

A SCHENSTED ALGORITHM FOR SHIFTED RIM HOOK TABLEAUX

JAEJIN LEE

0. Introduction

In [S] Schensted constructed the Schensted algorithm giving a bijection between permutations and pairs of Young standard tableaux (see also [Kn1]). After Knuth generalized it to column strict tableaux in [Kn2], various analogs of the Schensted algorithm came: versions for rim hook tableaux ([W], [SW]), shifted tableaux ([Sa], [Wo]), oscillating tableaux [B], and skew tableaux [SS].

In this paper we give the Schensted algorithm for shifted rim hook tableaux. If k is a fixed odd positive integer it shows a one-to-one correspondence between all pairs (P, Q) , where P is a shifted (first tail circled) k -rim hook tableau of shape λ and content k^m and Q is a circled shifted k -rim hook tableau of the same shape λ and content k^m , and all circled hook permutations of content k^m . In particular, if all the rim hooks of P were of size one and $|\sigma| = 1$ then this algorithm reduces to the Schensted algorithm for ordinary shifted tableaux given by Sagan [Sa].

In Section 1 we provide the definitions and notation used in this paper. Section 2 describes the “bumping” algorithm which is the basic building block of the subsequent algorithms. It is an analog to Schensted “bumping.” In Section 3 the “insertion” and “deletion” algorithms are given. In Section 4 we give the “encode” and “decode” algorithms and state the theorems which follow from these algorithms.

1. Preliminaries

We use standard notation \mathbf{P} , \mathbf{Z} for the set of all positive integers and the ring of integers, respectively.

Received January 14, 1993.

Partially supported by Korea Research Foundation, 1991.

DEFINITION 1.1. A *partition* λ of a nonnegative integer n is a sequence of nonnegative integers $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_\ell)$ such that

- (1) $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_\ell > 0$,
- (2) $\sum_{i=1}^{\ell} \lambda_i = n$.

We write $\lambda \vdash n$, or $|\lambda| = n$. We say each term λ_i is a *part* of λ and n is the *weight* of λ . The number of nonzero parts is called the *length* of λ and is written $\ell = \ell(\lambda)$. Let \mathcal{P} be the set of all partitions and \mathcal{P}_n be the set of all partitions of n .

We sometimes abbreviate the partition λ with $1^{j_1} 2^{j_2} 3^{j_3} \dots$, where j_i is the number of parts of size i . Sizes which do not appear are omitted and if $j_i = 1$, then it is not written. Thus, a partition $(5, 3, 2, 2, 2, 1) \vdash 15$ can be written $12^3 35$.

DEFINITION 1.2. Let $\lambda = (\lambda_1, \dots, \lambda_\ell)$ be a partition. The *Ferrers diagram (shape)* D_λ of λ is the array of cells or boxes arranged in rows and columns, λ_1 in the first row, λ_2 in the second row, etc., with each row left-justified. That is,

$$D_\lambda = \{(i, j) \in \mathbf{Z}^2 \mid 1 \leq i \leq \ell(\lambda), 1 \leq j \leq \lambda_i\},$$

where we regard the elements of D_λ as a collection of boxes in the plane with matrix-style coordinates. Sometimes we identify a partition with its diagram, so that $x \in \lambda$ should be interpreted as $x \in D_\lambda$. See Figure 1.1 for a Ferrers diagram of $(6, 5, 5, 3, 2) \vdash 21$. In particular, if $\lambda = 1^i j$ for some nonnegative integer i, j then D_λ is called a *hook*.

NOTATION. We denote

$$DP = \{\mu \in \mathcal{P} \mid \mu \text{ has all distinct parts}\}.$$

DEFINITION 1.3. For each $\lambda \in DP$, a *shifted diagram* D'_λ of shape λ is defined by

$$D'_\lambda = \{(i, j) \in \mathbf{Z}^2 \mid i \leq j \leq \lambda_j + i - 1, 1 \leq i \leq \ell(\lambda)\}.$$

And for $\lambda, \mu \in DP$ with $D'_\mu \subseteq D'_\lambda$, a *shifted skew diagram* $D'_{\lambda/\mu}$ is defined as the set-theoretic difference $D'_\lambda \setminus D'_\mu$. Figure 1.2 shows D'_λ and $D'_{\lambda/\mu}$ respectively when $\lambda = (9, 7, 4, 2)$ and $\mu = (5, 3, 1)$.

DEFINITION 1.4. A shifted skew diagram θ is called a *single rim hook* if θ is connected and contains no 2×2 block of cells. If θ is a single rim hook, then its *head* is the upper rightmost cell in θ and its *tail* is the lower leftmost cell in θ . See Figure 1.3.

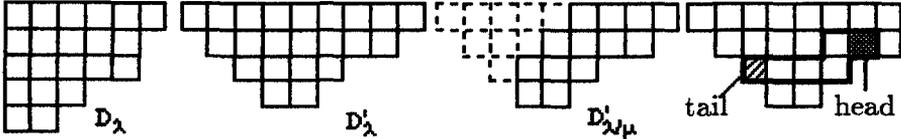


Figure 1.1

Figure 1.2

Figure 1.3

DEFINITION 1.5. A *double rim hook* is a shifted skew diagram θ formed by the union of two single rim hooks both of whose tails are on the main diagonal. If θ is a double rim hook, we denote by $\mathcal{A}[\theta]$ (resp., $\alpha_1[\theta]$) the set of diagonals of length two (resp., one). Also let $\beta_1[\theta]$ (resp., $\gamma_1[\theta]$) be a single rim hook in θ which starts on the upper (resp., lower) of the two main diagonal cells and ends at the head of $\alpha_1[\theta]$. The tail of $\beta_1[\theta]$ (resp., $\gamma_1[\theta]$) is called the *first tail* (resp., *second tail*) of θ and the head of $\beta_1[\theta]$ or $\gamma_1[\theta]$ (resp., $\gamma_2[\theta], \beta_2[\theta]$, where $\beta_2[\theta] = \theta \setminus \beta_1[\theta]$ and $\gamma_2[\theta] = \theta \setminus \gamma_1[\theta]$) is called the *1st head* (resp., *second head, third head*) of θ . Hence we have the following descriptions for a double rim hook θ :

$$\begin{aligned} \theta &= \mathcal{A}[\theta] \cup \alpha_1[\theta] \\ &= \beta_1[\theta] \cup \beta_2[\theta] \\ &= \gamma_1[\theta] \cup \gamma_2[\theta]. \end{aligned}$$

Definition 1.5 is illustrated in Figure 1.4. We write \mathcal{A}, α_1 , etc. for $\mathcal{A}[\theta], \alpha_1[\theta]$, etc. when there is no confusion.

