Factorization of the Robinson-Schensted-Knuth Correspondence

David P. Little

September 12, 2004

Abstract

In [4], a bijection between collections of reduced factorizations of elements of the symmetric group was described. Initially, this bijection was used to show the Schur positivity of the Stanley symmetric functions. Further investigations have revealed that our bijection has strong connections to other more familiar combinatorial algorithms. In this paper we will show how the Robinson-Schensted-Knuth correspondence can be decomposed into a sequence of applications of this bijection.

1 Introduction

In [4], a bijection between collections of reduced factorizations of elements of the symmetric group was described. This bijection was used to give a completely combinatorial proof of the Schur positivity of the Stanley symmetric functions [6]. One unexpected consequence of this bijection was a natural correspondence between permutations and pairs of standard Young tableaux. Further investigation revealed that this correspondence is identical to that of Robinson-Schensted-Knuth [2, 5].

The main contribution of this paper is to translate the Robinson-Schensted-Knuth correspondence into the context of reduced factorizations of the symmetric group. Specifically, we will present an algorithm called RSK which converts a generalized permutation into a pair of column strict tableaux. Our main result is the following:

Theorem 1 *Let π be a generalized permutation. If $RSK(\pi) = (R, S)$ and (P, Q) is the result of applying the Robinson-Schensted-Knuth correspondence to π then $R = P$ and $S = Q$.*

Our paper begins by describing the relevant theory of reduced factorizations of the symmetric group and it's connections to pattern avoiding permutations. Subsequent sections will review our bijection and show how to translate the process of row insertion into that of reduced words. We will then show how to implement the Robinson-Schensted-Knuth correspondence. As we will see, one benefit of our algorithm is that the symmetry property of Robinson-Schensted-Knuth is immediate. However, it is not immediately clear that RSK will produce tableaux of the same shape. A number of conjectures regarding our algorithm will be discussed at the end of the paper.

2 Reduced Words

We say that a word $w = w_1 w_2 \cdots w_l$ in the alphabet $\{1, 2, \ldots, n-1\}$ corresponds to the permutation $\sigma \in S_n$ if

$$\sigma = s_{w_1} s_{w_2} \cdots s_{w_l}$$

where s_i represents the transposition $(i, i+1)$. The word $w = w_1 w_2 \cdots w_l$ is said to be *reduced* if there does not exist another word $v = v_1 v_2 \cdots v_k$ such that v corresponds to the same permutation as w and $k < l$. The collection of all reduced words corresponding to σ is denoted $Red(\sigma)$. If $w = w_1 w_2 \cdots w_l \in Red(\sigma)$ then l is called the length of σ, denoted $l(\sigma)$. It is a simple exercise to show that $l(\sigma)$ is the number of inversions of σ, that is

$$l(\sigma) = |\{(\sigma_i, \sigma_j) \mid i < j \ \& \ \sigma_i > \sigma_j\}|.$$

One can also easily verify the following fundamental fact.

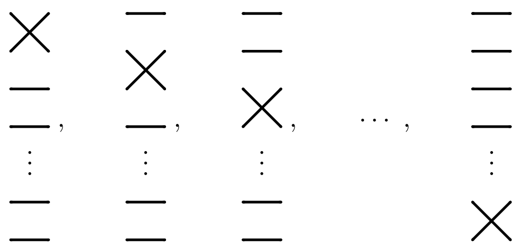
Lemma 2 *Any two reduced words corresponding to the same permutation are related by a sequence of commutativity relations of the form*

$$ij = ji \text{ if } |i - j| > 1$$

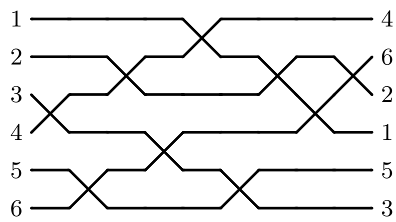
or

$$i(i+1)i = (i+1)i(i+1).$$

The *line diagram* of w, denoted $LD(w)$, is a graph of the trajectories of the numbers 1 through n as they are rearranged into the target permutation σ according to the transpositions $(w_i, w_i + 1)$. Formally, the line diagrams for the letters 1 through $n-1$ are the following.



The line diagram of an arbitrary word $w = w_1 \cdots w_l$ is given by the juxtaposition of $LD(w_1), LD(w_2), \ldots, LD(w_l)$. For example, the line diagram corresponding to the word $w = 352415232$ is given below.



Labeling the left endpoints of the lines with the numbers 1 through n from top to bottom allows us to easily identify the corresponding permutation. From the example above, we see that w corresponds to the permutation $(4, 6, 2, 1, 5, 3)$. Notice that the property of a word being reduced can be seen in the line diagram via the following lemma.

Lemma 3 *The word w is reduced if and only if no two lines in $LD(w)$ cross more than once.*

Proof. If two lines cross twice, then the corresponding letters of w can be removed without altering the permutation and thus w is not reduced. If no two lines cross twice, then each letter of w accounts for a unique inversion in the corresponding permutation. Moreover, if (α, β) is an inversion, then lines α and β must cross somewhere in the line diagram. Therefore the length of the word must be $l(\sigma)$ and thus w is reduced. $\square$

The *labeled circle diagram* of w, denoted $LCD(w)$, is an $n \times n$ array that records which letter of the word w created a particular inversion in the corresponding permutation σ. We begin by building the *circle diagram* of $\sigma = (\sigma_1, \ldots, \sigma_n)$, denoted $CD(\sigma)$. Starting with an $n \times n$ grid, label the rows 1 through n from top to bottom and label the columns $\sigma_1, \sigma_2, \ldots, \sigma_n$ from left to right. For each i, place a $\times$ in the row labeled i and column labeled i. Now place a $\bullet$ in each cell which is immediately to the right or directly below a $\times$. And finally, place a circle in each empty cell.

	4	6	2	1	5	3
1	○	○	○	×	•	•
2	○	○	×	•	•	•
3	○	○	•	•	○	×
4	×	•	•	•	•	•
5	•	○	•	•	×	•
6	•	×	•	•	•	•

By labeling the columns by the values $\sigma_1, \sigma_2, \ldots, \sigma_n$, we invite some confusion when referring to certain cells of the diagram. When we refer to column j, do we mean the jth column from the left or the column labeled j? For this reason we will use the convention that the (i, j)-*entry* of $CD(\sigma)$ refers to the cell in the ith row from the top and the jth column from the left. The phrases "row i" and "column j" will refer to the row labeled i (which is the same as the ith row from the top) and the column labeled j.

The code of a permutation is defined to be the vector which records the number of circles in each column. That is the code of σ, $c(\sigma)$, is given by

$$c(\sigma) = (c_1(\sigma), c_2(\sigma), \ldots, c_n(\sigma))$$

where $c_i(\sigma) =$ the number of circles in the ith column (from left to right) of $CD(\sigma)$. The shape of σ, $\lambda(\sigma)$ is given by the decreasing rearrangement of $c(\sigma)$. For example, the code of $(4, 6, 2, 1, 5, 3)$ is $(3, 4, 1, 0, 1, 0)$ and therefore the shape is $(4, 3, 1, 1)$.

