Kangweon-Kyungki Math. Jour. **14** (2006), No. 1, pp. 125–136

SCHENSTED INSERTION AND DELETION ALGORITHMS FOR SHIFTED RIM HOOK TABLEAUX

Jaejin Lee

Abstract. Using the Bumping algorithm for the shifted rim hook tableaux described in [5], we construct Schensted insertion and deletion algorithms for shifted rim hook tableaux. This may give us the combinatorial proof for the orthogonality of the second kind of the spin characters of S_n.

1. Introduction

In [8] Schensted constructed the Schensted algorithm giving a bijection between permutations and pairs of Young standard tableaux (see also [2]). After Knuth generalized it to column strict tableaux in [3], various analogs of the Schensted algorithm came. See [1], [4], [6], [7], [9], [10] and [11].

In [10] White gives Schensted algorithm for rim hook tableaux. The basic idea of the proof is as follows. A rim hook σ, called the "attacking hook", percolates in a generally outward (SE) direction in a rim hook tableau P, bumping rim hooks out of its way. When the attacking hook reaches the outer boundary of P, a new P has been created and the location of the attacking hook marks where to begin the reverse process and thus the location of the last rim hook in a corresponding Q.

In this paper we will give similar Schensted algorithm for shifted rim hook tableaux using the Bumping algorithm for shifted rim hook tableaux described in [5].

In section 2, we construct Schensted insertion algorithm for shifted rim hook tableaux. Schensted deletion algorithm for shifted rim hook

Received May 18, 2006.
2000 Mathematics Subject Classification: 05E10.
Key words and phrases: Schensted algorithm, shifted rim hook tableaux.

tableaux is given in section 3. See [5] for definitions and notations not described in this paper.

2. Schensted insertion algorithm

Using the Bumping algorithms in [5] 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 odd (or 0) and a hook tableau of odd size. The hook tableau must first be positioned so that we can apply Algorithm BumpOut in [5]. Suppose λ is a shifted shape and τ is a hook of odd size.

Algorithm Position (Input: λ, τ; Output: $\hat{\tau}$, mark)
begin
    if $\lambda = \emptyset$ **then**
        $\hat{\tau} \leftarrow$ MakeRimHook($\tau; \hat{\tau}$)
        mark$\leftarrow$ false
    else $(* \ \lambda \neq \emptyset \ *)$
        $j \leftarrow 1$
        repeat
            $\hat{\tau} \leftarrow \mathit{RepeatedSlide}_{j,\emptyset}(\text{tail of } \tau, \text{NE})$
        until $\lambda \cap \hat{\tau} = \emptyset$
        $j \leftarrow 1$
        repeat
            $\hat{\tau} \leftarrow \mathit{RepeatedSlide}_{j,\lambda}(\text{head of } \hat{\tau}, \text{SW})$
        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 first 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←true

end.

If $\hat{\tau}$ has an illegal tail on λ, then $\mathit{RepeatedSlide}_{1,\lambda}(\text{head of } \hat{\tau}, \text{SW})$ 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 2.1*.