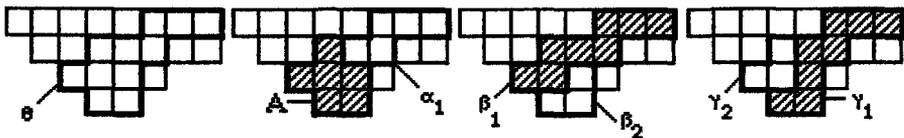


Figure 1.4

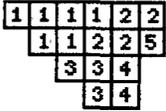


Figure 1.5

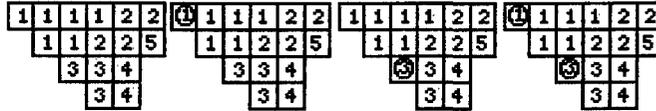


Figure 1.6

We will use the term *rim hook* to mean a single rim hook or a double rim hook.

DEFINITION 1.6. A *shifted rim hook tableau* of shape $\lambda \in DP$ and content $\rho = (\rho_1, \dots, \rho_m)$ is defined recursively. If $m = 1$, a rim hook with all 1's and shape λ is a shifted rim hook tableau. Suppose P of shape λ has content $\rho = (\rho_1, \rho_2, \dots, \rho_m)$ and the cells containing the m 's form a rim hook inside λ . If the removal of the m 's leaves a shifted rim hook tableau, then P is a shifted rim hook tableau. We define a *shifted skew rim hook tableau* in a similar way.

Let P be a shifted rim hook tableau. We write $\kappa_P\langle r \rangle$ (or just $\kappa\langle r \rangle$) for a rim hook of P containing r . Figure 1.5 shows an example of a shifted rim hook tableau P of shape $(7, 5, 3, 2)$ and content $(6, 5, 3, 2, 1)$.

DEFINITION 1.7. Suppose P is a shifted rim hook tableau. Then we denote by P^1 one of the tableaux obtained from P by circling or not circling the first tail of each double rim hook in P . The P^1 is called a *first tail circled rim hook tableau*. We use the notation $|\cdot|$ to refer to the uncircled version; e.g., $|P^1| = P$. Figure 1.6 shows all first tail circled rim hook tableaux obtained from P , where P is a shifted rim hook tableau in Figure 1.5.

From now on, unless we explicitly specify to the contrary, all rim hook tableaux will be first tail circled rim hook tableaux.

DEFINITION 1.8. Suppose λ is a shifted shape and α a shifted skew shape, where $\alpha = \mu/\nu$. The *outer rim* of λ is the set of cells in λ with no cell in λ immediately below or no cell in λ immediately to the right. A rim hook τ in λ is called an *outer rim hook* of λ if

- (1) removal of τ from λ leaves a shifted Young diagram and
- (2) τ has non-empty intersection with the outer rim of λ .

Similarly we define the *inner rim* of α and an *inner rim hook* of α . The *outside rim* of λ is the set of cells immediately to the right or

below cells in λ , or in the first row and to the right of λ . The *inside rim* of α is the outer rim of ν , plus a row of cells along the top border of ν starting at the boundary with ν . A rim hook τ is an *outside rim hook* of λ if

- (1) $\lambda \cup \tau$ is a shifted Young diagram and $\lambda \cap \tau = \emptyset$,
- (2) τ has non-empty intersection with the outside rim of λ .

An *inside rim hook* of α is defined in a similar way. These definitions are illustrated in Figure 1.7.

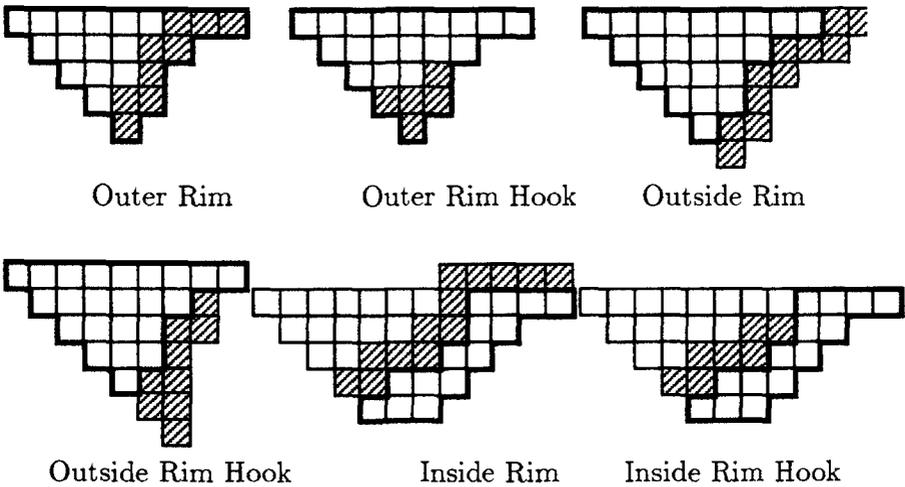


Figure 1.7

It is necessary in discussions involving tableaux and shapes to refer to the directions within the shape. In what follows, compass directions will be used to refer to the relative positions within shape. Generally speaking, x will be SE of y if the row of x is the same as or below the row of y and the column of x is the same as or to the right of the column of y . Also, x will be *strictly* SE of y if x is SE of y but not in the same row or same column.

DEFINITION 1.9. Let λ be a shifted shape and x be a fixed cell on the outside rim of λ . Every cell y along the outside rim of λ can be described by its distance from x in one of two ways: either directly or

reflected off the main diagonal. If $x = (i_x, j_x)$ and $y = (i_y, j_y)$, write $d_x = j_x - i_x$ and $d_y = j_y - i_y$. If $d_x \leq d_y$ (y is NE of x), we say y is $d_y - d_x$ above x on λ , or we say y is a reflected cell $d_y + d_x + 1$ below x on λ . If $d_x > d_y$ (y is SW of x), we say y is $d_x - d_y$ below x on λ , or we say y is a reflected cell $d_y + d_x + 1$ below x on λ . Thus, we may refer to the cell numbered j above x on λ , or the cell numbered k below x on λ .

Any outside rim hook of λ can be described by a cell x , a length k , and a direction d (SW or NE). That is, if $d = \text{NE}$, then the rim hook consists of the cells in the outside rim of λ above x and numbered $0, 1, \dots, k - 1$.

On the other hand, suppose now $d = \text{SW}$. If $k - 1 \leq d_x$, then the rim hook consists of the cells in the outside rim of λ below x and numbered $0, 1, \dots, k - 1$. If $d_x < k - 1 \leq 2d_x + 1$, the rim hook consists of the cells in the outside rim of λ below x and numbered $0, 1, \dots, d_x$, plus the cells diagonally SE of the cells below x and numbered $d_x + 1, \dots, k - 1$. If $k - 1 > 2d_x + 1$, then the rim hook consists of the cells in the outside rim of λ below x and numbered $0, 1, \dots, d_x, 2d_x + 2, 2d_x + 3, \dots, k - 1$, plus the cells diagonally SE of the cells below x and numbered $d_x + 1, \dots, 2d_x + 1$.

We shall say an outside rim hook τ of λ admits (x, k, d) and (x, k, d) describes τ .

REMARK 1.10.

