted, it enters into some column weakly to the right of column j. Thus, the element y in cell $(1, j)$ satisfies the inequality $y \leq x_i$ and since x_i precedes x_k in the increasing subsequence we have,

$$y \leq x_i < x_k.$$

By Lemma 2.9 (iii), we know that during the insertion process, elements of the tableau never increase. Therefore, the element in cell $(1, j)$ of P_{k-1} is smaller than x_k. This contradicts that x_k displaces it. ☐

4. The Knuth Relations

Definition 4.1. Two permutations $\pi, \sigma \in S_n$ are said to be P-equivalent if $P(\pi) = P(\sigma)$. When this happens we write $\pi \overset{P}{\cong} \sigma$.

Example 4.2. Consider the permutation group S_3,

$$S_3 = \left\{ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \right\}$$

By performing the RS algorithm on each of the permutations in S_3 we can show that the set of permutations,

$$\mathcal{P}_1 = \left\{ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \right\}, \qquad \mathcal{P}_2 = \left\{ \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} \right\},$$

$$\mathcal{P}_3 = \left\{ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} \right\}, \qquad \mathcal{P}_4 = \left\{ \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \right\},$$

correspond to the tableaux,

$$P_1 \;=\; \begin{matrix} 1 & 2 & 3 \end{matrix} \qquad P_2 \;=\; \begin{matrix} 1 & 3 \\ 2 & \end{matrix} \qquad P_3 \;=\; \begin{matrix} 1 & 2 \\ 3 & \end{matrix} \qquad P_3 \;=\; \begin{matrix} 1 \\ 2 \\ 3 \end{matrix}$$

respectively. From this example, we might conjecture that P-equivalence is an equivalence relation on the set S_n. Indeed, this is the case; however, we also want to express this equivalence relation in another way. This leads us to what are known as *Knuth relations*. ■

Definition 4.3. (**Knuth Relations/Equivalence**) Suppose $\mathbf{x} < \mathbf{y} < \mathbf{z}$. Then we say, $\pi, \sigma \in S_n$ differ by a Knuth relation of the first kind if,

$$\pi = x_1 \cdots \mathbf{y}\,\mathbf{x}\,\mathbf{z} \cdots x_n \qquad \text{and} \qquad \sigma = x_1 \cdots \mathbf{y}\,\mathbf{z}\,\mathbf{x} \cdots x_n$$

or vice versa. When this happens we write $\pi \overset{1}{\cong} \sigma$. Similarly, π and σ differ by a Knuth relation of the second kind if,

$$\pi = x_1 \cdots \mathbf{x}\,\mathbf{z}\,\mathbf{y} \cdots x_n \qquad \text{and} \qquad \sigma = x_1 \cdots \mathbf{z}\,\mathbf{x}\,\mathbf{y} \cdots x_n$$

or vice versa. When this happens, we write $\pi \overset{2}{\cong} \sigma$. In the case that π and σ satisfy both $\pi \overset{1}{\cong} \sigma$ and $\pi \overset{2}{\cong} \sigma$, we denote this situation by $\pi \overset{KB}{\cong} \sigma$.

Finally, we say that two permutations are **Knuth Equivalent** if there is a sequence of permutations such that,

$$\pi = \pi_1 \overset{i}{\cong} \pi_2 \overset{j}{\cong} \cdots \overset{l}{\cong} \pi_k = \sigma$$

where $i, j, \cdots, l \in \{1, 2\}$. We denote when two permutations are Knuth Equivalent by $\pi \overset{K}{\cong} \sigma$. ▨

Remark 4.4. As the reader might have guessed, these two notions—P-equivalence and Knuth equivalence—are related to one another. It turns out, in fact, they are equivalent to one another.

Theorem 4.5. Let $\pi, \sigma \in S_n$. Then Knuth equivalence is equivalent to P-equivalence,

$$\pi \overset{K}{\cong} \sigma \iff \pi \overset{P}{\cong} \sigma.$$

Remark 4.6. Before we proceed with the proof, we will introduce a definition and a lemma that will make our task easier.

Definition 4.7. Let P be a tableau. The row word of P is defined to be the permutation,

$$\pi_P = R_l R_{l-1} \cdots R_1.$$

Where R_l , R_{l-1} , $\cdots$, R_1 are the rows of P. ▨

Example 4.8. Consider the tableau from Example 2.7: $P = \begin{array}{lll} 1 & 3 & 6 \\ 2 & 5 & 7 \\ 4 & & \end{array}$. The row word of P is given by:

$$\pi_P = 4\ 2\ 5\ 7\ 1\ 3\ 6.$$

■

Example 4.9. Consider the tableau $P' = \begin{array}{llll} 1 & 2 & 7 & 8 \\ 3 & 5 & & \\ 4 & 6 & & \end{array}$. The row word of P is given by:

$$\pi_{P'} = 4\ 6\ 3\ 5\ 1\ 2\ 7\ 8.$$

■

As one might guess, when you apply the RS algorithm to the row reading word π_P of a tableau P, you get back the tableau P. That is, $P(\pi_P) = P$, as seen in the following lemma.

Lemma 4.10. *If P is a standard tableau, then* $\pi_P \overset{R-S}{\longleftrightarrow} (P, \cdot)$.

Proof of Lemma. Let π_P be the reading word of the standard tableau P. We aim to prove that $P(\pi_P) = P$. First denote π_P by,

$$\pi_P = R_k R_{k-1} \cdots R_1,$$

where R_i is the i^{th} row of P. Furthermore, let each row be denoted $R_i = a_{i,1}\, a_{i,2}\, a_{i,3} \cdots\, a_{i,k_i}$. The keys to this proof are the following observations regarding the row R_i:

(i) $a_{i,j} < a_{i,j+1}$ for all i and j,
(ii) $a_{i,j} < a_{i+1,j}$ for all i and j, and
(iii) the length of R_i is less than or equal to the length of R_{i+1} for all i.

Which are true since R_k are assumed to be rows of the standard tableau P.

Using part (i), we deduce that there will be no bumping when we row-insert all the elements from R_k and hence, will form a single row. Then by (ii) and (iii) we deduce that every element in R_{k-1} bumps out every element in R_k while not bumping out any of the other elements in R_{k-1}. This happens for all the elements in each of the rows R_i. □

Proof of Theorem. Let $\pi, \sigma \in S_n$ and consider each direction of the proof.

Forward: $\pi \overset{K}{\cong} \sigma \implies \pi \overset{P}{\cong} \sigma$.

Let $\pi \overset{K}{\cong} \sigma$. We will start by showing that if $\pi \overset{1}{\cong} \sigma$ then $\pi \overset{P}{\cong} \sigma$.

To this end, assume $\pi \overset{1}{\cong} \sigma$ and let P_y be the tableau you'd get before inserting y into the tableau using the RS algorithm. What we need to do is show that,

$$r_z r_x r_y(P_y) = r_x r_z r_y(P_y). \tag{★}$$

We do this using induction on the number of rows in P_y.