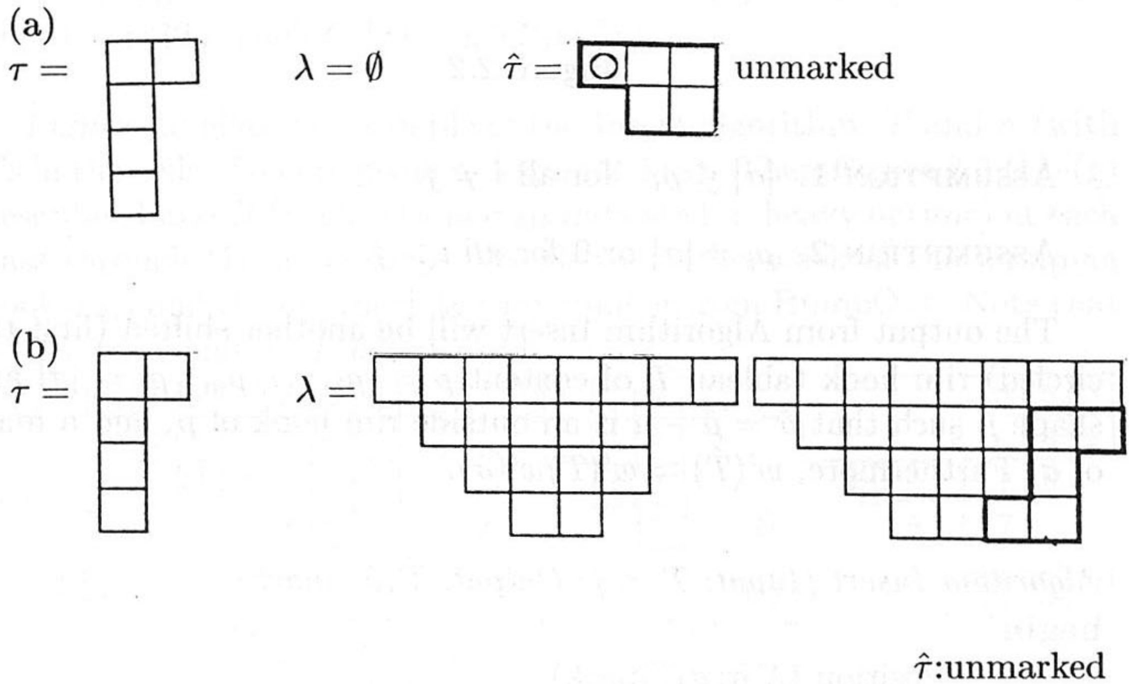


Figure 2.1

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 2.2*.

Let λ and α be shapes of T_{j-1} and T^j, respectively. Now suppose T has shape μ and content $(\rho_1, \ldots, \rho_{j-1}, 0, \rho_{j+1}, \ldots, \rho_m)$, with ρ_i odd or 0. Let σ be a hook of odd size. We make the following two assumptions about $|\sigma|$ so that σ bumps only rim hooks of equal length:

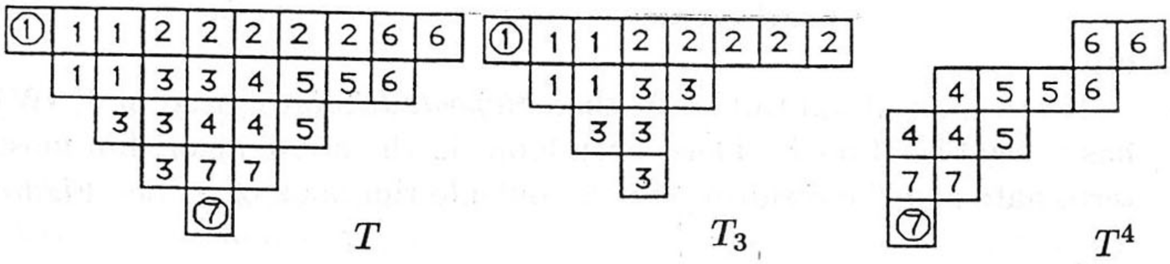


Figure 2.2

Assumption 1. $|\sigma| \le \rho_i$ for all $i \neq j$.

Assumption 2. $\rho_i = |\sigma|$ *or* 0 *for all* $i > j$.

The output from Algorithm Insert will be another shifted (first tail circled) rim hook tableau $\hat{T}$ of content $\rho = (\rho_1, \ldots, \rho_m), \rho_j = |\sigma|$ and shape $\hat{\mu}$ such that $\hat{\sigma} = \hat{\mu} - \mu$ is an outside rim hook of μ, and a mark of $\hat{\sigma}$. Furthermore, $w'(\hat{T}) = w'(T)w'(\hat{\sigma})$.