(1) Any outside rim hook of λ has two such descriptions. Single rim hooks may be described by (x, k, NE) , where x is the tail, and also by (y, k, SW) , where y is the head. Double rim hooks may be described by (x, k, SW) , where x is the first head, and also by (y, k, SW) , where y is the second head.

(2) More generally, (λ, x, k, d) describes a set of cells which intersects the outside rim of λ as detailed in the preceding paragraphs. This set of cells need not be a legal outside rim hook of λ .

DEFINITION 1.11. Suppose τ is an outside rim hook of λ and τ admits (x, n, d) . A j -repeated-slide of τ along λ from x in the direction d is the set of cells described by (λ', y, n, d) , where $\lambda' = \lambda \cup \tau$ and y and d' are constructed as follows. If $d = \text{NE}$, the cell y is the cell numbered jn above x . If $d = \text{SW}$, let x' be the cell numbered jn below x . Then

y is the cell diagonally SE of the cell x' if $x' \in \tau$. Otherwise, $y = x'$. If x' is a reflected cell, then $d' = NE$. Otherwise, $d' = d$. Let us also write $RepeatedSlide_{j,\lambda}(x, d)$ to denote the pair (y, d') which emerges from a j -repeated-slide. See Figure 1.8. Here x, y are the head and tail of τ , respectively.

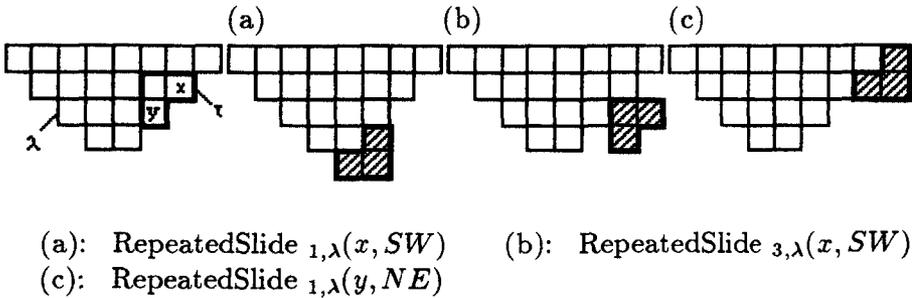


Figure 1.8

Informally, a j -repeated-slide of τ from x moves the cells of τ along the outside rim of $\lambda \cup \tau$ in the direction d by $j|\tau|$ steps. If α is a shifted skew shape, $RepeatedSlide_{j,\alpha}(x, d)$ are defined in similar ways.

As we mentioned earlier in Remark 1.10, (λ, x, k, d) may not be a legal outside rim hook of λ' . Algorithm Bump will fix such a set so that the resulting new set is an outside rim hook of a certain shape $\hat{\lambda}$. Let τ be a rim hook of shifted skew shape and $x = (i, j) \in \tau$.

Algorithm Bump (Input: $\tau, x, direction$; Output: \hat{x})

begin

if direction is outward **then**
 if τ is a single rim hook **then**
 $\hat{x} \leftarrow (i + 1, j + 1)$
 else (* τ is a double rim hook *)
 if $x = (i, j) \in \alpha_1[\tau]$ **then**
 $\hat{x} \leftarrow (i + 1, j + 1)$
 else (* $x = (i, j) \in \mathcal{A}[\tau]$ *)
 $\hat{x} \leftarrow (i + 2, j + 2)$
 else (* direction is inward *)
 if τ is a single rim hook **then**

```

 $\hat{x} \leftarrow (i - 1, j - 1)$ 
else (*  $\tau$  is a double rim hook *)
  if  $x = (i, j) \in \alpha_1[\tau]$  then
     $\hat{x} \leftarrow (i - 1, j - 1)$ 
  else (*  $x = (i, j) \in \mathcal{A}[\tau]$  *)
     $\hat{x} \leftarrow (i - 2, j - 2)$ 

```

end .

DEFINITION 1.12. Let us write $\text{Bump}(\tau, x, \text{direction})$ to mean the resulting cell \hat{x} in $\text{Bump}(\tau, x, \text{direction}; \hat{x})$. If τ is a rim hook of shifted skew shape and $\psi \subseteq \tau$, then we write

$$\text{Bump}_\tau(\psi, \text{direction}) = \{\text{Bump}(\tau, x, \text{direction}) \mid x \in \psi\}.$$

In Figure 1.9 (a), the set of shaded cells shows $\text{Bump}_\tau(\psi, \text{out})$, where ψ is the set of cells marked with x's. $\text{Bump}_\sigma(\phi, \text{in})$ is given in Figure 1.9 (b). Here ϕ is the set of cells containing x's.

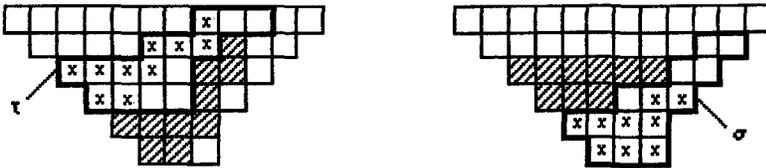


Figure 1.9

We now describe Algorithm MakeRimHook which tells us how to make a rim hook from a given hook of odd size. The input to this algorithm is a hook $\sigma = 1^i j$ ($i + j$ odd, $j \neq 1$) and the output is an outside rim hook of \emptyset .

Algorithm MakeRimHook (Input: σ ; Output: $\hat{\sigma}$)

begin

```

if  $j = 0$  then
  if  $i = 1$  then
     $\hat{\sigma} \leftarrow$  rim hook of shape  $(1, 0)$ 
  else (*  $i \neq 1$  *)

```

```

       $\hat{\sigma} \leftarrow$  double rim hook of shape  $(i-1, 1)$  with a circle on
      its first tail
    else (*  $j \neq 0$  *)
      if  $j > i$  then
         $\hat{\sigma} \leftarrow$  rim hook of shape  $(j, i)$ 
      else (*  $i > j$  *)
         $\hat{\sigma} \leftarrow$  rim hook of shape  $(i, j)$  with the first tail circled
  end .

```

We construct Algorithm MakeHook to reverse the MakeRimHook algorithm. Let τ be an outside rim hook of \emptyset and (i, j) ($i > j$) be the shape of τ .

Algorithm MakeHook (Input: τ ; Output: $\hat{\tau}$)

```

begin
  if  $\tau$  has no circle on its first tail then
     $\hat{\tau} \leftarrow 1^j i$ 
  else (*  $\tau$  has a circle on its first tail *)
    if  $j = 1$  then
       $\hat{\tau} \leftarrow 1^{i+j}$ 
    else (*  $j \neq 1$  *)
       $\hat{\tau} \leftarrow 1^i j$ 
end .

```

2. Bumping algorithms

From now on, we assume that k is a fixed odd number. Now we describe a procedure called the “bumping algorithm” which is the basic building block of the subsequent algorithms. The bumping algorithm shows us how an area within a shape, called the *bumping hook*, changes two tableaux, one a shifted skew rim hook tableau and the other a shifted rim hook tableau. The result of this procedure is a new bumping hook and two new tableaux.