Our base case is when our tableau has no rows, that is, when $P = \emptyset$. Recall $x < y < z$, so that when we row insert x and then z we obtain,

$$r_y(P_y) \;=\; y \quad ,$$

$$r_x r_y(P_y) \;=\; y \;\leftarrow x \;=\; \begin{array}{l} x \\ y \end{array}$$

$$r_z r_x r_y(P_y) \;=\; \begin{array}{l} x \\ y \end{array} \leftarrow z \;=\; \begin{array}{ll} x & z \\ y & \end{array}$$

Similarly, when we row insert z and then x we obtain,

$$r_y(P_y) \;=\; y \quad ,$$

$$r_z r_y(P_y) = y \;\leftarrow z \;=\; \begin{array}{ll} y & z \end{array}$$

$$r_z r_x r_y(P_y) \;=\; \begin{array}{ll} y & z \end{array} \leftarrow x \;=\; \begin{array}{ll} x & z \\ y & \end{array}.$$

This concludes our base case.

Now, assume that any partial tableau with $r-1$ rows has the property given in Equation (★). Consider a tableau P_y with r rows and let $\overline{P} = r_y(P_y)$. Suppose that y enters P_y in cell $(1,k)$ and, in the process, bumps $\overline{y}$ out. Denote the element in cell (n,m) of $\overline{P}$ by $\overline{P}_{n,m}$ and note the following two properties,

(☺) $\overline{P}_{1,j} \leq y$ for all $j \leq k$, and
(🥁) $\overline{P}_{1,t} > \overline{y}$ for all $t > k$.

Since y bumps $\overline{y}$ out of cell $(1,k)$ we know that $\overline{P}_{1,j} < y < \overline{y} < \overline{P}_{1,t}$ for $j \leq k < t$. We now consider the two cases, depending on the order in which we row-insert x and z into the tableau.

Case 1: Row-inserting x and then z.

Consider what happens when we preform $r_x(\overline{P})$. Since $x < y$ we know that x will enter $\overline{P}$ (weakly) left of cell $(1,k)$, where y is. Say x enters $\overline{P}$ in cell $(1,j)$, where $j \leq k$, bumping out $\overline{x}$ from $(1,j)$. Using (☺) we know that $\overline{x} \leq y < \overline{y}$.

Next, preform $r_z r_x(\overline{P})$. We know that z enters $r_x(\overline{P})$ in a cell to the right of where y is located, since $z > y$. Denote the cell where z enters by $(1,t)$, note that $t > k$, and let z bump out $\overline{z}$ from $(1,t)$ in the process. Observe that $\overline{z} > \overline{y}$ by (🥁). Altogether we have shown $\overline{x} < \overline{y} < \overline{z}$.

Case 2: Row-inserting z and then x.

Note that both (☺) and (🥁) hold independently of the order in which we row insert x and z. Therefore, we still deduce that z ends up in a cell to the right of $(1,k)$ and x still ends up in a cell weakly to the left of $(1,k)$. This leads us to the same conclusion regarding the numbers that z and x bump out, namely $\overline{z}$ and $\overline{x}$ respectively. Altogether, we still conclude $\overline{x} < \overline{y} < \overline{z}$.

Bringing Cases 1 and 2 together.

The previous two paragraphs lead us to the conclusion that the first row of $r_z r_x r_y(P_y)$ and $r_x r_z r_y(P_y)$ are the same, and thus the only difference between the tableaux $r_z r_x r_y(P_y)$ and $r_x r_z r_y(P_y)$, that can occur, is in the bottom $r-1$ rows.

Let P' denote the tableau with only the bottom $r-1$ rows of P_y. We can find the bottom rows of $r_z r_x r_y(P_y)$ and $r_x r_z r_y(P_y)$ using $r_{\bar{z}} r_{\bar{x}} r_{\bar{y}}(P')$ and $r_{\bar{x}} r_{\bar{z}} r_{\bar{y}}(P')$, but, by induction, are equal. This concludes the proof that if π and σ are type 1 Knuth equivalent, then π and σ are P-equivalent.

To finish with the forward direction, we show that $\pi \overset{2}{\cong} \sigma \implies \pi \overset{1}{\cong} \sigma$, and by what we have shown thus far implies $\pi \overset{P}{\cong} \sigma$. With this goal in mind, observe the following chain of implications,

$$\begin{aligned}
\pi \overset{2}{\cong} \sigma &\implies \pi^r \overset{1}{\cong} \sigma^r && \text{by definition}\\
&\implies P(\pi^r) = P(\sigma^r) && \text{just proved}\\
&\implies P(\pi)^t = P(\sigma)^t && \text{Theorem 2.12}\\
&\implies P(\pi) = P(\sigma) && \text{by definition}\\
&\implies \quad \pi \overset{P}{\cong} \sigma. && \text{by definition}
\end{aligned}$$

This concludes the forward direction of the proof.

Backward: $\pi \overset{P}{\cong} \sigma \implies \pi \overset{K}{\cong} \sigma$.

Let $\pi \overset{P}{\cong} \sigma$ and $P(\pi) = P$. We will show is that $\pi \overset{K}{\cong} \pi_P$, which would prove the backwards direction, since equivalence relations are transitive.

We use induction (again) on the number of elements n in π. The base case is simply when $\pi = x_1\, x_2\, x_3 \in S_3$. Note that we already showed that $\pi \overset{P}{\cong} \pi_p \implies \pi \overset{K}{\cong} \pi_p$ in example 1.13. Now, assume that any permutation π' with n elements has the property $\pi' \overset{P}{\cong} \pi'_P \implies \pi' \overset{K}{\cong} \pi'_p$ and let π have $n+1$ elements with x_{n+1} being the last element in the permutation π. Thus, we can express π in the following way, $\pi = \pi'\, x_{n+1}$, where π' is the permutation with the first n elements of π. By induction we have $\pi' \overset{K}{\cong} \pi_{P'}$ and thus all we need to show is,

$$\pi'\, x_{n+1} \overset{K}{\cong} \pi_P.$$

We can see that this must be the case by considering P''s rows and demonstrating that Knuth relations can be used to perform a mock row insertion. Let the rows of P' be given by $R_1, R_2, \cdots, R_t$. Furthermore, let $R_1 = p_1\, p_1\, \cdots\, p_k$ and have x enter P' in cell $(1, j)$. Hence,

$$p_1 < p_2 < \cdots < p_{j-1} < x < p_j < \cdots < p_k$$

Thus, we have the sequence of Knuth relations,

$$\begin{aligned}
\pi_{P'}x &= R_t \cdots R_2 p_1 p_2 \cdots p_{j-2} p_{j-1} p_j \cdots p_{k-2} \boxed{p_{k-1}}\boxed{p_k\ x} \\
&\overset{1}{\cong} R_t \cdots R_2 p_1 p_2 \cdots p_{j-2} p_{j-1} p_j \cdots p_{k-2} \boxed{p_{k-1}}\boxed{x\ p_k} \\
&\overset{1}{\cong} R_t \cdots R_2 p_1 p_2 \cdots p_{j-2} p_{j-1} p_j \cdots \boxed{p_{k-2}}\boxed{x\ p_{k-1}}\, p_k \\
&\ \vdots \\
&\overset{1}{\cong} R_t \cdots R_2 p_1 p_2 \cdots p_{j-2} p_{j-1} p_j x \cdots p_{k-2} p_{k-1} p_k x \\
&\overset{2}{\cong} R_t \cdots R_2 p_1 p_2 \cdots p_{j-2} \boxed{p_j\ p_{j-1}}\boxed{x} \cdots p_{k-2} p_{k-1} p_k \\
&\overset{2}{\cong} R_t \cdots R_2 p_1 p_2 \cdots \boxed{p_{j-2}\ p_j}\boxed{p_{j-1}} x \cdots p_{k-2} p_{k-1} p_k \\
&\ \vdots \\
&\overset{2}{\cong} R_t \cdots R_2 p_j\, \underbrace{p_1 p_2 \cdots p_{j-2} p_{j-1} x \cdots p_{k-2} p_{k-1} p_k x}_{\text{first row of } P = r_x(P')}
\end{aligned}$$