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

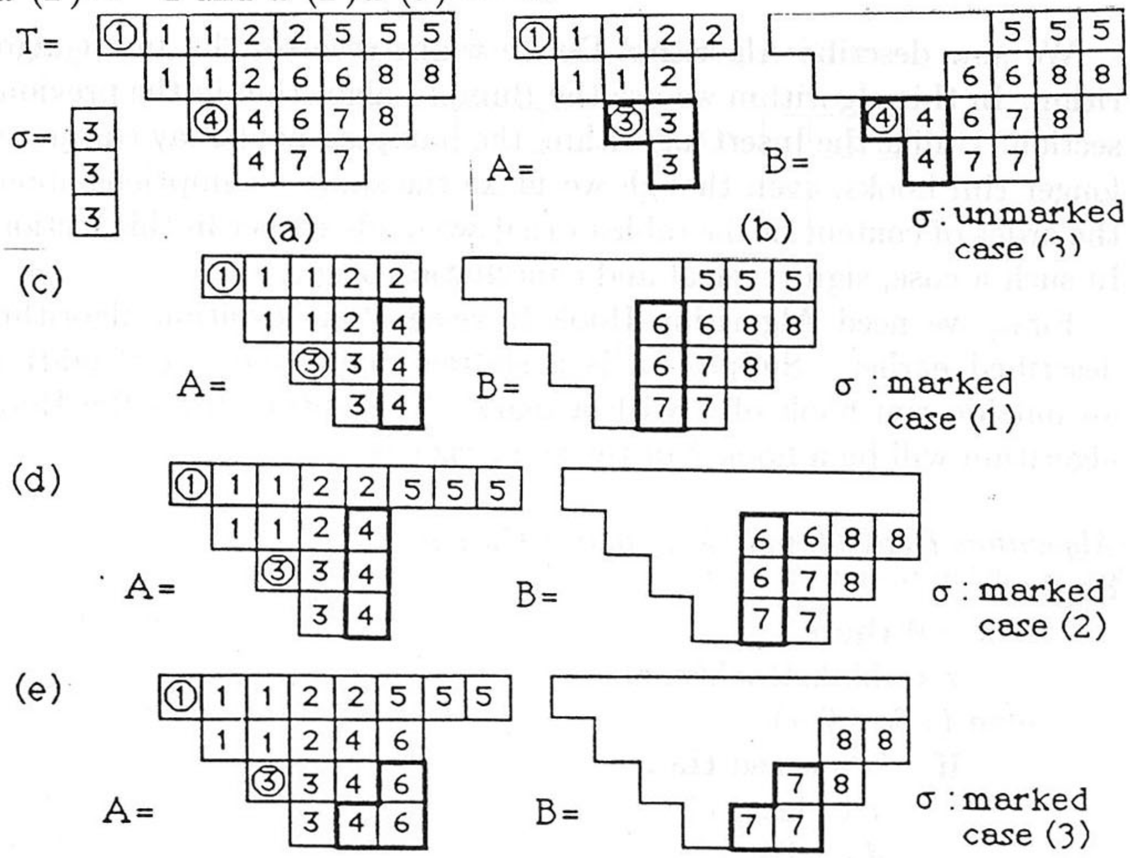
> Position $(\lambda, \sigma; \sigma_1$, mark)
> $A \leftarrow T_{j-1} \cup \sigma_1(j)$
> $B \leftarrow T^{j+1}$
> mark$\leftarrow$ mark of σ_1
> **while** B contains finite entries **do**
>
> > BumpOut (A, B, mark; $\hat{A}, \hat{B}$, mark, direction)
> > $\sigma_{\hat{A}\hat{B}} \leftarrow$ bumping 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 [5], $\sigma_{\hat{A}\hat{B}}$ is the intersection of $\hat{T}$ and $\hat{B}$, and $\sigma_{\hat{A}\hat{B}}$ is an outside rim hook of μ.

Because of Assumption 1 and 2, no direction reversal occurs and we can use BumpOut exclusively. Hence $w'(A)w'(B)w'(\sigma)$ is invariant. At the first step, $w'(A) = w'(T_{j-1})w'(\sigma)$, $w'(B) = w'(T^{j+1})$ and $w'(\sigma_{AB}) = w'(\sigma_1)$. At the end, $w'(\hat{A}) = w'(\hat{T})$, $w'(B) = 1$ and $w'(\sigma_{\hat{A}\hat{B}}) = w'(\hat{\sigma})$. Since $w'(A)w'(B)w'(\sigma)$ is invariant, we have $w'(T) = w'(T_{j-1})w'(T^{j+1}) = w'(\hat{T})w'(\hat{\sigma})$.

Figure 2.3 gives an example of the Insert algorithm. T and σ (with j's in the cells of σ) are given in Figure 2.3 (a). Then Figure 2.3 (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. Note that $w'(T) = -1$ and $w'(\hat{T})w'(\hat{\sigma}) = -1$.