Notice that each circle in $CD(\sigma)$ corresponds to a particular inversion of σ. Suppose that there is a circle in row i and column j. This implies that the $\times$ in column j must be below row i and therefore $j > i$. Also, the $\times$ in row i must appear to the right of column j and thus i is to the right of j in σ. This means that the pair (j, i) is an inversion of σ. If $w \in Red(\sigma)$ then each letter of w is responsible for a particular inversion in σ. Therefore, to form $LCD(w)$, simply place a k in the circle in row i and column j if lines i and j are crossed in the kth position of $LD(w)$. The labeled circle diagram of the word 352415232 is shown below.

	4	6	2	1	5	3
1	(5)	(8)	(7)	×	•	•
2	(3)	(9)	×	•	•	•
3	(1)	(4)	•	•	(6)	×
4	×	•	•	•	•	•
5	•	(2)	•	•	×	•
6	•	×	•	•	•	•

If w is not reduced, we may still define the $LCD(w)$ using the following procedure. Place a k in row i and column j if lines i and j are crossed in the kth position of $LD(w)$ where line i was above line j prior to crossing them.

3 321-Avoiding Permutations

A permutation $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ is said to contain a *321-pattern* if there exists indices $i < j < k$ such that $\sigma_i > \sigma_j > \sigma_k$. A permutation σ is said to be *321-avoiding* if σ does not contain a 321-pattern. Billey, Jockusch and Stanley [1] showed that σ is 321-avoiding if and only if the circles in $CD(\sigma)$ form a Young diagram of French skew shape. The main goal of this section is to translate this notion of a 321-avoiding permutation into properties of line and labeled circle diagrams.

We begin by re-examining our construction of the circle diagram of σ. Recall that a circle in row i and column j indicates that the pair (j, i) is an inversion of σ. Therefore line i must cross line j from above in the line diagram of *any* reduced word corresponding to σ. Or equivalently, line j must cross line i from below. This leads us to the following definitions. We say that a line has *positive trajectory* if it always crosses another line from below. Similarly, a line is said to have *negative trajectory* if it always crosses another line from above. If a line does not cross any other line, then it is said to have *zero trajectory*. In the event that a line has positive, negative or zero trajectory, then the line is said to be *monotonic*.

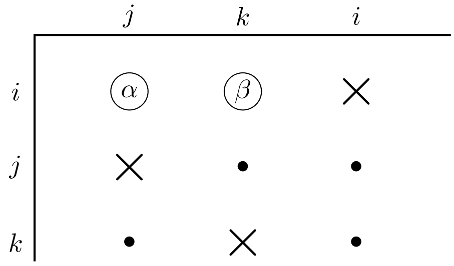
We can easily translate the above definitions into properties of $CD(\sigma)$, since they are independent of the reduced word. In particular, line i has negative trajectory if row i of $CD(\sigma)$ has at least one circle and column i has none. Similarly, line i has positive trajectory if column i of $CD(\sigma)$ has at least one circle and row i has none. And finally, line i has zero trajectory if row i and column i have no circles in them.

Lemma 4 *Let $w \in Red(\sigma)$. Then σ is 321-avoiding if and only if every line in $LD(w)$ is monotonic.*

Proof. First suppose σ contains the 321-pattern (α, β, γ). Then line β cannot be monotonic since it must cross either line α from below and line γ from above or vice versa. Now suppose that line α of $LD(w)$ is not monotonic. If line α crosses line β from below and subsequently crosses line γ from above then (γ, α, β) is a 321-pattern. Otherwise, if line α crosses line β from above and subsequently crosses line γ from below then (β, α, γ) is a 321-pattern. □

Lemma 5 *Let $w \in Red(\sigma)$ where σ is a 321-avoiding permutation. Then the labels of $LCD(w)$ are strictly increasing from left to right in each row and from bottom to top in each column.*

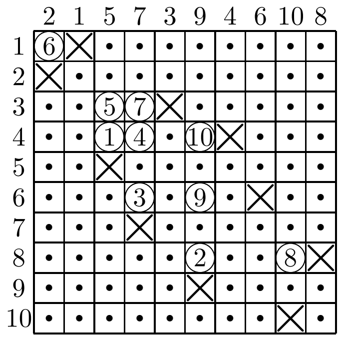
Proof. Let $w = w_1 w_2 \cdots w_l$ correspond to the 321-avoiding permutation σ. Suppose that w_α interchanges lines i and j and w_β interchanges lines i and k in $LD(w)$ with $i < j < k$. Therefore, the labeled circle diagram of w must appear as follows.



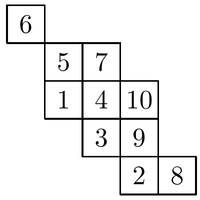
Note that j must appear before k in σ, otherwise (k, j, i) would constitute a 321-pattern. Thus lines j and k cannot cross. Therefore line i must cross line j prior to crossing line k which implies that $\alpha < \beta$. In other words, in any given row in $LCD(w)$, the labels must increase from left to right. A similar reasoning shows why the columns increase from bottom to top. $\square$

This last lemma combined with the result of Billey, Jockusch and Stanley allows us to think of $LCD(w)$ as a standard Young tableau of French skew shape whenever w corresponds to a 321-avoiding permutation. Specifically, by removing all cells of $LCD(w)$ that are void of circles we form the tableau $T(w)$.

$LCD(w) =$



$T(w) =$



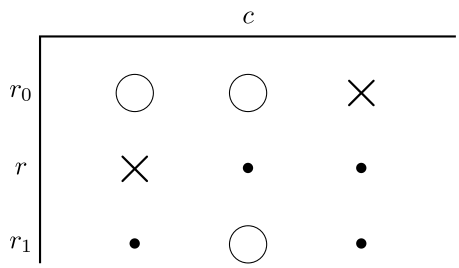
Lemma 6 *Let $w = w_1 w_2 \cdots w_l$ be a reduced word corresponding to a 321-avoiding permutation and let $v = w_1 \cdots w_{i-1} x w_i \cdots w_l$ for any letter x and index $1 \leq i \leq l+1$. Suppose that x interchanges lines α and β in $LD(v)$. Then v is reduced if lines α and β have the same trajectory in $LD(w)$ or at least one of these lines has zero trajectory in $LD(w)$.*

Proof. If lines α and β have the same trajectory in $LD(w)$ or one of them has zero trajectory, then they cannot cross in w. Therefore these lines cross exactly once in $LD(v)$. If any other pair of lines were to cross more than once in $LD(v)$ then w would not have been reduced. Therefore no pair of lines in $LD(v)$ cross more than once and thus v is reduced. $\square$

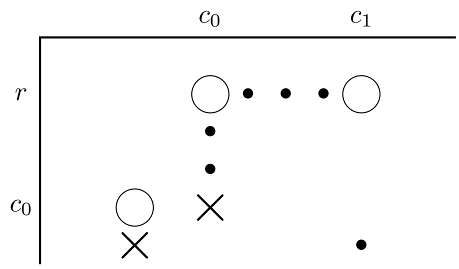
If σ is a 321-avoiding permutation different from the identity and the circles of $CD(\sigma)$ form a non-skew Young diagram, then σ will be referred to as a *Young permutation.* Any reduced word that corresponds to a Young permutation will be called a *Young word.* If w is a Young word, then by definition, $T(w)$ is a standard Young tableau of non-skew shape. One simple consequence of this observation is that all reduced words corresponding to σ must start with the same letter. Another more noteworthy consequence is the following lemma.