Notice that the end of $\pi_{P'}x$ is the first row of $P = r_x(P')$. Furthermore, what remains to the left of what is in the first row of P is set up to perform a pseudo row-insertion of p_j into R_2. We can do this using the same steps as above. Thus, we can perform a pseudo row-insertion of x into P' using Knuth relations and the row word of P. This concludes the proof. □

5. Viennot's Geometric Construction

There is yet another way to think about the RS algorithm, known as Viennot's geometric construction. To begin, we'll consider an example. Let's take the permutation we used in a previous example and first write out the steps we would follow using the RS algorithm. Then, we will go through Viennot's construction.

First, we take the permutation,

$$\pi = \begin{pmatrix} 1 & 2 & \cdots & n \\ \pi(1) & \pi(2) & \cdots & \pi(n) \end{pmatrix} \tag{⚡}$$

and run through the RS algorithm and algorithm 2 to construct both the insertion and recording tableaux. These give the sequence of tableaux,

$P_1 = 4$	$P_2 = \begin{array}{cc} 4 & 5 \end{array}$	$P_3 = \begin{array}{ccc} 4 & 5 & 7 \end{array}$	$P_4 = \begin{array}{ccc} 2 & 5 & 7 \\ 4 & & \end{array}$	$P_5 = \begin{array}{ccc} 2 & 3 & 7 \\ 4 & 5 & \end{array}$	$P_6 = \begin{array}{ccc} 2 & 3 & 6 \\ 4 & 5 & 7 \end{array}$	$P = \begin{array}{ccc} 1 & 3 & 6 \\ 2 & 5 & 7 \\ 4 & & \end{array}$
$Q_1 = 1$	$Q_2 = \begin{array}{cc} 1 & 2 \end{array}$	$Q_3 = \begin{array}{ccc} 1 & 2 & 3 \end{array}$	$Q_4 = \begin{array}{ccc} 1 & 2 & 3 \\ 4 & & \end{array}$	$Q_5 = \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & \end{array}$	$Q_6 = \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array}$	$Q = \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & & \end{array}$

TABLE 1

Next, take each pair i and $\pi(i)$ from (↯), and interpret them as an ordered pair $(i, \pi(i))$, ready to be plotted on a graph as in Fig. 1.

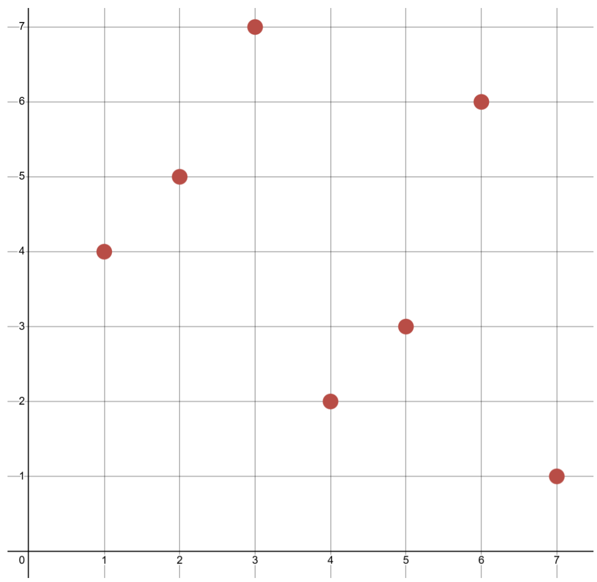


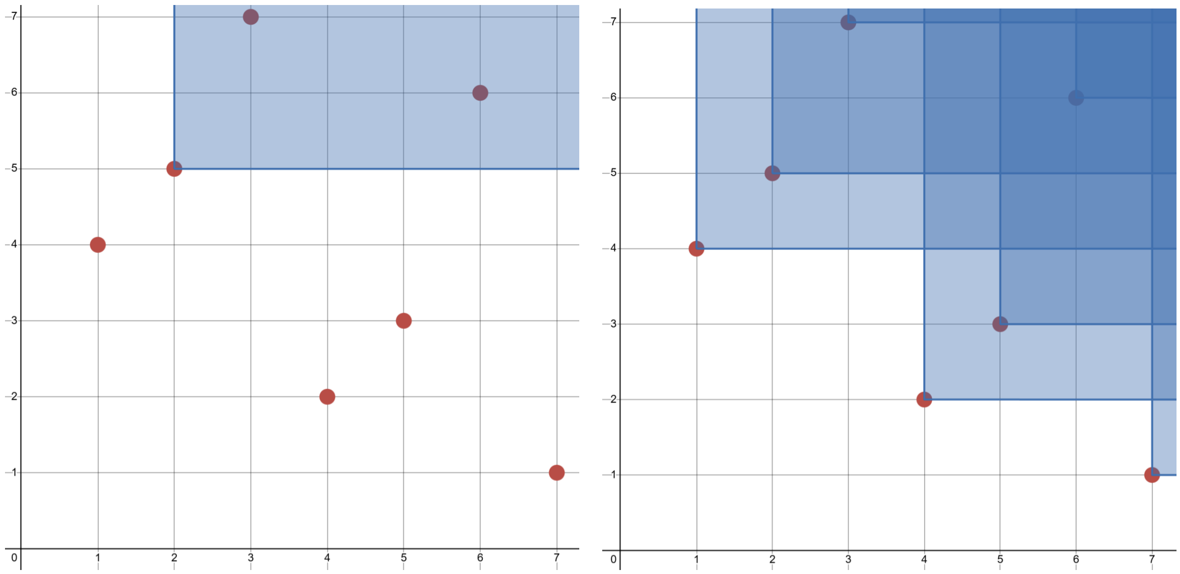
FIGURE 1

This results in a unique plot for each permutation. We claim that the tableaux,

$$P = \begin{array}{ccc} 1 & 3 & 6 \\ 2 & 5 & 7 \\ 4 & & \end{array} \qquad \text{and} \qquad Q = \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & & \end{array}$$

are hidden in the above plot.

Our goal is to find the tableaux P and Q using Fig. 1. The key idea is to shade different regions of our graph in order to determine the tableaux P and Q. There are a few ways to determine which portion of the graph we need to shade. One of them is to think of each point on the graph as the origin of its own coordinate system, and then shade in its first quadrant. For example, consider the point $(2, 5)$ in the graph, Fig. 1. We shade the pseudo first quadrant of $(2, 5)$ in Fig. 2a.