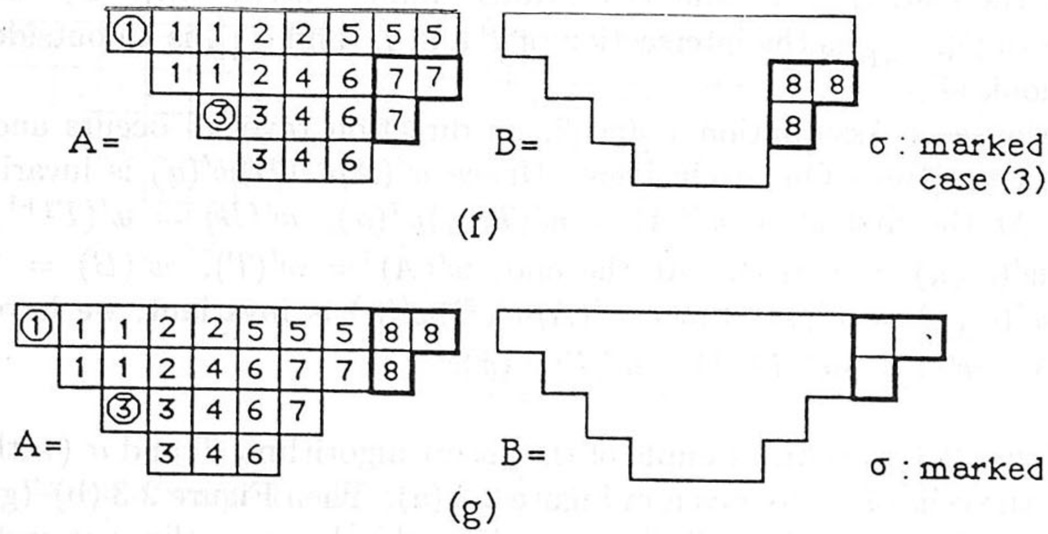


Figure 2.3

3. Schensted deletion algorithms

We now describe Algorithm Delete which reverses the Insert algorithm. In this algorithm we use the BumpIn algorithm in the previous section. Unlike the Insert algorithm, the bumping hook may encounter longer rim hooks, even though we make the same assumptions about the order of content of the tableau that we made earlier in this section. In such a case, sign reversal and cancellation occur.

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

Algorithm Hook (Input: λ, τ, mark; Output: $\hat{\tau}$)
begin
  if $\lambda = \emptyset$ **then**
    $\hat{\tau} \leftarrow$ MakeHook$(\tau; \hat{\tau})$
  else $(* \ \lambda \neq \emptyset \ *)$
    if τ is marked **then**
      $x \leftarrow$ head of τ
      $d \leftarrow$ SW
    else if τ is unmarked and single **then**
      $x \leftarrow$ tail of τ

$d \leftarrow$ NE

else if τ has a circle on its first tail **then**

$x \leftarrow$ 1st head of τ

$d \leftarrow$ SW

else ($*$ τ has no circle on its first tail $*$)

$x \leftarrow$ 2nd head of τ

$d \leftarrow$ SW

$j \leftarrow 1$

repeat

$\hat{\tau} \leftarrow \mathit{RepeatedSlide}_{j,\lambda}(x, d)$

until $\hat{\tau}$ is contained in the first row

$j \leftarrow 1$

repeat

$\hat{\tau} \leftarrow \mathit{RepeatedSlide}_{j,\emptyset}(\text{head of } \hat{\tau}, \text{SW})$

until $\hat{\tau}$ intersects the first column

end .

Figure 3.1 shows an example of the Hook algorithm.

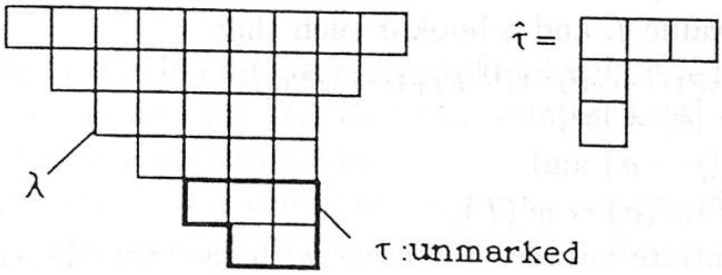


Figure 3.1

Certainly we have

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

begin

Position $(\lambda, \tau; \hat{\tau}, \text{mark})$

Hook $(\lambda, \hat{\tau}, \text{mark}; \hat{\hat{\tau}})$

end.