Lemma 7 *Let w be a reduced word. Then w is a Young word if and only if there exists integers $I \leq J < K$ such that in $LD(w)$ lines I through J have negative trajectory, lines $J+1$ through K have positive trajectory and all other lines have zero trajectory.*

Proof. Let w be a Young word. The rows of $LCD(w)$ which contain circles must appear consecutively in $LCD(w)$ for the following reason. Suppose that the upper-left most circle appears in row r_0 and column c and the lower-left most circle appears in row r_1 and column c. If there exists an r between r_0 and r_1 such that the cell in row r and column c does not contain a circle, then the $\times$ in row i must appear to the left of column c. Which in turn implies that there is a circle in row r_0 directly above this $\times$. This contradicts our choice of c as illustrated below. Since rows in $LCD(w)$ with circles in them correspond to lines in $LD(w)$ with negative trajectory, the lines of negative trajectory appear consecutively.



Now consider the columns of $LCD(w)$ which contain circles. These columns must be labeled consecutively for the following reason. Suppose that row r contains the lower-most circle in each column. Furthermore, assume that there are no circles between the circles in row r and columns c_0 and c_1. The $\times$ in column c_0 appears in row c_0. If the $\times$ in column c_1 does not appear in row c_0+1 then the $\times$ in row c_0+1 must appear to the left of column c_0. This implies that there is a circle in row c_0, which contradicts our choice of r, as illustrated below. Therefore $c_1 = c_0 + 1$. Since columns in $LCD(w)$ with circles in them correspond to lines in $LD(w)$ with positive trajectory, the lines of positive trajectory appear consecutively.



Since the first cross in $LD(w)$ involves the last line of negative trajectory and the first line positive trajectory, all of the lines of positive trajectory appear immediately after the lines of negative trajectory. All other lines must have zero trajectory since w corresponds to a 321-avoiding permutation.

Now suppose that there exists $I \leq J < K$ such that lines I through J of $LD(w)$ are the only lines of negative trajectory, lines $J+1$ through K are the only lines of positive trajectory and all other lines have zero trajectory. First, pick c between $J+1$ and K. In other words, line c has positive trajectory. There must be a circle in row J and column c since if line c does not cross line J from below, it cannot cross any other line, and thus it would have zero trajectory. Now assume that column c has a circle in row $r < J$. This implies that line r has positive trajectory. However, in order for line c to cross line r from below, it must also cross lines $r+1$ through J from below as well. In other words, we have shown that the circles in each column appear in consecutive rows, ending at row J.

And lastly, suppose that r_c is the minimum row index r such that there is a circle in row r and column c. If line $c+1$ has positive trajectory, then line $c+1$ cannot cross line $r_c - 1$, since line c does not cross line $r_c - 1$. Thus the number of circles in each column weakly decreases from left to right. This, combined with the previous obsservation, implies that the circles form a Young tableau and therefore w is a Young word. □

4 The Bumping Process

In this section we will give a formal description of our bijection on reduced words. For a given word $w = w_1 w_2 \cdots w_l$, we define

$$w^{(t)} = w_1 w_2 \cdots w_{t-1} w_{t+1} \cdots w_l$$

and

$$w\uparrow_t = \begin{cases} w_1 w_2 \cdots w_{t-1}(w_t - 1)w_{t+1} \cdots w_l & \text{if } w_t > 1 \\ (w_1+1)(w_2+1)\cdots(w_{t-1}+1)w_t(w_{t+1}+1)\cdots(w_l+1) & \text{if } w_t = 1 \end{cases}$$

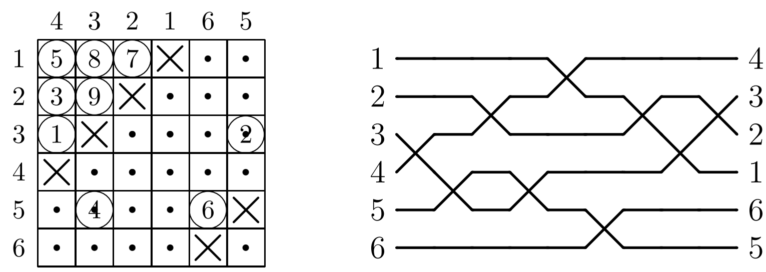
where we will refer to $w\uparrow_t$ as *the word obtained from w by bumping up at time t*. Note that w need not be a reduced word in the above definitions.

Let w be a word such that $w^{(t_0)}$ is reduced. Consider what happens when we bump w up at time t_0. Since $w^{(t_0)}$ is reduced, no two lines cross twice in $LD(w^{(t_0)})$. Therefore when we consider the line diagram of $v = w\uparrow_{t_0}$, the only two lines that could conceivably cross twice are the two lines that are crossed at time t_0. If they cross at some other time $t_1 \neq t_0$, then v is not reduced, however $v^{(t_1)}$ is reduced since it must correspond to the same permutation as $w^{(t_0)}$. We can now ask what happens if we bump v at time t_1. By the same reasoning, if it is not reduced, there must be some other time $t_2 \neq t_1$ such that $u^{(t_2)}$ is reduced, where $u = v\uparrow_{t_1}$.

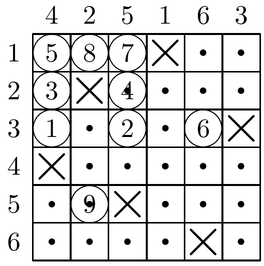
We may continue this process until the resulting word is reduced. We refer the reader to [4] for an explanation of why we will eventually reach a reduced word assuming that we began with one. This fact forms the basis for the following algorithm. Formally, we define the bumping algorithm starting at time t, $Bump_t$, as

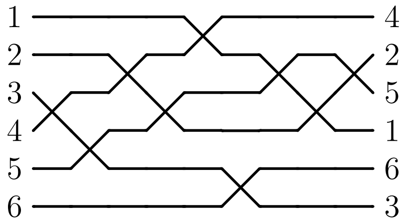
Input: $w \in Red(\sigma)$ such that $w^{(t)}$ is reduced
Algorithm body:
  $(i,j) :=$ row and column index of t in $LCD(w)$ (i.e. $t = (i,j)$-entry of $LCD(w)$)
  $v := w\uparrow_t$
  while v **is not reduced do**
    $t := (i,j)$-entry of $LCD(v)$
    $v := v\uparrow_t$
  od
Ouput: $Bump_t(w) := v$.

We should point out here that each time v is not reduced, the new value of t given by the (i,j)-entry of $LCD(v)$ can also be found using the line diagram. In general, the new value of t is the index of the other letter of v which switches the same two lines in $LD(v)$ that are switched by letter t. Of course, if v is reduced, this "other" letter does not exist, which is precisely when the algorithm terminates. If v is not reduced, this "other" letter will always appear in the (i,j)-entry of $LCD(v)$. An example should make our point clear. If we let $w = 352415232$ and $t = 2$ then (i,j) is initialized to $(5,2)$ and v is initialized to 342415232. The labeled circle and line diagrams for v are shown below.