(A) In this example, we have chosen the point $(2, 5)$. Then, by thinking of $(2, 5)$ as the origin of its own coordinate system, we shade its first quadrant. In other words, we shade the portion of the graph that is both to the right and above $(2, 5)$.

(B) This is what we get after shading all the regions using the process described.

FIGURE 2

Note, we can also think of this process as shading in the portion of the graph that is both to the right and above the point $(2, 5)$.

We are not done shading Fig 1. We must repeat this process for every point in Fig. 1. More precisely, we must shade every point's first quadrant, in Fig. 1. When we do this, we end up with two separate regions: a shaded and an unshaded region, as shown in Fig. 2b.

We are now interested in the *border* between the shaded and unshaded regions of Fig. 2b. We call this border ***the first shadow line*** and we label it L_1. In Fig. 3a, the first shadow line, L_1, is the thick black line.

Next, remove all the points that are on the line L_1 as well as the area that we shaded because of those points, as shown in Fig. 3b.

There is now a new, smaller shaded region. As before, focus on the line that separates the new shaded and unshaded regions. This line, which we call **the second shadow line**, is labeled by L_2 and in the Fig. 3c, L_2 is the thick green line.

Continue in this manner, first eliminate the points on the line L_2 and the area they shade. Next, draw a new shadow line and label it L_3. Then remove the points on line L_3, and continue until there is no shaded region and hence no new shadow line to draw.

In the Fig. 3, we see that we only need one more shadow line, L_3 in orange, to reach the point where there is no shaded region left.

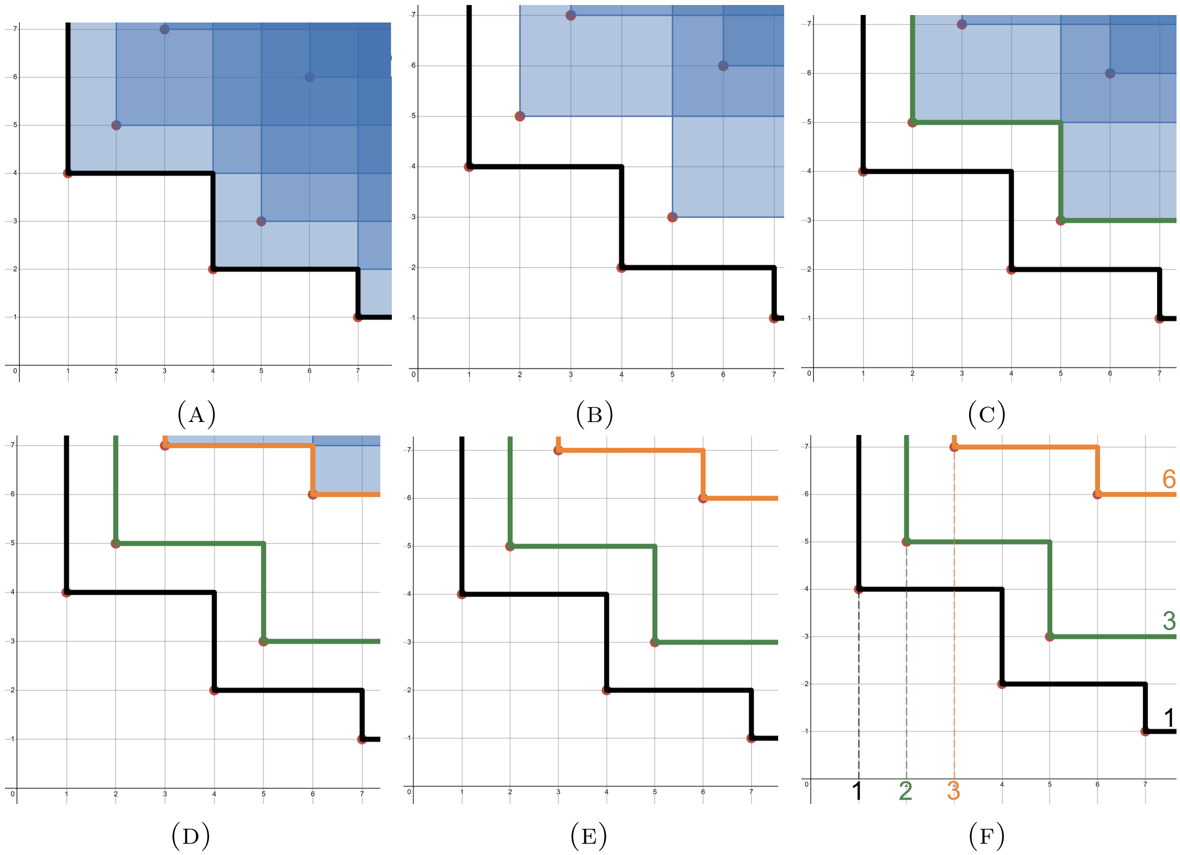


FIGURE 3. After constructing all the shadow lines for our permutation given in Equation (ϟ), we get Fig. 3f.

Recall our goal is to find the tableaux P and Q given Fig. 1. Take a look at each shadow line, and observe that they all have one horizontal and one vertical ray. Denote the coordinates of these rays,

$$\begin{aligned} x_{L_i} &:= \text{the } x-\text{coordinate of } L_i\text{'s vertical ray} \\ y_{L_i} &:= \text{the } y-\text{coordinate of } L_i\text{'s horizontal ray.} \end{aligned}$$

Here's the amazing part: the values of y_{L_i} and x_{L_i} in Fig. 3f match the first row of P and Q, respectively!

$$y_{L_1}\, y_{L_2}\, y_{L_3} = \mathbf{1}\ \mathbf{3}\ \mathbf{6}, \qquad P = \begin{array}{ccc} \mathbf{1} & \mathbf{3} & \mathbf{6} \\ 2 & 5 & 7 \\ 4 & & \end{array}$$

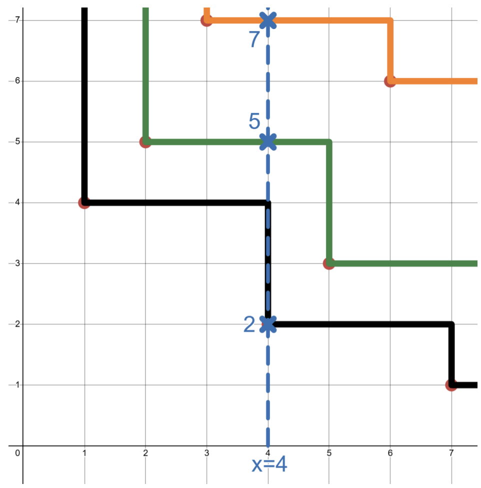
$$x_{L_1}\, x_{L_2}\, x_{L_3} = \mathbf{1}\ \mathbf{2}\ \mathbf{3}, \qquad Q = \begin{array}{ccc} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ 4 & 5 & 6 \\ 7 & & \end{array}$$

How awesome is that! However, to prove that this always happens, a lemma is required.