yields $\tau = \hat{\hat{\tau}}$; *and the procedure:*

begin

Hook $(\lambda, \tau, \text{mark}; \hat{\tau})$

Position $(\lambda, \hat{\tau}; \hat{\hat{\tau}}, \text{mark})$

end.

also yields $\tau = \hat{\hat{\tau}}$ *and mark of* $\tau =$ *mark of* $\hat{\hat{\tau}}$ *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 $\rho = (\rho_1, \rho_2, \ldots, \rho_m)$, ρ_i odd for $1 \leq i \leq m$, and an outer rim hook σ ($|\sigma|$ odd) of μ with a mark. We make the following assumption so that the bumping hooks do not encounter smaller rim hooks:

Assumption 3. $|\sigma| \leq \rho_i$ for all i.

Algorithm Delete will produce one of the following two outcomes:

(1) A shifted (first tail circled) rim hook tableau $\hat{T}$ of shape $\hat{\mu}$ and content $\hat{\rho}$; and an outer rim hook $\hat{\sigma}$ of $\hat{\mu}$ with a mark such that

(a) $\hat{\rho} = \rho$,

(b) $\hat{\mu} = (\mu - \sigma) \cup \hat{\sigma}$ and

(c) $w'(\hat{T})w'(\hat{\sigma}) = -w'(T)w'(\sigma)$.

(2) 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) $\hat{\rho} = (\rho_1, \ldots, \rho_{j-1}, 0, \rho_{j+1}, \ldots, \rho_m)$,

(b) $\rho_j = |\hat{\sigma}| = |\sigma|$,

(c) $\hat{\mu} = (\mu - \sigma)$ and

(d) $w'(\hat{T})w'(\sigma) = w'(T)$.

We differentiate between the cases with the variable *outcome* which takes values *cancellation* in case (1) and *deletion* in case (2).

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

begin

  $A \leftarrow T$

  $B \leftarrow \sigma(\infty)$

  mark$\leftarrow$ mark of σ

  direction$\leftarrow$ inward

  repeat

    if direction is inward **then**

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

    else $(*$ direction is outward $*)$

BumpOut $(A, B, \text{mark}; \hat{A}, \hat{B}, \text{mark, direction})$
$\sigma_{\hat{A}\hat{B}} \leftarrow$ bumping hook of $\hat{A}$ and $\hat{B}$
mark$\leftarrow$ mark of $\sigma_{\hat{A}\hat{B}}$
$A \leftarrow \hat{A}$
$B \leftarrow \hat{B}$
until B has no finite entries or timetostop
if timetostop **then**
$\hat{T} \leftarrow \hat{A} \cup \hat{B}$
mark$\leftarrow$ mark of $\sigma_{\hat{A}\hat{B}}$
$\lambda \leftarrow$ shape of $\hat{A}$
Hook $(\lambda - \sigma_{\hat{A}\hat{B}}, \sigma_{\hat{A}\hat{B}}, \text{ mark };\hat{\sigma})$
outcome$\leftarrow$ deletion
else $(*\ B$ has no finite entries $*)$
$\hat{T} \leftarrow \hat{A}$
$\hat{\sigma} \leftarrow \sigma_{\hat{A}\hat{B}}$
mark$\leftarrow$ mark of $\sigma_{\hat{A}\hat{B}}$
outcome$\leftarrow$ cancellation
end.

Recall that direction reversals occur if and only if sign reversals take place. Hence, no pair (A, B) can be encountered twice in the Delete algorithm, i.e., it must terminate.

If the outcome is a cancellation, direction reversals have occurred an odd number of times. Therefore, $w'(\hat{T})w'(\hat{\sigma}) = -w'(T)w'(\sigma)$. If the outcome is a deletion, the final direction must be inward, so $w'(A)w'(B)w'(\sigma_{AB})$ has changed signs an even number of times. Thus, $w'(\hat{T}) = w'(T)w'(\sigma)$. *Figure 3.2* shows an example where deletion occurs, while *Figure 3.3* shows an example where cancellation occurs.