We describe two such algorithms, *BumpOut* and *BumpIn*. The movement of BumpOut is outward while the movement of BumpIn is inward. However, we will analyze only BumpOut in detail because these two algorithms are “mirror images” of one another.

The input to this algorithm is a pair of tableaux (T, S) , whose overlap is called the bumping hook, satisfying Conditions 1-3 below and a mark (either true or false) of the bumping hook satisfying Condition 4. We say the bumping hook is *marked* if the mark is true, and *unmarked* if the mark is false. The result is a mark and a new pair of tableaux (\hat{T}, \hat{S}) which also satisfy these four conditions:

Condition 1. T is a shifted (first tail circled) rim hook tableau of shape λ with entries $\leq j - 1$; all parts of $\text{content}(T)$ are k .

Condition 2. S is a shifted skew (first tail circled) rim hook tableau of shape α with entries $\geq j$; all parts of $\text{content}(S)$ are k . We assume j occurs in S .

Condition 3. $\sigma = \lambda \cap \alpha$ is an outer rim hook of λ or, equivalently, an inner rim hook of α and $|\sigma|$ is k . We call σ the *bumping hook*.

Condition 4. σ is unmarked if $\sigma \cap \mathcal{M} \neq \emptyset$, where $\mathcal{M} = \{(i, i) \mid i \in \mathbf{P}\}$.

It is convenient, especially for the termination, to assume from now on that any shifted (skew) rim hook tableau has ∞ in every cell in its outside rim and 0 in every cell in its top borders.

Associated with the pair (T, S) is the bumping hook σ , a value j which appears in S and is smallest in S . Let $\tau = \kappa\langle j \rangle$. Since $|\sigma| = |\tau| = k$, we have the following three basic cases:

- Case (1) σ and τ are disjoint ($\sigma \cap \tau = \emptyset$).
- Case (2) σ and τ overlap ($\sigma \cap \tau \neq \emptyset$ and $\sigma \neq \tau$).
- Case (3) σ and τ coincide ($\sigma = \tau$).

Algorithm BumpOut (Input: T, S, mark ; Output: $\hat{T}, \hat{S}, \text{mark}$)

begin

```

if  $\sigma \cap \tau = \emptyset$  then
     $\hat{S} \leftarrow S - \tau(j)$ 
     $\hat{T} \leftarrow T \cup \tau(j)$ 
     $\text{mark} \leftarrow \text{mark of } \sigma$ 
else if  $\sigma \cap \tau \neq \emptyset$  and  $\sigma \neq \tau$  then
     $\tau' \leftarrow \text{Bump}_\tau(\tau \cap \sigma, \text{out}) \cup (\tau - (\tau \cap \sigma))$ 
     $\hat{S} \leftarrow S - \tau(j)$ 
     $\hat{T} \leftarrow T \cup \tau'(j)$ 
     $\text{mark} \leftarrow \text{mark of } \sigma$ 
else  $(\ast \sigma = \tau \ast)$ 
    
```

```

if  $\sigma$  is marked then
   $x \leftarrow$  tail of  $\tau$ 
   $d \leftarrow$  NE
else (*  $\sigma$  is unmarked *)
  if  $\tau$  is a single rim hook then
     $x \leftarrow$  head of  $\tau$ 
     $d \leftarrow$  SW
  else if  $\tau$  has first tail circled then
     $x \leftarrow$  1st head of  $\tau$ 
     $d \leftarrow$  SW
  else (*  $\tau$  has first tail uncircled *)
     $x \leftarrow$  2nd head of  $\tau$ 
     $d \leftarrow$  SW
   $j \leftarrow 1$ 
  repeat
     $\tau' = (y, |\tau|, d')$  with  $(y, d') \leftarrow \text{RepeatedSlide}_{j, (\lambda - \sigma)}(x, d)$ 
  until  $\tau'$  is legal on  $\lambda$ 
  if  $d' =$  NE then
    mark  $\leftarrow$  true
  else if  $d' =$  SW and  $|\tau' \cap \mathcal{M}| \leq 1$  then
    mark  $\leftarrow$  false
  else (*  $d' =$  SW and  $|\tau' \cap \mathcal{M}| = 2$  *)
    if the number of non-reflected cells of  $\tau'$  is greater than the
      number of reflected cells of  $\tau'$  then
       $\tau' \leftarrow$   $\tau'$  with no circle on the 1st tail of  $\tau'$ 
      mark  $\leftarrow$  false
    else
       $\tau' \leftarrow$   $\tau'$  with a circle on the 1st tail of  $\tau'$ 
      mark  $\leftarrow$  false
   $\hat{S} \leftarrow S - \tau(j)$ 
   $\hat{T} \leftarrow T \cup \tau'(j)$ 
end.

```

See Figures 2.1–2.3. In each figure, the boundaries of λ and α will be indicated in heavy line, so that it clearly shows σ enclosed in heavy outline. T, S and a mark of the bumping hook are given in the left

figures. Right figures show the resulting \hat{T}, \hat{S} and the new bumping hook $\hat{\sigma}$ with a mark from BumpOut. Verification that in each case

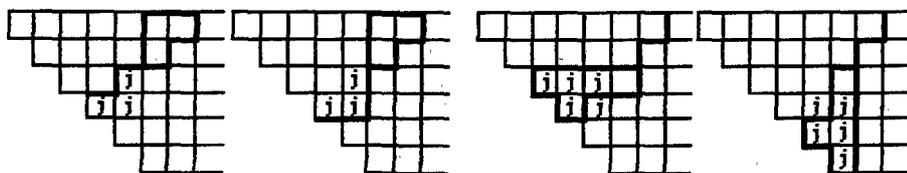


Figure 2.1

Figure 2.2

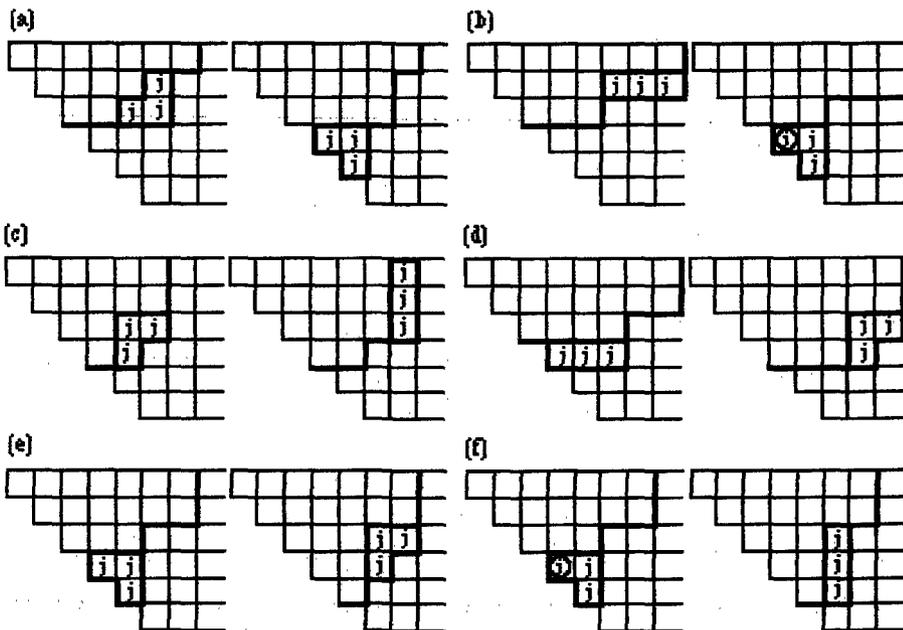


Figure 2.3

Conditions 1–4 are maintained is easily accomplished by careful analysis of the various cases. Since the basic idea for this verification is similar to White’s in [W], details are omitted.