Lemma 5.1. Let the shadow band diagram of $\pi = x_1 x_2 \cdots x_n$ be constructed. Suppose the vertical line $x = k$ intersects i different shadow lines. Denote the y_j coordinate of the lowest point of intersection with L_j. Then the first row of $P_k = P(x_1 x_2 \cdots x_k)$ is

$$R_1 = y_1 y_2 \cdots y_i$$

Example 5.2. Before we get to the proof, let's consider a quick example using the same permutation $\pi = \left(\begin{smallmatrix}1&2&3&4&5&6&7\\4&5&7&2&3&6&1\end{smallmatrix}\right)$. Consider the vertical line $x = 4$:



We see from the figure that the line $x = 4$ intersects all three shadow lines. The y-coordinates of these intersections are 2, 5, 7. Recall that $P_{k=4}$ in table 1 is equal to $P_4 = \begin{smallmatrix}2&5&7\\4&&\end{smallmatrix}$. Notice that $P_{k=4}$'s first row is exactly the y-coordinates of these intersections 2, 5, 7. ■

Proof of Lemma. We induct on k. The base case is the trivial case, since $P_0 = \emptyset$. Now, assume that the result holds for the line $x = k$ and consider the vertical line $x = k = 1$. We break this situation into two cases based on how the element that we will row insert into P_k, which is x_{k+1}, compares to the values in the first row of P_k.

Case 1: $x_{k+1} > y_i$.

In this situation the point $(k+1, \pi(k+1)) = (k+1, x_{k+1})$ must sit above the shadow lines before it. For this reason, $(k+1, x_{k+1})$ starts a new shadow line. Therefore, the vertical line $x = k+1$ still intersects the same shadow lines as before, and now also intersects this new shadow line at $(k+1, x_{k+1})$. This means that intersection values don't change, except for adding x_{k+1} to the end.

Turing our attention to the partial tableau P_{k+1}, not that x_{k+1} is larger than all the elements in the first row of P_k, thus x_{k+1} simply enters to the right of y_i and the first row of P_{k+1} is simply $R_1 x_{k+1}$.

Case 2: $y_1 < \cdots < y_{j-1} < x_{k+1} < y_j < \cdots y_i$.

In this situation, $(k+1, x_{k+1})$ does not start a new shadow line, but it does lie on the line L_j. We can see this is the case because we already know that the vertical line

$x = k$ intersects the previous shadow lines at the y-values $y_1, \cdots, y_i$. But, now we are adding in a new point that is above $y_1, \cdots, y_{j-1}$ and below $y_j, \cdots, y_i$. The only possible line that $(k+1, x_{k+1})$ could sit on is the one that also has (k, y_j) on it. This is the line L_j. All this together tells us that the vertical line at $x = k+1$ will intersect shadow lines at $y_1, \cdots, y_{j-1}$ and then x_{k+1} and then $y_{j+1}, \cdots, y_i$. Thus, we have the list: $y_1, \cdots, y_{j-1} x_{k+1} y_{j+1}, \cdots, y_i$.

Our situation also tells us that when preforming $r_{x_{k+1}}(P_k)$ that x_{k+1} will bump out y_j from the first row and give will prove the result. □

Here we go!

Corollary 5.3. If the permutation π has the RS tableaux (P, Q) and show lines L_i. Then, for all i we have,

$$P_{1,i} = y_{L_i} \qquad \text{and} \qquad Q_{1,i} = x_{L_i}.$$

Proof. The statement regarding the P tableau is a special case of the previous lemma, where $k = n$. However, for Q we will use induction on k.

The base case is again the trivial case. Assume that the result is true for $x = k-1$. Consider now the process of row-inserting x_k in P_{k-1} and how this will affect Q_{k-1}'s first row.

We know that Q_{k-1}'s first row is only affected if and only if x_k is added to the end of the first row, which happens only when x_k is greater than $P_{1,i}$ for all i. However, recall that we argued in the former proof that this meant that the line $x = k$ intersects a new shadow line L_i. Since the vertical line $x = k-1$ did not intersect L_i it must be that the vertical ray of L_i has a x-value of k. This concludes the proof. □

There is a way to get the rest of the rows of P and Q using the shadow diagram of π. But we have to change our perspective. Instead of focusing on the points that we plotted, we have to turn our attention to the **northeast corners** of the shadow lines. Then, we use these to create new shadow lines, $L_i^{(2)}$. and determine the x- and y-coordinates, as shown in Fig. 4.

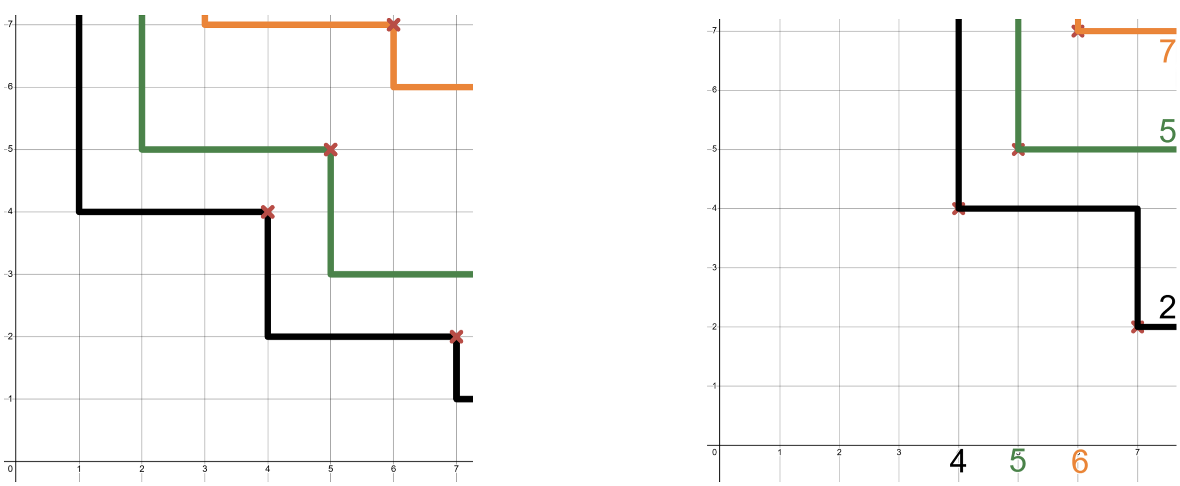


FIGURE 4

Notice that these new points are the second row of P and Q, in a manner similar to before. Then, in order to construct the full tableau, we keep turning our attention to the new northeast corners. See Fig. 5 for the final step using our example permutation.