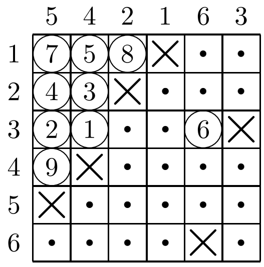


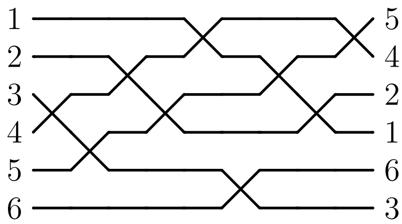
From the above diagrams we can see that t now equals 4 and v is reset to be 342315232. Notice that we can also see that $t = 4$ using the above line diagram. Using the previous values of $t = 2$ and $v = 342415232$, note that the 2nd letter of v switches lines 3 and 5 which are again switched by the 4th letter of v. Thus the new value of t is 4. The diagrams for the new value of v are shown below.





From the circle diagram, we see that the new value of t is 9 and thus the new value of v is 342315231. Again, we can see from the above line diagram that lines 2 and 5 are switched by the 4th and 9th letters of v. Thus the new value of t is 9. The diagram for the new value of v are shown below.





Notice that v is now reduced and thus the algorithm terminates. We end this section with the following property of our bijection.

Lemma 8 *Let w be a reduced word such that $w^{(t)}$ is also reduced. Then at each stage of the bumping algorithm, using the current values of v and t, $v^{(t)}$ corresponds to the same permutation as $w^{(t)}$.*

Proof. Suppose that $w^{(t)}$ corresponds to the permutation σ. Let $v = w{\uparrow}_t$. Clearly $v^{(t)}$ also corresponds to σ since this is the same word as $w^{(t)}$. But if v is not reduced, then there exists a value $s \neq t$ such that $v^{(s)}$ is reduced and in fact, the lines that cross at time s in $LD(v)$ are the same two lines that cross at time t. Therefore $v^{(s)}$ also corresponds to σ. And the result follows by induction. □

5 Row Insertion

The Robinson-Schensted correspondence is a map between permutations and pairs of standard tableaux of the same shape. The basic operation needed in performing this bijection is called row insertion. In this section, we will show how to translate this operation into that of the bumping algorithm described earlier.

Suppose that we are given a Young word $w = w_1 \cdots w_l$ corresponding to a permutation in S_n. For a given t between 1 and $l+1$ we will define the procedure $Insert_t$, which will convert the word

$$w_1 \cdots w_{t-1}(n+1)w_t \cdots w_l$$

into a Young word. The algorithm is as follows.

Input: Young word w corresponding to $\sigma \in S_n$
Algorithm Body:
$v := w_1 \cdots w_{t-1}(n+1)w_t \cdots w_{l(\sigma)}$
while v **is not a Young word do**
$L :=$ first line above cross in position t of $LD(v)$ with nonpositive trajectory
while v_t **remains strictly below line** L **in** $LD(v)$ **do**
$v := Bump_t(v)$
od
$t :=$ index of last letter bumped up in most recent application of $Bump_t$
od
Output: $Insert_t(w) := v$

The following comments justify the above algorithm as well as explain why the results are precisely the same as Robinson-Schensted row insertion. In the following, we assume that lines I through J of $LD(w)$ have negative trajectory, lines $J+1$ through K have positive trajectory and all other lines have zero trajectory just as in Lemma 7. Furthermore, to simplify our explanations, when we refer to the trajectory of a line, we are referring to the trajectory of that line in $LD(w)$, even though the trajectory of that line in $LD(v)$ may not be the same or even monotonic.

1. Using the initial value of v, we have that v is reduced since v contains precisely one more letter than w and the corresponding permutation has precisely one more inversion than σ. We also see that $v^{(t)}$ is reduced since $v^{(t)} = w$. This justifies applying $Bump_t$ to v for the first time. Throughout the **while** loop, if the value of t stays the same, then $v^{(t)}$ remains the same and is thus reduced. If t changes, then $v^{(t_{new})}$ still corresponds to the same permutation as $v^{(t_{old})}$ using Lemma 8. Thus at each step, applying $Bump_t$ to v is justified (i.e. $v^{(t)}$ is reduced).

2. If the cross at time t in $v{\uparrow}_t$ remains below line L in $LD(v{\uparrow}_t)$ then $v{\uparrow}_t$ is reduced. This is because the cross at time t in $v{\uparrow}_t$ interchanges two lines with positive trajectory. Therefore by Lemma 6, $v{\uparrow}_t$ is reduced. Thus $Bump_i(v)$ is equivalent to $v{\uparrow}_t$ in this case. As we will see below, this part of the algorithm is literally seeking out the smallest number in row L that is greater than t. The next comment highlights what happens when that number is found.

3. If the cross at time t in $LD(v{\uparrow}_t)$ is not below line L, then $Bump_t(v)$ involves the following process. First, let $v = v{\uparrow}_t$. Now the cross at time t in $LD(v)$ involves line L and some other line, say line M. Notice that line M must have positive trajectory by our choice of L and therefore lines L and M cannot cross prior to time t.

   (a) If t is greater than every entry in row L of $LCD(w)$ then v must be reduced since line L does not cross any line, M included, after time t. Thus we have added t at the end of the list of times at which line L crosses another line from above. In terms of $T(w)$, we have simply added a t to the end of the row corresponding to line L. Therefore, v is a Young word since $T(v)$ is a standard tableau of non-skew shape and thus the algorithm terminates.

   (b) If t is not greater than every entry in row L of $LCD(w)$ then v is not reduced. In fact, the smallest number greater than t in row L, call it s, marks the time at which lines L and M cross again. This is because line M is the first line below line L with positive trajectory. In other words, the next time line L crosses another line from above in $LD(w)$, it must cross line M. Notice that this implies that $v_s = v_t$. In other words, each time we bump up a letter, the value of that letter is one less then the value of the previous letter bumped. Now we simply set $t = s$, $v = v{\uparrow}_s$ and $L = L - 1$.

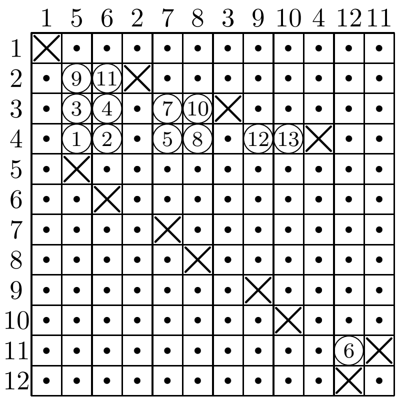
i. If v is reduced, then we determine whether or not v is a Young word. If it is, then we are done, otherwise we ask whether or not the cross at time t in $LD(v{\uparrow}_t)$ is below line L.

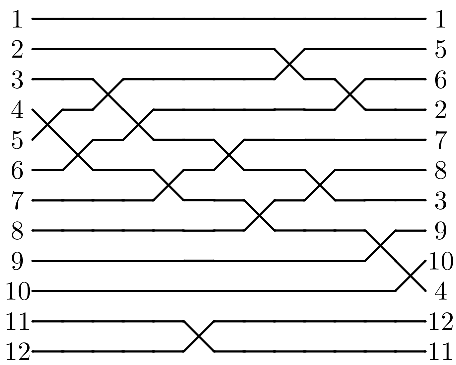
ii. If v is not reduced, then we ask whether or not t is greater than every entry in row L of $LCD(w)$.