We now describe the BumpIn algorithm. This algorithm can be obtained from BumpOut by reversing the construction in BumpOut. But Algorithm BumpIn differs significantly from BumpOut in Case

(3). This case provides for the only circumstances under which a hook can be bumped out of the tableau. This occurs when τ' cannot be constructed because *RepeatedSlide* encounters cells above the first row. See Figure 2.4. This special case will stop the Delete algorithm in Section 3 and a hook of j 's will be removed from T .

BumpIn has two additional outputs: *timetostop*, which indicates when the special circumstances described above happen, and j , the value in τ at the time of this occurrence.

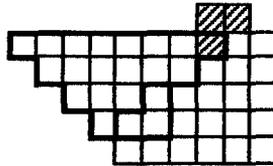


Figure 2.4

Algorithm BumpIn (Input: T, S, mark ; Output: $\hat{T}, \hat{S}, j, \text{timetostop}, \text{mark}$)

begin

if $\sigma \cap \tau = \emptyset$ **then**

$\hat{S} \leftarrow S \cup \tau(j)$

$\hat{T} \leftarrow T - \tau(j)$

$\text{mark} \leftarrow \text{mark of } \sigma$

else if $\sigma \cap \tau \neq \emptyset$ and $\sigma \neq \tau$ **then**

$\tau' \leftarrow \text{Bump}_\tau(\tau \cap \sigma, \text{in}) \cup (\tau - (\tau \cap \sigma))$

$\hat{S} \leftarrow S \cup \tau'(j)$

$\hat{T} \leftarrow T - \tau(j)$

$\text{mark} \leftarrow \text{mark of } \sigma$

else (* $\sigma = \tau$ *)

if σ is marked **then**

$x \leftarrow \text{head of } \tau$

$d \leftarrow \text{SW}$

else (* σ is unmarked *)

if τ is a single rim hook **then**

$x \leftarrow \text{tail of } \tau$

$d \leftarrow \text{NE}$

else if τ has first tail circled **then**

```

     $x \leftarrow$  1st head of  $\tau$ 
     $d \leftarrow$  SW
    else (*  $\tau$  has first tail uncircled *)
         $x \leftarrow$  3rd head of  $\tau$ 
         $d \leftarrow$  SW
     $j \leftarrow 1$ 
    repeat
         $\tau' = (y, |\tau|, d')$  with  $(y, d') \leftarrow \text{RepeatedSlide}_{j, (\alpha - \sigma)}(x, d)$ 
    until  $\tau'$  is legal on  $\alpha$  or  $\tau'$  encounters cells above 1st row
    if  $\tau'$  encounters cells above 1st row then
        timetostop  $\leftarrow$  true
    else (*  $\tau'$  is legal on  $\alpha$  *)
        if  $d' = \text{NE}$  then
            mark  $\leftarrow$  false
        else if  $d' = \text{SW}$  and  $\tau' \cap \mathcal{M} = \emptyset$  then
            mark  $\leftarrow$  true
        else if  $d' = \text{SW}$  and  $|\tau' \cap \mathcal{M}| = 1$  then
            mark  $\leftarrow$  false
        else (*  $d' = \text{SW}$  and  $|\tau' \cap \mathcal{M}| = 2$  *)
            if the number of non-reflected cells of  $\tau'$  is greater than
                the number of reflected cells of  $\tau'$  then
                 $\tau' \leftarrow \tau'$  with no circle on the 1st tail of  $\tau'$ 
                mark  $\leftarrow$  false
            else
                 $\tau' \leftarrow \tau'$  with a circle on the 1st tail of  $\tau'$ 
                mark  $\leftarrow$  false
         $\hat{S} \leftarrow S \cup \tau'(j)$ 
         $\hat{T} \leftarrow T - \tau(j)$ 
    end.

```

Since every construction in *BumpOut* is inverted in *BumpIn*, we have the following crucial lemma:

LEMMA 2.2. *BumpOut* and *BumpIn* are inverse algorithms. That is, the procedure:

begin

BumpOut ($T, S, \text{mark}; \hat{T}, \hat{S}, \text{mark}$)

BumpIn ($\hat{T}, \hat{S}, \text{mark}; \hat{T}, \hat{S}, j, \text{timetostop}, \text{mark}$)

end .
 yields $T = \hat{T}, S = \hat{S}$ and $\text{timetostop} = \text{false}$ and mark of σ_{TS} is equal to the mark of $\sigma_{\hat{T}\hat{S}}$; and the procedure:
begin
 BumpIn ($T, S, \text{mark}; \hat{T}, \hat{S}, j, \text{timetostop}, \text{mark}$)
 if not timetostop **then**
 BumpOut ($\hat{T}, \hat{S}, \text{mark}; \hat{T}, \hat{S}, \text{mark}$)
end .
 also yields $T = \hat{T}, S = \hat{S}$ and mark of $\sigma_{TS} = \text{mark of } \sigma_{\hat{T}\hat{S}}$.

3. Schensted insertion and deletion algorithms.

Using the Bumping algorithms in Section 2 we now describe insertion and deletion algorithms which are shifted rim hook analogs of the ordinary Schensted insertion and deletion algorithms for identifying permutations with pairs of standard tableaux.

Algorithm Insert has as input a shifted (first tail circled) rim hook tableau with all content parts k or 0, and a hook tableau of size k . The hook tableau must first be positioned so that we can apply Algorithm BumpOut in Section 2. Suppose λ is a shifted shape and τ is a hook of size k .