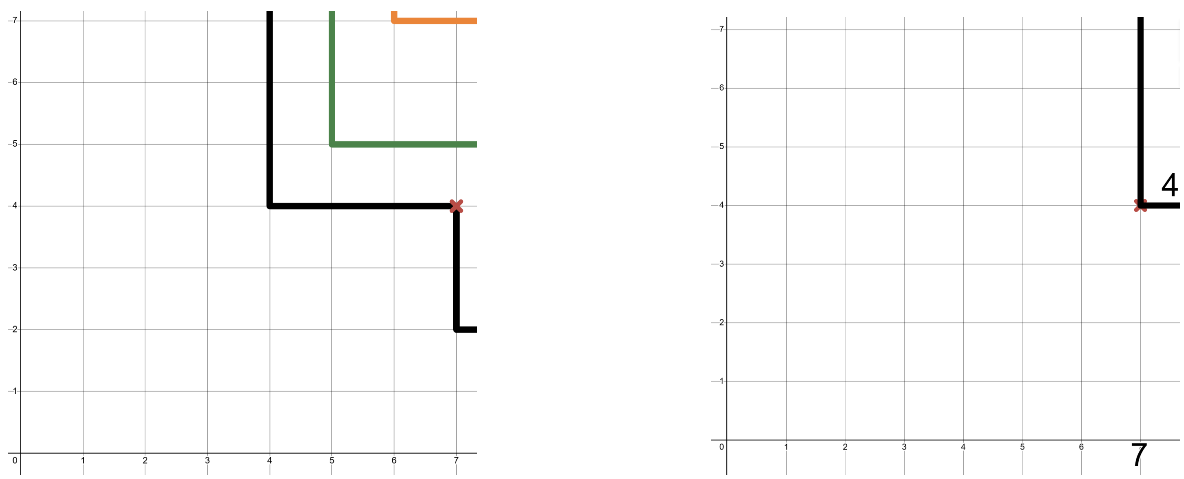


FIGURE 5

This is summed up in the following definition and theorem.

Definition 5.4. The i^{th} skeleton of $\pi \in S_n$ is denoted $\pi^{(i)}$ and defined inductively by $\pi^{(1)} = \pi$ and,

$$\pi^{(i)} = \begin{matrix} k_1 & k_2 & \cdots & k_m \\ l_1 & l_2 & \cdots & l_m \end{matrix}$$

where the ordered pairs (k_j, l_j) are the northeast corners of the shadow line as described in the paragraphs preceding this definition. The shadow lines for $\pi^{(i)}$ are denoted $L_j^{(i)}$.

Theorem 5.5. Suppose that $\pi \xrightarrow{\text{R–S}} (P, Q)$. Then $\pi^{(i)}$ is a partial permutation such that,

$$\pi^{(i)} \xrightarrow{\text{R–S}} (P^{(i)}, Q^{(i)})$$

Where $P^{(i)}$ and $Q^{(i)}$ consist of the row i and all the other rows below row i. Moreover,

$$P_{i,j} = y_{L_j^{(i)}} \qquad \text{and} \qquad Q_{i,j} = x_{L_j^{(i)}}$$

6. Small Results and Conjectures

6.1. **Specific Tableaux and Knuth Graphs.**

In this section, we put some *Knuth graphs* for different shape tableaux. First, however, we must define what a Knuth graph is.

Definition 6.1. Let $\pi_a \in S_n$ for all $a \in \{1, 2, 3, \cdots, k\}$ such that

$$\pi = \pi_1 \overset{i}{\cong} \pi_2 \overset{j}{\cong} \cdots \overset{l}{\cong} \pi_k$$

where $i, j, \cdots, l \in \{1, 2\}$. We construct a Knuth graph for these permutations as follows.

(1) First, each of the permutations is located at the vertex of a graph.
(2) Second, we draw an edge from π_a to π_b if and only if π_a is related to π_b by a single Knuth relation.
(3) Third, label the edge connecting π_a to π_b using the convention,
    - if $\pi_a \overset{1}{\cong} \pi_b$ label the edge by $K1$,
    - if $\pi_a \overset{2}{\cong} \pi_b$ label the edge by $K2$, or
    - if $\pi_a \overset{1}{\cong} \pi_b$ and $\pi_a \overset{2}{\cong} \pi_b$, then label the edge by KB.

▨

6.2. **Knuth Graphs for S_4.**

The Knuth graphs for all permutations in S_4 are given in Figure 6. The two numbers that are swapped by a Knuth relation are highlighted in blue.

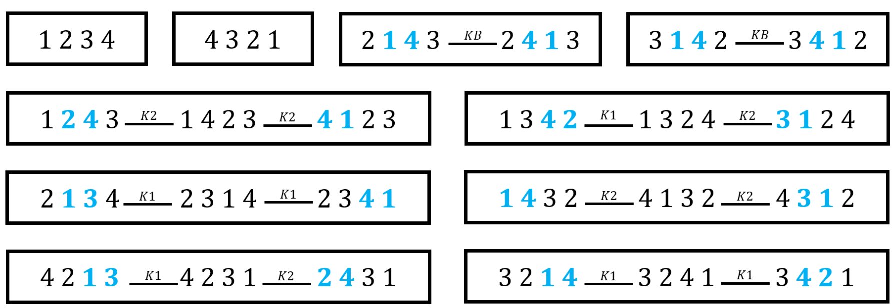
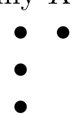


Figure 6. These are all the Knuth graphs for the permutations in S_4.

Some observations regarding Fig. 6 are the following:

(1) There are 10 equivalence classes.
(2) There are an equal number of $K1$ and $K2$ moves made when you count all the moves made throughout the 10 graphs, (this makes sense by a symmetry argument).
(3) The equivalence classes have sizes 1, 2, and 3.
(4) All the graphs are *trees*, i.e. there are no loops.

(5) There is are sets that have only $K1$ moves, have only $K2$ moves, and have both $K1$ and $K2$ moves for the shapes and .

(6) The tableau for the sets that have two permutations that are related by KB moves has shape $\begin{matrix}\bullet & \bullet \\ \bullet & \bullet\end{matrix}$.

Definition 6.2. Let $k \in \mathbb{N}$. If the shape of the tableau is $(k, 1, 1, \cdots, 1)$, then we say that P has shape Γ_k.

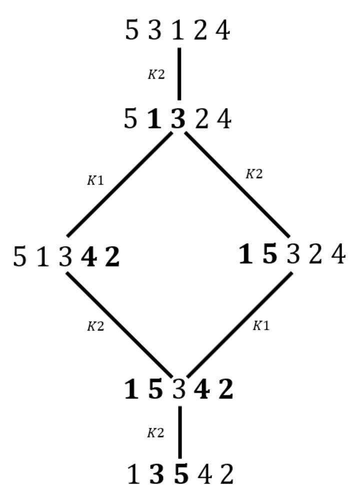
Conjecture 6.3. *If the shape of P is Γ_k, the the Knuth graph a tree.*

Claim: False

Proof. We will show that the permutation $\pi = 5\,3\,1\,2\,4$ is a counter-example. First note that $\pi = 5\,3\,1\,2\,4$'s tableau is the following,

$$P(\pi) \;=\; \begin{matrix} 1 & 2 & 4 \\ 3 & & \\ 5 & & \end{matrix}$$

Furthermore, $P(\pi)$ has shape $\lambda = (3, 1, 1)$. Part of the Knuth graph for π is shown below. Note that we have formed a *loop*.



□

Conjecture 6.4. If the shape of $P(\pi)$ is Γ_k, for $k \in \mathbb{N}_0$, then the Knuth Graph of π has no KB moves.