In terms of $T(w)$, we have exchanged the s for a t and we start the comparison of s to the entries of the row corresponding to $L - 1$.

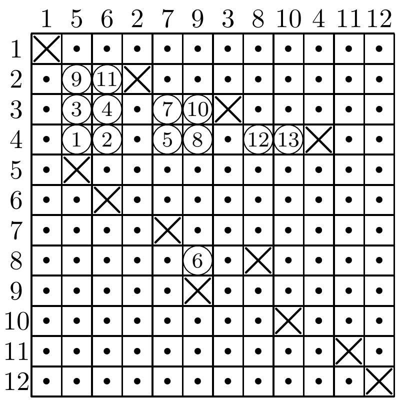
4. The process ends of course when v corresponds to a Young word. Since the algorithm always bumps up letters of strictly decreasing value, the worst case scenario is that we cross a line of zero trajectory with the first line of positive trajectory. But this simply corresponds to creating a new row on top of our tableau and inserting the value t into that new position. This insures that the algorithm terminates.

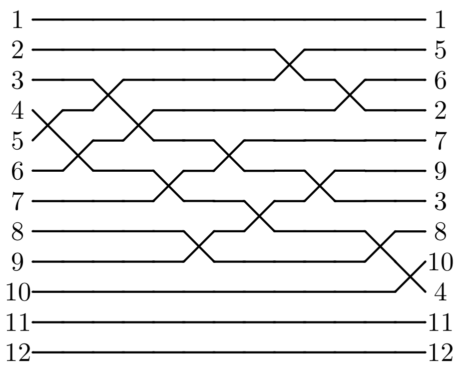
It should be clear from comments 3 and 4 that $Insert_t$ acts just like Robinson-Schensted row insertion. To make these points clear, let us consider the following example with $w = 453465726389$, $n = 10$ and $t = 6$.



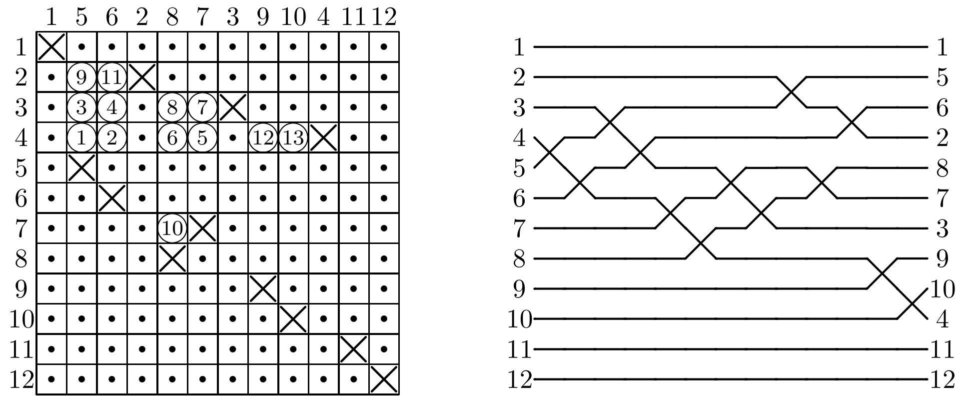


The first time through the **while** loop, we must apply $Bump_6(v)$ 4 times before the cross at time $t = 6$ is no longer below the line $L = 4$. Thus after the first 3 applications of $Bump_6(v)$, we arrive at the following word.

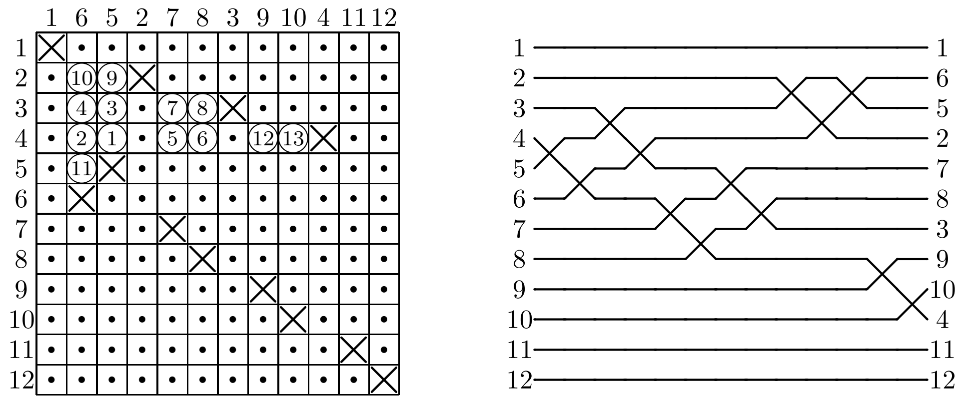




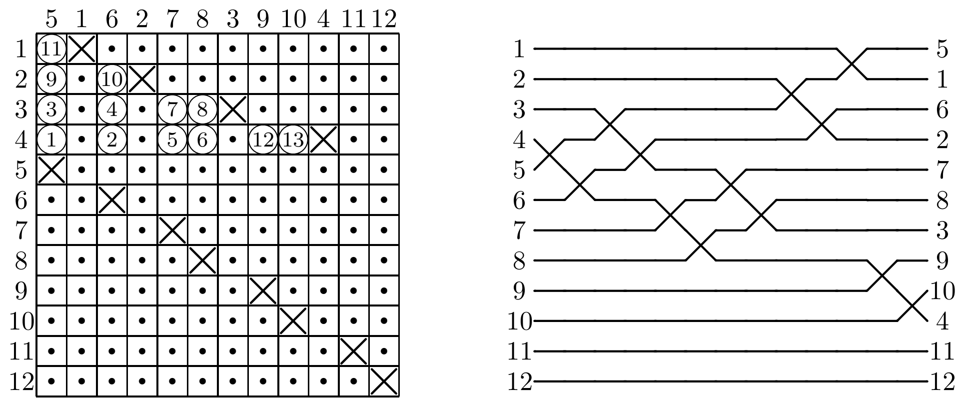
Notice that in $LCD(v)$, the 6 has positioned itself below the 8, the smallest number in row 4 that is greater than 6. Now the next time we apply $Bump_6$ to v, the cross at time $t = 6$ will no longer be below the line $L = 6$. This also forces us to bump up at time 8 and 10 as well, before reaching a reduced word. Thus the new values of t and L are 10 and 2 respectively. The current status of v is illustrated below.



We now apply $Bump_{10}$ to v. This results in $v{\uparrow}_{10}$ since there is a line of positive trajectory, namely line 6, between lines 2 and 7. Having done so, the cross at time $t = 10$ is now directly below line $L = 2$. Thus we need to apply $Bump_{10}$ to v one last time. This results in the following word. The new values of t and L are 11 and 1, respectively.



One last time through the loop results in the following Young word.