Algorithm Position (Input: λ, τ ; Output: $\hat{\tau}, \text{mark}$)

begin
 if $\lambda = \emptyset$ **then**
 $\hat{\tau} \leftarrow \text{MakeRimHook}(\tau; \hat{\tau})$
 $\text{mark} \leftarrow \text{false}$
 else ($* \lambda \neq \emptyset *$)
 $j \leftarrow 1$
 repeat
 $\hat{\tau} \leftarrow \text{RepeatedSlide}_{j, \emptyset}(\text{tail of } \tau, \text{NE})$
 until $\lambda \cap \hat{\tau} = \emptyset$
 $j \leftarrow 1$
 repeat
 $\hat{\tau} \leftarrow \text{RepeatedSlide}_{j, \lambda}(\text{head of } \hat{\tau}, \text{SW})$
 until $\hat{\tau}$ is legal on λ
 if $\hat{\tau}$ contains a non-reflected cell **then**

```

if the number of non-reflected cells of  $\hat{\tau}$  is bigger
    than the number of reflected cells of  $\hat{\tau}$  then
         $\hat{\tau} \leftarrow \hat{\tau}$  with no circle on the 1st tail of  $\hat{\tau}$ 
        mark  $\leftarrow$  false
    else
         $\hat{\tau} \leftarrow \hat{\tau}$  with a circle on the 1st tail of  $\hat{\tau}$ 
        mark  $\leftarrow$  false
    else (* every cell of  $\hat{\tau}$  is reflected *)
        mark  $\leftarrow$  true

```

end .

If $\hat{\tau}$ has an illegal tail on λ , then $RepeatedSlide_{1,\lambda}(\text{head of } \hat{\tau}, SW)$ has a legal head on λ . Thus, both loops in the above algorithm must terminate and the resulting $\hat{\tau}$ is an outside rim hook of λ . See Figure 3.1.

Let T be a shifted (first tail circled) rim hook tableau. We denote by T_j the shifted (first tail circled) rim hook tableau obtained from T by removing all the rim hooks whose entry is larger than j . Similarly, we denote by T^j the shifted skew (first tail circled) rim hook tableau obtained from T by removing all the rim hooks whose entry is smaller than j . See Figure 3.2. Let λ and α be shapes of T_{j-1} and T^j , respectively.

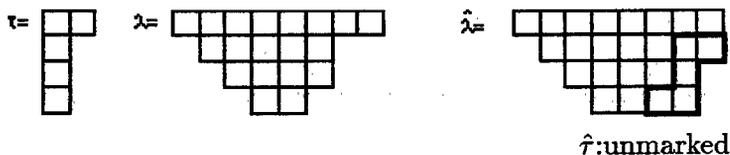


Figure 3.1

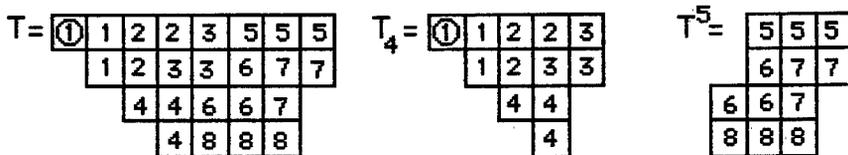


Figure 3.2

Now suppose T has shape μ and content $(k, \dots, k, 0, k, \dots, k)$, where 0 lies in the j th coordinate. Let σ be a hook of size k . The output from Algorithm Insert will be another shifted (first tail circled) rim hook tableau \hat{T} of content (k, \dots, k) and shape $\hat{\mu}$ such that $\hat{\sigma} = \hat{\mu} - \mu$ is an outside rim hook of μ , and a mark of $\hat{\sigma}$.

Algorithm Insert (Input: T, σ, j ; Output: $\hat{T}, \hat{\sigma}$, mark)
begin

Position $(\lambda, \sigma; \sigma_1, \text{mark})$
 $A \leftarrow T_{j-1} \cup \sigma_1(j)$
 $B \leftarrow T^{j+1}$
 mark \leftarrow mark of σ_1
while B contains finite entries **do**
 BumpOut $(A, B, \text{mark}; \hat{A}, \hat{B}, \text{mark})$
 $\sigma_{\hat{A}\hat{B}} \leftarrow$ bumping hook of \hat{A} and \hat{B}
 $A \leftarrow \hat{A}$
 $B \leftarrow \hat{B}$
 mark \leftarrow mark of bumping hook of \hat{A} and \hat{B}
 $\hat{T} \leftarrow \hat{A}$
 $\hat{\sigma} \leftarrow \sigma_{\hat{A}\hat{B}}$
 mark \leftarrow mark of $\sigma_{\hat{A}\hat{B}}$

end .

At the end, $\hat{A} = \hat{T}$, and \hat{B} contains infinite entries only. By the result of Section 2, $\sigma_{\hat{A}\hat{B}}$ is the intersection of \hat{T} and \hat{B} , and $\sigma_{\hat{A}\hat{B}}$ is an outside rim hook of μ .

Figure 3.3 gives an example of the Insert algorithm. T and σ (with j 's in the cells of σ) are given in Figure 3.3 (a). Then Figure 3.3 (b)–(g) describe A and B (with cells in σ_{AB} indicated in heavy outline) at each pass through the main loop. We also give the mark of the bumping hook σ_{AB} and the appropriate case number from BumpOut.

We now describe Algorithm Delete which reverses the Insert algorithm. In this algorithm we use the BumpIn algorithm in the previous section.

First, we need Algorithm Hook to reverse the Position algorithm described earlier. Suppose λ is a shifted shape and τ ($|\tau| = k$) is an outside rim hook of λ with a mark. The output from the Hook algorithm will be a hook $\hat{\tau}$ of the size k .

Algorithm Hook (Input: λ, τ , mark; Output: $\hat{\tau}$)

```

begin
  if  $\lambda = \emptyset$  then
     $\hat{\tau} \leftarrow \text{MakeHook}(\tau; \hat{\tau})$ 
  else (*  $\lambda \neq \emptyset$  *)
    if  $\tau$  is marked then
       $x \leftarrow$  head of  $\tau$ 
       $d \leftarrow$  SW
    else if  $\tau$  is unmarked and single then
       $x \leftarrow$  tail of  $\tau$ 
       $d \leftarrow$  NE
    else if  $\tau$  has a circle on its first tail then
       $x \leftarrow$  1st head of  $\tau$ 
       $d \leftarrow$  SW
    else (*  $\tau$  has no circle on its first tail *)
       $x \leftarrow$  2nd head of  $\tau$ 
       $d \leftarrow$  SW
     $j \leftarrow 1$ 
    repeat
       $\hat{\tau} \leftarrow \text{RepeatedSlide}_{j,\lambda}(x, d)$ 
    until  $\hat{\tau}$  is contained in the first row
     $j \leftarrow 1$ 
    repeat
       $\hat{\tau} \leftarrow \text{RepeatedSlide}_{j,\emptyset}(\text{head of } \hat{\tau}, \text{SW})$ 
    until  $\hat{\tau}$  intersects the first column
  end .

```

Certainly we have

LEMMA 3.1. *Position and Hook are inverses of one another. That is, the procedure:*

```

begin
  Position ( $\lambda, \tau; \hat{\tau}$ , mark)
  Hook ( $\lambda, \hat{\tau}$ , mark;  $\hat{\hat{\tau}}$ )
end .
yields  $\tau = \hat{\hat{\tau}}$ ; and the procedure:
begin
  Hook ( $\lambda, \tau$ , mark;  $\hat{\tau}$ )
  Position ( $\lambda, \hat{\tau}; \hat{\hat{\tau}}$ , mark)

```

end .

also yields $\tau = \hat{\tau}$ and $\text{mark of } \tau = \text{mark of } \hat{\tau}$ under the special circumstances that Hook will be used.