Claim: True

Theorem 6.5. *The shape of $P(\pi)$ is Γ if and only if π has no KB moves in its Knuth graph.*

Proof. ($\Longrightarrow$) For the forward direction, we will prove the contrapositive, namely, *if there is a KB move in π's Knuth graph, then the shape of $P(\pi)$ is not* Γ. With this goal in mind, let there be a KB move in π's Knuth graph. Therefore, we can write π (up to Knuth equivalence) as,

$$\pi = a_1a_2 \cdots y\, x\, z\, \overline{y} \cdots a_n,$$

where $x < y, \overline{y} < z$. Thus, π is Knuth equivalent to,

$$\pi = a_1a_2 \cdots y\, x\, z\, \overline{y} \cdots a_n \overset{KB}{\cong} a_1a_2 \cdots y\, z\, x\, \overline{y} \cdots a_n = \sigma,$$

We will show that having $yxz\overline{y}$ as part of the permutation π will result in a structure that is not Γ.

First, note that there is no way to empty a cell using the RS algorithm. Although the element occupying a cell can change throughout the algorithm, once a cell is filled, it will remain filled. Next, we start running the RS algorithm on π until we reach the element x. Let $\overline{P}$ denote the partial tableau that we get after row-inserting all the elements up to and including y. If $\overline{P}$ already has cell $(2,2)$ filled in, then we are done. So assume that $\bar{P}$ has the shape $(t, 1 \cdots, 1)$. We will focus on the insertions $r_{\overline{y}}r_zr_x(\overline{P})$ and see what happens to $\overline{P}$.

Consider $r_x(\overline{P})$. We will focus on the case where y is located in the first row, as the proof is similar when y is in the first column when you work with π^r. When we row insert x into $\overline{P}$, it must be the case that x bumps out some $x' \leq y$. As a result, two different situations can occur: (1) x' enters the second row in cell $(2,2)$, in which case we would be done, or (2) x' enters cell $(2,1)$ by either bumping out the previous occupant of that cell or by entering that cell directly in the case that cell $(2,1)$ was empty. Since we would be done in the first case, let's assume situation (2) occurs. Therefore, we end up with x in the first row, x' in the first column of the second row of $r_x(\overline{P})$, and $x < x' \leq y$.

Moving on to row-inserting z into $r_x(\overline{P})$, we can either end up with z entering the last cell on the first row, or z bumping out some z' into the second row. If the latter happens, then either z' enters cell $(2,2)$ and we would be done, or z' bumps out x' from cell $(2,1)$. However, we claim that this second situation cannot happen since this would imply $x' > z'$, which contradicts that $x' \leq y < z < z'$. Thus, if z doesn't enter the first row in the last cell, we're done. So, assume that z enters the last cell of the first row.

The last step is to consider what happens when we row insert $\overline{y}$ into $r_zr_x(\overline{P})$. Since we know that z is in the first row, we know that $\overline{y}$ will bump out some $y'' \leq z$ from the first row. We have two cases to consider: (1) when $y < \overline{y}$ and (2) $y > \overline{y}$. In the first case, by row-inserting y'' into the second row, it must be the case that y'' enters cell $(2,2)$. This is because the occupant of cell $(2,1)$ is x' and $x' \leq y$. Together with $y < \overline{y}$ we see that $x' \leq y < \overline{y}$ and hence y'' enters into cell $(2,2)$. In the second case, suppose that we have $\overline{y} < y$. In this situation, consider the series of logical consequences of facts that we've already deduced: first,

$$\pi = a_1a_2 \cdots yxz\overline{y} \cdots a_n \overset{KB}{\cong} a_1a_2 \cdots yzx\overline{y} \cdots a_n = \sigma.$$

Second, that $P(\pi) = P(\sigma)$ and third that if cell $(2,2)$ is filled in $P(\pi)$ then it's also filled in for $P(\pi^r) = P(\pi)^t$. Now, consider

$$\pi^r = a_n a_{n-1} \cdots \overline{y} z x y \cdots a_1 \overset{KB}{\cong} a_n a_{n-1} \cdots \overline{y} x z y \cdots a_1 = \sigma^r.$$

From this, we see that σ^r is in the same situation that we just finished when we were considering π and $y < \overline{y}$. Thus, cell $(2,2)$ is filed in for σ^r and hence cell $(2,2)$ is filled in for σ and thus cell $(2,2)$ is filled in for π. This concludes our proof of the forward direction.

($\Longleftarrow$) We must now show that if there are no KB moves in π's Knuth graphs, then the shape of $P(\pi)$ is Γ. We will use the same strategy we used for the forward direction of the proof and prove the contrapositive, *if the shape of $P(\pi)$ is not Γ, then there are KB moves.*

Let the shape of $P(\pi)$ be $\lambda = (\lambda_1, \lambda_2, \lambda_3, \cdots, \lambda_n)$ for $\lambda_i \in \mathbb{N}_0$. Focus on the first two rows of $P(\pi)$ and denote them,

$$P(\pi)'\text{s first two rows} \qquad \begin{matrix} a_1 & a_2 & \cdots & a_m & \cdots & a_n \\ b_1 & b_2 & \cdots & b_m & & \end{matrix} .$$

Where $m \geq 2$ since we need $P(\pi)$ to have a shape different from shape Γ. Recall that the row reading word of $P(\pi)$ is such that $\pi_P \overset{K}{\cong} \pi$, so that we can focus on showing that π_P has a KB move. We will do so by considering the subsequence $b_{m-1} b_m a_1 a_2$ of π_P. Note that we have a series of inequalities:

$$a_1 < b_1 < b_{m-1} < b_m$$
$$a_1 < a_2 < b_2 \leq b_m$$

Thus, we have

$$a_1 < a_2, b_{m-1} < b_m.$$

Note that this means we can perform a KB move on π_P since it contains the subsequence $b_{m-1} b_m a_1 a_2$. This concludes the proof. □

6.3. **All the Knuth Graphs for a Tableau with Shape** $(3,1,1)$**.**

See Figure 7 on the following page.

Observations/Conjectures

- **Observation**: Every Knuth graph has the same structure.
- **Conjecture**: Every permutation that has the same tableau shape (for this family of Γ_k-shaped tableaux) has the same Knuth graph structure.
- **Observation**: The number of K1 and K2 moves are even numbers for all Knuth graphs.
- **Observation**: When you write out all the Knuth graphs for the tableaux of shape $(3,1,1)$, you get all the ways to have a symmetric set of K1 and K2 moves.
- **Conjecture**: This last observation will always happen.

- **Observation**: A symmetry in the tableau seems to correlate with a symmetry in the Knuth graph.
- **Conjecture**: The tableau is symmetric across a diagonal line through the cells $(1,1)$, $(2,2)$, $\cdots$, (n,n) if and only if the Knuth graphs are symmetric.
- **Observation**: The row reading word of each tableau ends up at a *node* of the Knuth graph.