In general, Robinson-Schensted row insertion is designed for inserting an integer into an arbitrary tableau filled with distinct numbers. In the above definition of $Insert_t$, we simplified things by dealing only with the numbers 1 through $l+1$. However, we can easily make adjustments to deal with this more general case. To this end, let $\alpha \in S_n$ and $\beta \in S_m$. We define $\alpha \otimes \beta$ to be the permutation in S_{n+m} given by

$$\alpha \otimes \beta = (\alpha_1, \ldots, \alpha_n, n + \beta_1, \ldots, n + \beta_m).$$

Clearly we have

$$l(\alpha \otimes \beta) = l(\alpha) + l(\beta)$$

and in fact, any reduced word for $\alpha \otimes \beta$ can be formed by taking a reduced word w corresponding to α and shuffling it with the word

$$v_1 + n, v_2 + n, \ldots, v_k + n$$

where $v = v_1 v_2 \cdots v_k$ is a reduced word corresponding to β.

In particular, let A_m be the alternating permutation in S_{2m} defined as

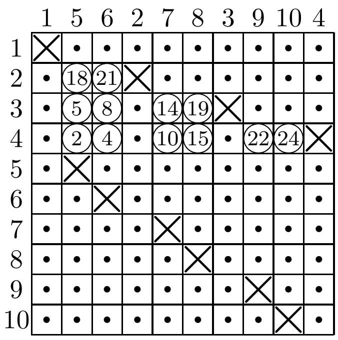
$$A_m = (2, 1, 4, 3, 6, 5, \ldots, 2m, 2m - 1).$$

One reduced word corresponding to A_m is $1, 3, 5, \ldots, 2m - 1$. In light of Lemma 2, each reduced word for A_m is simply a rearrangement of the first n odd integers and therefore there are exactly $m!$ such reduced words.

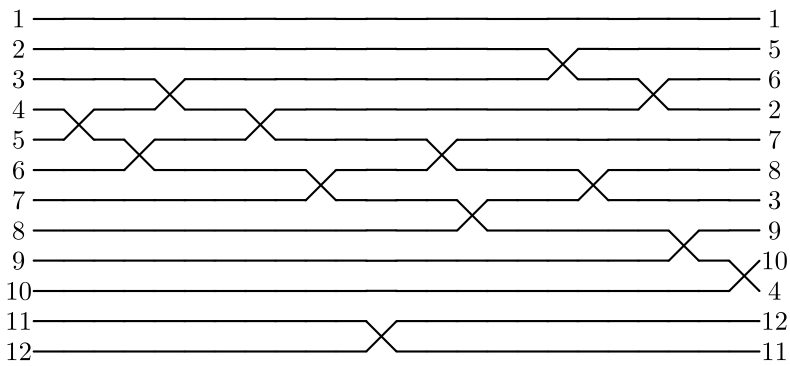
Now consider the permutation $\sigma = \alpha \otimes A_m$, where α is a Young permutation in S_n. A reduced word for σ consists of the word $w \in Red(\alpha)$ with the letters $n+1, n+3, \ldots, n+2m-1$ placed anywhere in between the letters of w. In this way, we can easily manipulate the entries of the tableau.

Suppose we are given a tableau T with distinct entries $t_1 < t_2 < \cdots < t_l$. First, construct the word $w := w_1 w_2 \cdots w_l$ corresponding to a permutation in S_n such that $T(w)$ is the same as T with t_i replaced by i. Now insert the numbers $n + 1, n + 3, \ldots, n + 2(t_l - l) - 1$ into w such that the t_ith letter is w_i. If we intend to apply $Insert_t$ to w, then we would make sure that $n + 1$ is in position t.

For example, suppose that we are given the following tableau (shown as a labeled circle diagram).



Starting with $w = 453465726389$, we then insert the numbers $11, 13, \ldots, 33$ such that the second letter is 4, the fourth letter is 5, and so on. We now have a word corresponding to $(1, 5, 6, 2, 7, 8, 3, 9, 10, 4) \otimes A_{12}$. In the event we wanted to apply $Insert_{12}$ to such a word, we would simply insist that we place 11 into position 12. The first 12 lines of this particular line diagram are illustrated below.



The insertion process would proceed just as before, except that we would ignore the letters $n+3$ through $n+2m-1$ when considering whether or not an intermediate value of v is a Young word. Of course after we have inserted one of these positions into w, the letters $n+3$ through $n+2m-1$ will remain in our word. This leads us to the following more general definition of $Insert_t$.

Input: $w \in Red(\sigma \otimes A_m)$ where $\sigma \in S_n$ is a Young permutation, $w_t = n+1$
Algorithm Body:
  $v := w$
  while σ **is not a Young permutation do**
    $L :=$ first line above cross in position t of $LD(v)$ with nonpositive trajectory
    while v_t **remains strictly below line** L **in** $LD(v)$ **do**
      $v := Bump_t(v)$
    od
    $t :=$ index of last letter bumped up in most recent application of $Bump_t$
    $\sigma :=$ permutation in S_{n+2} represented by first $n+2$ lines of $LD(v)$
  od
Output: $Insert_t(w) := v$

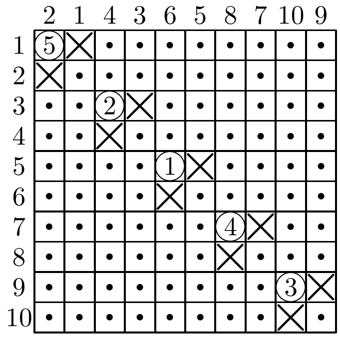
6 Robinson-Schensted Correspondence

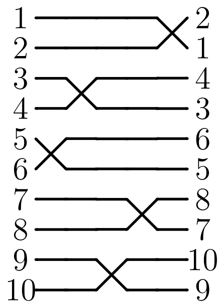
Now that we have described how the process of row insertion relates to the bumping algorithm, we must complete the story by showing how to convert a permutation into a pair of tableaux. We will first show how to translate a permutation into a reduced word.

Notice that the circle diagram of A_n consists of n circles placed in every other position along the diagonal, starting with the upper left most cell. Therefore, we can think of each permutation $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n) \in S_n$ as a reduced word for A_n by placing σ_i in the $(2i-1, 2i-1)$-entry. This corresponds to the reduced word $w(\sigma) := w_1 w_2 \cdots w_n$ where

$$w_{\sigma_i} = 2i - 1.$$

For example, the permutation $\sigma = (5, 2, 1, 4, 3)$ corresponds to the word $w = 53971$ as illustrated below.





That being done, the following algorithm translates the Robinson-Schensted correspondence into applications of the bumping algorithm.

Input: $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$
Algorithm Body:
  $u := w(\sigma)$
  $v := w(\sigma^{-1})$
  for $i = 2$ **to** n **do**
    $u := Insert_{\sigma_i}(u)$
    $v := Insert_{\sigma_i^{-1}}(v)$
  od
Output: $RS(\sigma) := (T(u), T(v))$

Notice that for each value of i, we begin the bumping process applied to u at time σ_i. By construction, this is the time at which lines $2i-1$ and $2i$ are crossed in $LD(u)$. In other words, in terms of the $LD(u)$, we are starting the bumping process with the ith cross from top to bottom.