The Delete algorithm has as input a shifted (first tail circled) rim hook tableau T of shape μ and content $\rho = k^m$, and an outer rim hook σ ($|\sigma| = k$) of μ with a mark.

Algorithm Delete will produce the following:

A shifted (first tail circled) rim hook tableau \hat{T} of shape $\hat{\mu}$ and content $\hat{\rho}$; a value j ; and a hook $\hat{\sigma}$ such that

- (a) $\hat{\rho} = (\rho_1, \dots, \rho_{j-1}, 0, \rho_{j+1}, \dots, \rho_m)$,
- (b) $\rho_j = |\hat{\sigma}| = |\sigma| = k$,
- (c) $\hat{\mu} = (\mu - \sigma)$.

Algorithm Delete (Input: T, σ, mark ; Output: $\hat{T}, \hat{\sigma}, j, \text{mark}$),

begin

$A \leftarrow T$

$B \leftarrow \sigma(\infty)$

$\text{mark} \leftarrow \text{mark of } \sigma$

repeat

 BumpIn (A, B, mark ; $\hat{A}, \hat{B}, j, \text{timetostop}, \text{mark}$)

$\sigma_{\hat{A}\hat{B}} \leftarrow \text{bumping hook of } \hat{A} \text{ and } \hat{B}$

$\text{mark} \leftarrow \text{mark of } \sigma_{\hat{A}\hat{B}}$

$A \leftarrow \hat{A}$

$B \leftarrow \hat{B}$

until timetostop

$\hat{T} \leftarrow \hat{A} \cup \hat{B}$

$\text{mark} \leftarrow \text{mark of } \sigma_{\hat{A}\hat{B}}$

$\lambda \leftarrow \text{shape of } \hat{A}$

Hook ($\lambda - \sigma_{\hat{A}\hat{B}}, \sigma_{\hat{A}\hat{B}}, \text{mark}; \hat{\sigma}$)

end .

Figure 3.4 shows an example for Algorithm Delete. In Figure 3.4 (a), T and σ (with cells of σ indicated in heavy outline) are given. Then Figure 3.4 (b)–(e) describe A and B (with cells in σ_{AB} indicated in heavy outline) at each pass through the main loop. Again we give the mark of the bumping hook σ_{AB} , and the appropriate case number from BumpOut and BumpIn.

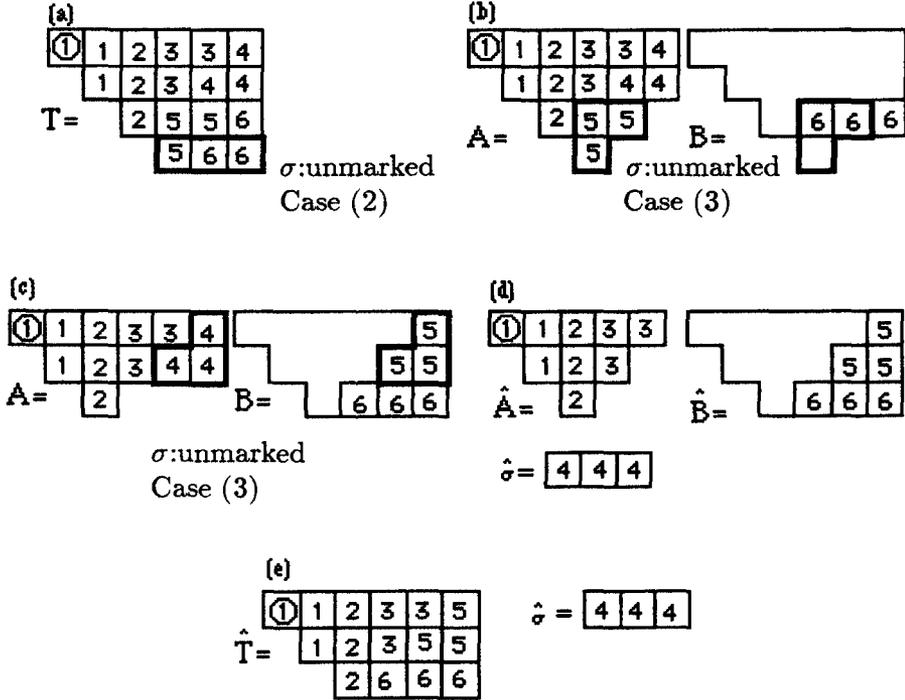


Figure 3.4

From Lemma 2.2 and Lemma 3.1 we have

THEOREM 3.2. *Insert and Delete are inverses of one other. That is, the procedure:*

begin

Insert ($T, \sigma, j; \hat{T}, \hat{\sigma}, \text{mark}$)

Delete ($\hat{T}, \hat{\sigma}, \text{mark}; \hat{\hat{T}}, \hat{\hat{\sigma}}, j, \text{mark}$)

end .

yields $T = \hat{\hat{T}}, \sigma = \hat{\hat{\sigma}}$; and the procedure:

begin

Delete ($T, \sigma, \text{mark}; \hat{T}, \hat{\sigma}, j, \text{mark}$)

Insert ($\hat{T}, \hat{\sigma}, j; \hat{\hat{T}}, \hat{\hat{\sigma}}, \text{mark}$)

end .

will yield $T = \hat{T}, \sigma = \hat{\sigma}$ and $\text{mark of } \sigma = \text{mark of } \hat{\sigma}$.

4. Schensted correspondence

DEFINITION 4.1. A *circled shifted rim hook tableau* is a shifted first tail circled rim hook tableau with every cell except main diagonal cells either circled or uncircled.

DEFINITION 4.2. $\mathcal{H} = (H_1, H_2, \dots, H_m)$ is said to be a *circled hook permutation* of content $\rho = k^m$, and shape $(\tau^{(1)}, \tau^{(2)}, \dots, \tau^{(m)})$ if the following conditions hold:

- (1) each H_i is a hook tableau of shape $\tau^{(i)}$,
- (2) $|\tau^{(i)}| = k$ and
- (3) for each i , all cells of H_i except its tail can be circled or uncircled.

Figure 4.1 gives a circled hook permutation of content 5^5 .

DEFINITION 4.3. Let τ be a rim hook of size k . Then every cell of τ is numbered as follows: Suppose τ is a single rim hook. If $x \in \tau$ is d distant from head (τ), x is numbered as $d + 1$. Suppose now τ is a double rim hook with $|\beta_1[\tau]| = a$ and $|\beta_2[\tau]| = b$ ($a > b$). We number cells of $\beta_1[\tau]$ with $1, 2, \dots, a$ as above, and then number cells of $\beta_2[\tau]$ with $a + 1, \dots, k$ in a similar way. If $x \in \tau$ is numbered d , then x is called the *dth cell* of τ .