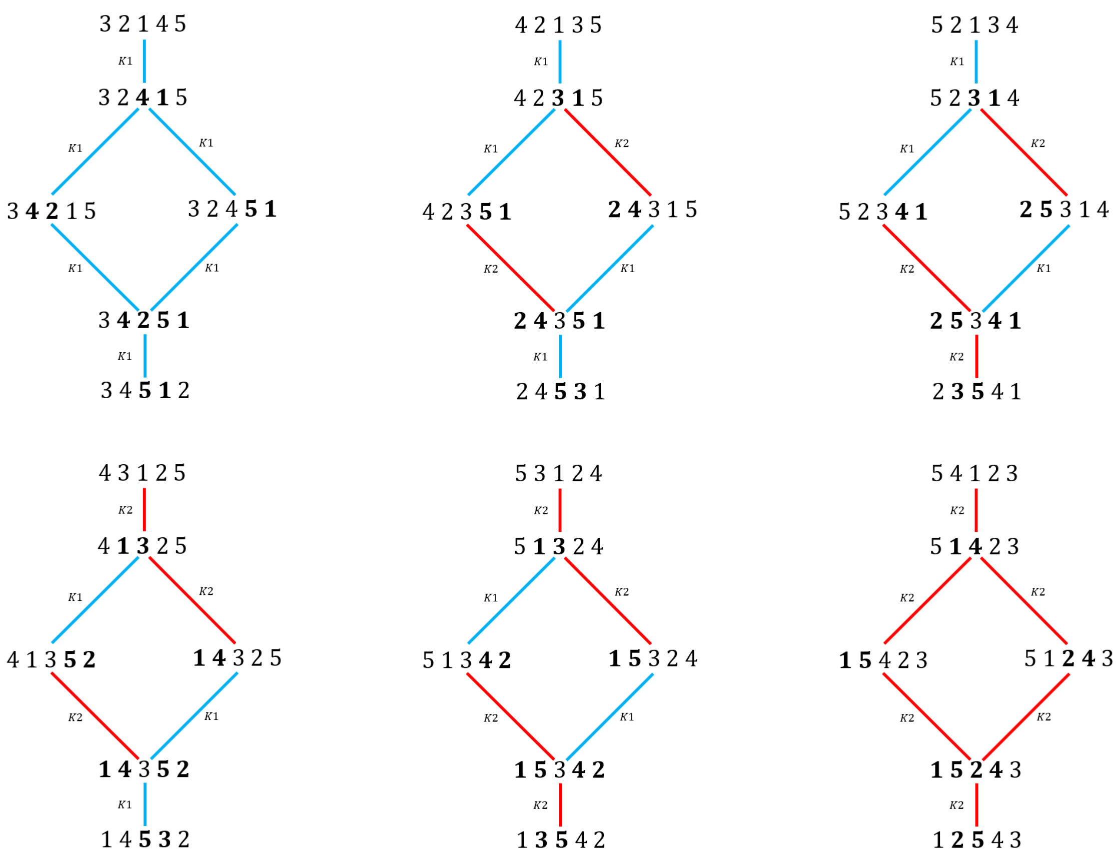


FIGURE 7. These are all the Knuth graphs for tableaux with shape $(3,1,1)$. The top vertex of the graphs is the row reading word of each tableau.

Lemma 6.6. The number of permutations in a Knuth graph of π equals the number of tableaux of a given shape.

Proof. Let $[\pi] := \{\sigma \ : \ \pi \overset{K}{\cong} \sigma\}$. Let $\alpha, \beta \in [\pi]$. Recall that the RS algorithm is also a bijection between permutations and (P, Q). Since $\alpha, \beta \in [\pi]$, we know that $P(\alpha) = P(\beta)$, so they have the same Ps and thus they must have different Qs. This means that the number of tableaux of shape P is less than or equal to $\#[\pi]$. $\square$

We conclude with some additional Knuth graphs where the tableau has shape Γ.

6.4. **Shape** $\lambda = (3,1,1,1)$.

In Figure 8, the top vertex for each Knuth graph is the row reading word of that tableau. Notice that all the row reading words have the same shape as the Knuth graph.

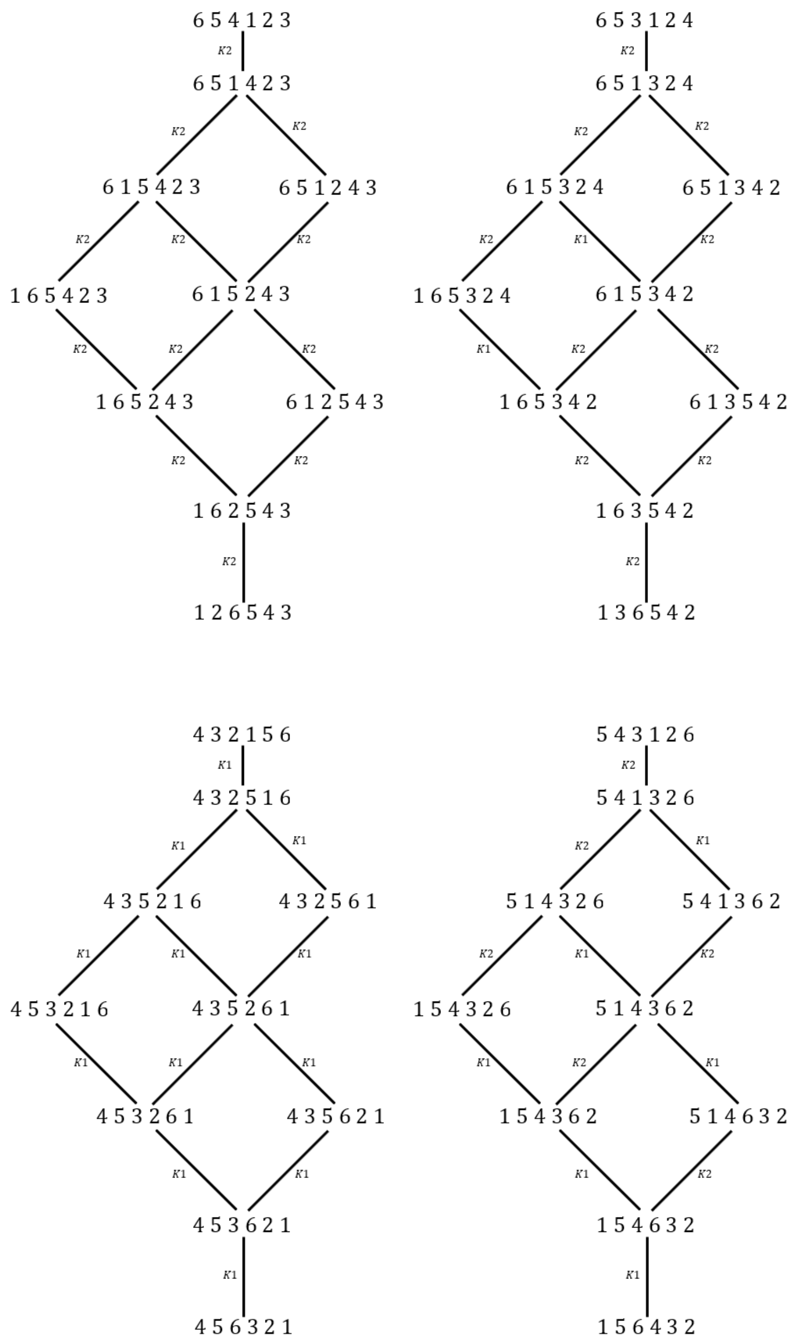


FIGURE 8. Some of the Knuth graphs for permutations whose tableau has shape $(3,1,1,1)$. The row reading word of each tableau is located at the top vertex.

References

[Sag01] Bruce E. Sagan. *The symmetric group*, volume 203 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 2001. Representations, combinatorial algorithms, and symmetric functions.

Department of Mathematics and Statistics, UMass Lowell