We illustrate the above algorithm with the following example. Starting with $\sigma = (5, 2, 1, 4, 3)$, we have $u = 53971$ and $v = 93175$. The following table shows the intermediate values of u and v.

i	u	v
	5,3,9,7,1	9,3,1,7,5
2	6,2,10,8,1	10,2,1,8,6
3	3,2,11,9,1	10,2,1,8,3
4	3,2,11,4,1	10,2,1,3,2
5	3,2,4,3,1	3,2,1,4,3

Translating these last values of u and v into the tableaux $T(u)$ and $T(v)$, we obtain the following correspondence.

$$(5, 2, 1, 4, 3) \leftrightarrow \left(\begin{array}{|c|c|} \hline 5 & \\ \hline 2 & 4 \\ \hline 1 & 3 \\ \hline \end{array} \, , \, \begin{array}{|c|c|} \hline 3 & \\ \hline 2 & 5 \\ \hline 1 & 4 \\ \hline \end{array} \right)$$

Theorem 9 *Let $RS(\sigma) = (R, S)$ and let (P, Q) be the result of applying the Robinson-Schensted correspondence to σ. Then $P = R$ and $Q = S$.*

Proof. As shown in the previous section, $Insert_i$, is precisely the same operator as the Robinson-Schensted operation of row inserting i into a tableau. Therefore $P = R$. However, we cannot yet conclude that $Q = S$ since Q was not formed using row insertion. We resolve this problem in the following manner. Recall that the Robinson-Schensted correspondence has the property that if σ corresponds to (P, Q) then σ^{-1} corresponds to (Q, P). It is clear from the above algorithm that RS has this same property, namely $RS(\sigma^{-1}) = (S, R)$. In this light, Q is now formed by using row insertion and therefore $Q = S$. □

As pointed out in the above proof, given the definition of RS, it is clear that $RS(\sigma) = (R, S)$ if and only if $RS(\sigma^{-1}) = (S, R)$. However, without the preceding theorem, it is not clear that R and S should be of the same shape. In fact, the shapes of $T(u)$ and $T(v)$ are not the same for intermediate values of u and v. It would be of interest to find a simple combinatorial proof of this fact that does not rely on Robinson-Schensted.

7 Robinson-Schensted-Knuth Correspondence

The Robinson-Schensted-Knuth correspondence converts a generalized permutation into a pair of column strict tableaux of the same shape. In this section, we will show how to adapt the RS algorithm to account for this modification. To this end, a pair of sequences $\{u_i\}_{i=1}^n$ and $\{v_i\}_{i=1}^n$ that satisfy

$$\begin{aligned} &1)\ u_i \leq u_{i+1} \text{ and} \\ &2)\ \text{if } u_i = u_{i+1} \text{ then } v_i \leq v_{i+1} \end{aligned} \tag{1}$$

is called a *generalized permutation*. We will denote a generalized permutation π using the following two-line notation

$$\pi = \begin{pmatrix} u_1 & u_2 & u_3 & \cdots & u_n \\ v_1 & v_2 & v_3 & \cdots & v_n \end{pmatrix}. \tag{2}$$

We will also need to generalize the notion of a labeled circle diagram. Given a sequence of positive integers $v = (v_1, v_2, \ldots, v_n)$ and a reduced word $w = w_1 w_2 \cdots w_n$ corresponding to σ, we define a *generalized labeled circle diagram*, denoted $LCD_v(w)$, in the following manner. First, draw the circle diagram of σ. Next, construct the unique permutation $\alpha := \alpha(v) \in S_n$ such that

$$\begin{aligned} &1)\ v_{\alpha_1} \leq v_{\alpha_2} \leq \cdots \leq v_{\alpha_n} \text{ and} \\ &2)\ \text{if } v_i = v_j \text{ and } i < j \text{ then } \alpha_i^{-1} < \alpha_j^{-1}. \end{aligned}$$

And lastly, if line i crosses line j from above in $LD(w)$ at time t then label the circle in row i and column j of $CD(\sigma)$ with v_{α_t}. For example, let $v = (4, 5, 2, 6, 7, 5, 4, 2, 1)$ and $w = 352415232$. Therefore $\sigma = (4, 6, 2, 1, 5, 3)$ and $\alpha = (9, 3, 8, 1, 7, 2, 6, 4, 5)$. $LCD_v(w)$ is shown below.

	4	6	2	1	5	3
1	④	⑤	④	×	•	•
2	⑤	⑦	×	•	•	•
3	①	②	•	•	⑥	×
4	×	•	•	•	•	•
5	•	②	•	•	×	•
6	•	×	•	•	•	•

Notice that our definition of the labeled circle diagram $LCD(w)$ corresponds to $LCD_\sigma(w)$ since in this case we have $\alpha = \sigma^{-1}$. In the event that σ is 321-avoiding, then we may form the tableau of $LCD_v(w)$, denoted $T_v(w)$, by removing the cells of $LCD_v(w)$ that are void of circles.

We will now show how to translate the Robinson-Schensted-Knuth correspondence into our bumping algorithm. Given a generalized permutation π as in (2), we form the word $w(\pi) := w_1 w_2 \cdots w_n$ where

$$w_i = 2\alpha_i - 1 \tag{3}$$

and $\alpha = \alpha(v)$. Again, this definition is in complete agreement with how we defined $w(\sigma)$ for a permutation $\sigma \in S_n$. Clearly, $w(\pi)$ corresponds to the permutation A_n and $LCD_v(w(\pi))$ is formed by placing v_i in the ith circle on the diagonal of $CD(A_n)$.

For example, consider the permutation

$$\pi = \begin{pmatrix} 1 & 1 & 1 & 2 & 3 & 3 & 3 & 4 & 4 \\ 2 & 3 & 5 & 3 & 1 & 2 & 2 & 3 & 5 \end{pmatrix}.$$

Then $\alpha(v) = (5, 1, 6, 7, 2, 4, 8, 3, 9)$ and the corresponding word is given by

$$w(\pi) = 9, 1, 11, 13, 3, 7, 15, 5, 17.$$

We next define $\bar{\pi}$ to be the generalized permutation corresponding to the sequences

$$\bar{v} = (v_{\alpha_1}, v_{\alpha_2}, v_{\alpha_3}, \cdots, v_{\alpha_n}) \quad \text{and} \quad \bar{u} = (u_{\alpha_1}, u_{\alpha_2}, u_{\alpha_3}, \cdots, u_{\alpha_n}).$$

From our previous example, we have

$$\bar{\pi} = \begin{pmatrix} 1 & 2 & 2 & 2 & 3 & 3 & 3 & 5 & 5 \\ 3 & 1 & 3 & 3 & 1 & 2 & 4 & 1 & 4 \end{pmatrix},$$

$\alpha(\bar{u}) = (2, 5, 8, 6, 1, 3, 4, 7, 9) = \alpha(v)^{-1}$ and

$$w(\bar{\pi}) = 3, 9, 15, 11, 1, 5, 7, 13, 17.$$

We are now ready to prove our main result. First, we define the RSK algorithm as follows.