Let σ and τ be hooks or rim hooks of the same size k . We say that σ and τ have *the same circling* if the following condition holds: The i th cell of σ is circled if and only if the i th cell of τ is also circled for $i = 1, \dots, k - 1$. In Figure 4.2, σ and τ have the same circling.

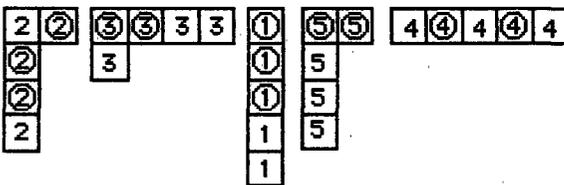


Figure 4.1

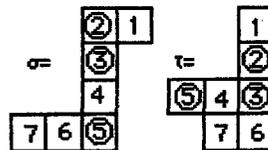


Figure 4.2

We now describe an algorithm which assigns to a circled hook permutation with content k^m a pair consisting of a shifted (first tail circled) rim hook tableau and a circled shifted rim hook tableau of the same shape and content. Let $\mathcal{H} = (H_1, H_2, \dots, H_m)$ be a circled hook permutation of content k^m and shape $(\tau^{(1)}, \tau^{(2)}, \dots, \tau^{(m)})$.

Algorithm Encode (Input: \mathcal{H} ; Output: P, Q)
begin

```

    P, Q ← ∅
    for i ← 1 to m
        j ← content of  $H_i$ 
        Insert ( $P, \tau^{(i)}, j; \hat{P}, \sigma, \text{mark}$ )
         $\sigma \leftarrow \sigma$  with the same circling as  $\tau^{(i)}$ 
        if  $\sigma$  is unmarked then
             $\hat{Q} \leftarrow Q \cup \sigma(i)$  with no circle on the tail of  $\sigma$ 
             $Q \leftarrow \hat{Q}$ 
             $P \leftarrow \hat{P}$ 
        else (*  $\sigma$  is marked *)
             $\hat{Q} \leftarrow Q \cup \sigma(i)$  with a circle on the tail of  $\sigma$ 
             $Q \leftarrow \hat{Q}$ 
             $P \leftarrow \hat{P}$ 

```

end .

For a circled hook permutation given in Figure 4.1, Figure 4.3 shows the final P and Q .

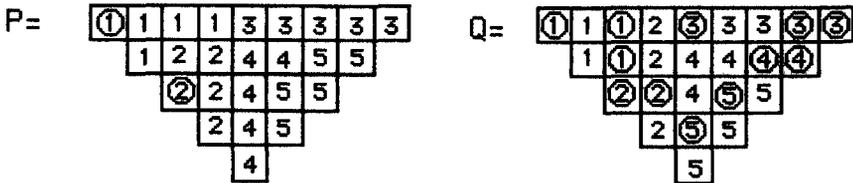


Figure 4.3

Now we construct Algorithm Decode which is the inverse of Encode. Suppose P is a shifted (first tail circled) rim hook tableau of shape $\lambda \in DP$ and content $\rho = k^m$ and Q is a circled shifted rim hook tableau

of shape λ and content ρ . Algorithm Decode has as input a pair (P, Q) . The result is a circled hook permutation $\mathcal{H} = (H_1, \dots, H_m)$ of content $\rho = k^m$ and shapes $(\tau^{(1)}, \dots, \tau^{(m)})$. In this case, $\rho_i = k$ for all i .

Algorithm Decode (Input: P, Q ; Output: (H_1, \dots, H_m))

begin

if P and Q are both empty **then**

 outcome \leftarrow decoding

else

if $\kappa_Q(l)$ has no circle on its tail **then**

 mark \leftarrow false

else (* $\kappa_Q(l)$ has a circle on its tail *)

 mark \leftarrow true

$\sigma \leftarrow \kappa_Q(l)$

 Delete $(P, \sigma, \text{mark}; \tilde{P}, \tilde{\sigma}, j, \text{mark})$

$\hat{Q} \leftarrow Q - \kappa_Q(l)(l)$

 Decode $(\tilde{P}, \hat{Q}; \hat{\tilde{P}}, \hat{\tilde{Q}}, (H_1, \dots, H_{m-1}),)$

else (* outcome is decoding *)

$\tilde{\sigma} \leftarrow \tilde{\sigma}$ with the same circling as $\kappa_Q(l)$

$H_m \leftarrow \tilde{\sigma}(j)$

end .

THEOREM 4.4. *Encode and Decode are inverses of one another.*

That is, the procedure:

begin

 Decode $(P, Q; \hat{P}, \hat{Q}, \mathcal{H} = (H_1, \dots, H_m))$

 Encode $(\mathcal{H}; \hat{\hat{P}}, \hat{\hat{Q}})$

end .

yields $P = \hat{\hat{P}}$ and $Q = \hat{\hat{Q}}$, and the procedure:

begin

 Encode $(\mathcal{H}; P, Q)$

 Decode $(P, Q; \hat{\mathcal{H}})$

end .

yields $\mathcal{H} = \hat{\mathcal{H}}$.

The algorithms and theorems of Section 2 and 3 yield the following bijection:

THEOREM 4.5. *Let k be an odd number. Then algorithms Encode and Decode construct a bijection between all pairs (P, Q) , where P is a shifted (first tail circled) k -rim hook tableau of shape λ and content k^m and Q is a circled shifted k -rim hook tableau of the same shape λ and content k^m , and all circled hook permutations of content k^m .*

References

- [B] A. Berele, *A Schensted-type correspondence for the symplectic group*, J. Combin. Theory. A **43** (1986), 320–328.
- [Kn1] D. E. Knuth, *Sorting and Searching*, “The Art of Computer Programming,” Vol. 3, Addison-Wesley, Mass., 1973.
- [Kn2] D. E. Knuth, *Permutations, matrices, and generalized Young tableaux*, Pacific J. Math. **34** (1970), 709–727.
- [Sa] B. E. Sagan, *Shifted tableaux, Schur Q -functions and a conjecture of R. Stanley*, J. Combin. Theory (A) **45** (1987), 62–103.
- [S] C. Schensted, *Longest increasing and decreasing subsequences*, Canad. J. Math. **13** (1961), 179–191.
- [SS] B. Sagan and R. Stanley, *Robinson-Schensted algorithms for skew tableaux*, J. Combin. Theory (A), to appear.
- [SW] D. W. Stanton and D. E. White, *A Schensted algorithm for rim hook tableaux*, J. Combin. Theory (A) **40** (1985), 211–247.
- [W] D. E. White, *A bijection proving orthogonality of the characters of S_n* , Advances in Math. **50** (1983), 160–186.
- [Wo] D. R. Worley, *A theory of Shifted Young Tableaux*, Ph. D. thesis, M.I.T., 1984.

Department of Mathematics
Hallym University
Chuncheon 200-20, Korea