Input: $\pi = \begin{pmatrix} u_1 & u_2 & u_3 & \cdots & u_n \\ v_1 & v_2 & v_3 & \cdots & v_n \end{pmatrix}$

Algorithm Body:

$\alpha := \alpha(v)$

$w := w(\pi)$

$\bar{w} := w(\bar{\pi})$

for $i = 2$ **to** n **do**

$w := Insert_{\alpha_i^{-1}}(w)$

$\bar{w} := Insert_{\alpha_i}(\bar{w})$

od

Output: $RSK(\pi) := (T_v(w), T_{\bar{u}}(\bar{w}))$

Notice that the **for** loop of RSK is precisely the same as the **for** loop of RS applied to α^{-1}. In fact, the only difference between RS and RSK is the labeling that takes place at the end. However, it is clear from our definition of $\alpha(v)$ that for any pair of indices i and j,

$$v_i \leq v_j \text{ if } \alpha_i^{-1} < \alpha_j^{-1}$$

and

$$\alpha_i^{-1} < \alpha_j^{-1} \text{ if } v_i < v_j.$$

In other words, when we start bumping w at time α_i^{-1}, the bumping process is seeking out the smallest number in a given row that is strictly larger than α_i^{-1}. When we label the α_i^{-1} circle with v_i, this implies that the bumping process is equivalently finding the smallest number which is strictly larger than v_i. For example, consider the situation where

$$\alpha_j^{-1} < \alpha_i^{-1} < \alpha_k^{-1}$$

with $i > j, k$. In other words, in the process of inserting α_i^{-1} using the RS algorithm, we find that α_k^{-1} is the smallest number greater that α_i^{-1} in the current row. In terms of the RSK algorithm, we have

$$v_j \leq v_i \leq v_k.$$

However, since $k < i$, the second defining condition on $\alpha(v)$ implies that $v_i < v_k$. Therefore, the *Insert* algorithm is consistent with the generalized Robinson-Schensted-Knuth row insertion. Therefore Theorem 1 follows from the same method used to prove Theorem 9 with σ replaced by π and σ^{-1} replaced by $\bar{\pi}$.

8 An Alternate Approach

In [4], we described a different method for selecting the time at which to start the bumping process. This process was guided by Lascoux and Schützenberger's work on the Littlewood-Richardson rule [3]. In particular, we define the *last inversion* of a permutation σ to be the pair (σ_r, σ_s) where r is the last descent of σ, namely

$$r = max\{i \mid \sigma_i > \sigma_{i+1}\}$$

and

$$s = max\{i > r \mid \sigma_i < \sigma_r\}.$$

Given a word $w \in Red(\sigma)$, assume that lines σ_r and σ_s are crossed at time t in $LD(w)$. Notice that $w^{(t)}$ is reduced since (σ_r, σ_s) is the only inversion that is lost and not replaced by a new inversion in the permutation corresponding to $w^{(t)}$. In other words, defining t in this manner guarantees that we can apply $Bump_t$ to w. After each application of $Bump_t$, we get a new reduced word to which we can apply the bumping algorithm all over again. As a result of Lascoux and Schützenberger's work, if we continue in this manner, we will inevitably reach a word that corresponds to a permutation with precisely one descent. Such permutations are said to be *Grassmanian*. One can easily show that a permutation σ is Grassmanian if and only if σ^{-1} is a Young permutation.

Defined below, the LS algorithm converts any reduced word into a reduced word corresponding to a Grassmanian permutation.

Input: $w \in Red(\sigma)$
Algorithm body:
  $v := w$
  while σ **is not Grassmanian do**
    $r := \max\{i \mid \sigma_i > \sigma_{i+1}\}$
    $s := \max\{i > r \mid \sigma_i < \sigma_r\}$
    $t := (\sigma_s, r)$-entry of $LCD(v)$
    $v := Bump_t(v)$
    $\sigma :=$ permutation corresponding to v
  od
Ouput: $LS(w) := v$.

For example, consider applying LS to $w = 17935$, a reduced word corresponding to the permutation $(2, 1, 4, 3, 6, 5, 8, 7, 10, 9)$. In this case $r = 9$, $s = 10$ and $t = 3$. Applying $Bump_3$ to

w yields 17835. The following table lists the values of v and σ which occur as intermediate steps in computing $LS(17935)$ and $LS(57139)$.

v	σ
17935	(2,1,4,3,6,5,8,7,10,9)
17835	(2,1,4,3,6,5,8,9,7,10)
16735	(2,1,4,3,7,5,8,6,9,10)
15635	(2,1,4,3,7,6,5,8,9,10)
14635	(2,1,5,3,7,4,6,8,9,10)
14534	(2,1,5,6,3,4,7,8,9,10)
13423	(2,4,5,1,3,6,7,8,9,10)

v	σ
57139	(2,1,4,3,6,5,8,7,10,9)
57138	(2,1,4,3,6,5,8,9,7,10)
56137	(2,1,4,3,6,7,8,5,9,10)
45136	(2,1,5,3,6,7,4,8,9,10)
34135	(2,1,5,4,6,3,7,8,9,10)
24134	(3,1,5,4,2,6,7,8,9,10)
24123	(3,2,5,1,4,6,7,8,9,10)
23123	(3,4,2,1,5,6,7,8,9,10)
34123	(2,4,5,1,3,6,7,8,9,10)

Notice that the starting and ending values of v are precisely the reverse words of the example from the previous section. Formally, if $w = w_1 \cdots w_l$ define the reverse of w, denoted w^r to be

$$w^r = w_l \cdots w_2 w_1.$$

If w corresponds to σ then clearly w^r corresponds to σ^{-1} and is reduced if and only if w is reduced. Specifically, if we let $w = 53971$, in the previous section, we saw that RS maps this word to $u = 32431$. Using LS, we see that w^r is mapped to u^r. Similarly, if $w = 93175$, RS mapped w to $u = 32143$. Using LS, we again see that w^r is mapped to u^r. Based on substantial empirical evidence, we conjecture that this is true for all permutations σ.

Conjecture 10 *Let $RS(\sigma) = (R, S)$. Then $R = T(LS(u^r)^r)$ and $S = T(LS(v^r)^r)$ where $u = w(\sigma)$ and $v = w(\sigma^{-1})$.*

Empirical evidence has also suggested the following much stronger conjecture.

Conjecture 11 *Let w be a reduced word and let $t_1, t_2, \ldots t_m$ be any sequence such that*

$$v := Bump_{t_m}(\cdots(Bump_{t_2}(Bump_{t_1}(w)))\cdots)$$

corresponds to a Grassmanian permutation. Then $T(v) = T(LS(w))$.

That is to say that no matter what process you use to determine the starting time of the bumping algorithm, if you continue until you reach a Grassmanian permutation, then the resulting tableau will always be the same. This conjecture has been verified by computer for $n \leq 6$.

References

[1] Sara C. Billey, William Jockusch, and Richard P. Stanley. Some combinatorial properties of Schubert polynomials. *J. Algebraic Combin.*, 2(4):345–374, 1993.

[2] Donald E. Knuth. Permutations, matrices, and generalized Young tableaux. *Pacific J. Math.*, 34:709–727, 1970.

[3] Alain Lascoux and Marcel-Paul Schützenberger. Schubert polynomials and the Littlewood-Richardson rule. *Lett. Math. Phys.*, 10(2-3):111–124, 1985.

[4] David P. Little. Combinatorial aspects of the Lascoux-Schützenberger tree. *Adv. Math.*, 174(2):236–253, 2003.

[5] C. Schensted. Longest increasing and decreasing subsequences. *Canad. J. Math.*, 13:179–191, 1961.

[6] Richard P. Stanley. On the number of reduced decompositions of elements of Coxeter groups. *European J. Combin.*, 5(4):359–372, 1984.