In Figure 3.2 (a) and Figure 3.3 (a), T and σ (with cells of σ indicated in heavy outline) are given respectively. Then Figure 3.2 (b)–(e) and Figure 3.3 (b)–(f) 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. Note that $w'(\hat{T}) = w'(T)w'(\sigma) = -1$ in Figure 3.2 and $w'(T)w'(\sigma) = -1$, $w'(\hat{T})w'(\hat{\sigma}) = +1$ in Figure 3.3.

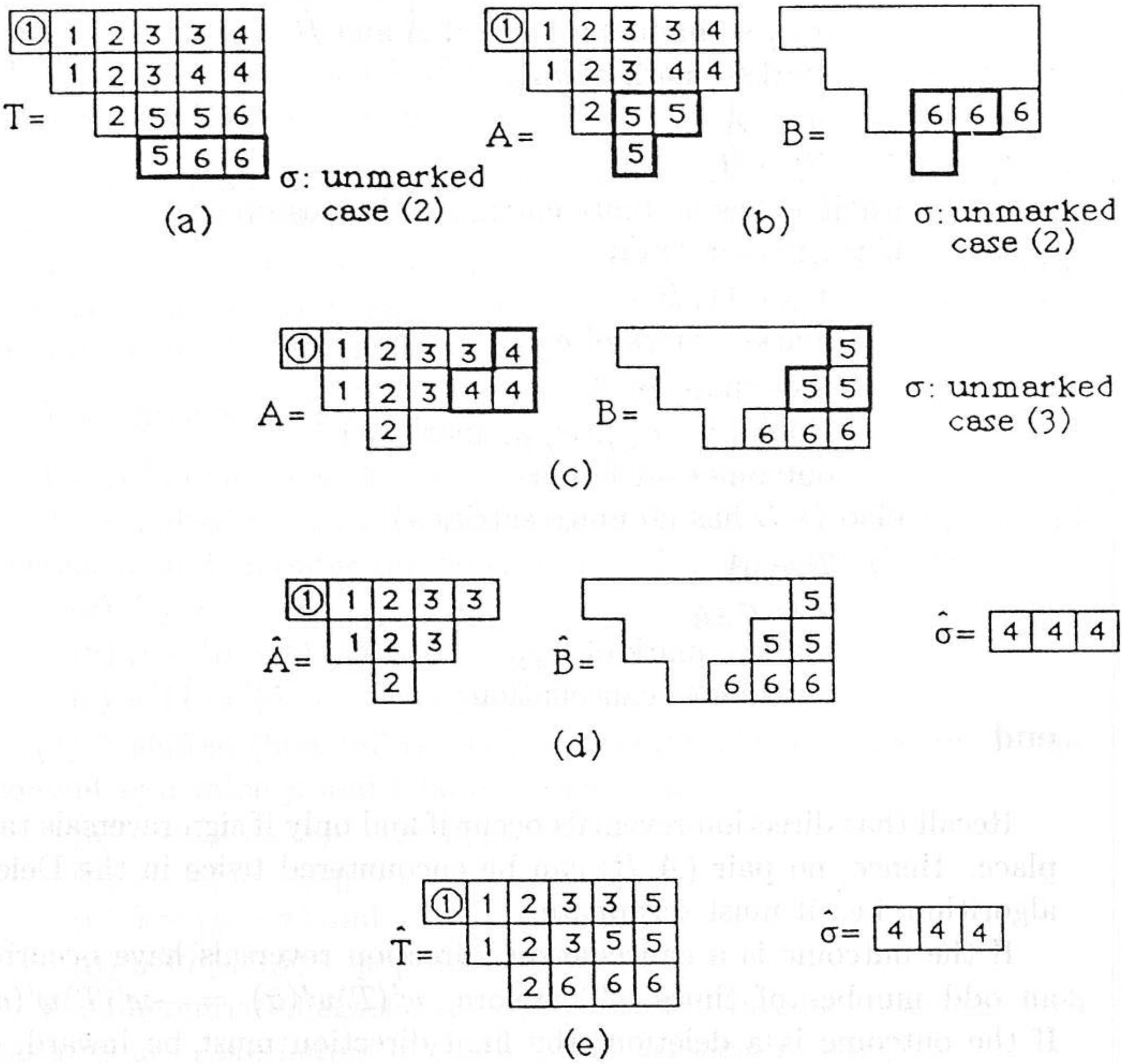


Figure 3.2

From Lemma 3.1 we have

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

begin

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

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

end.

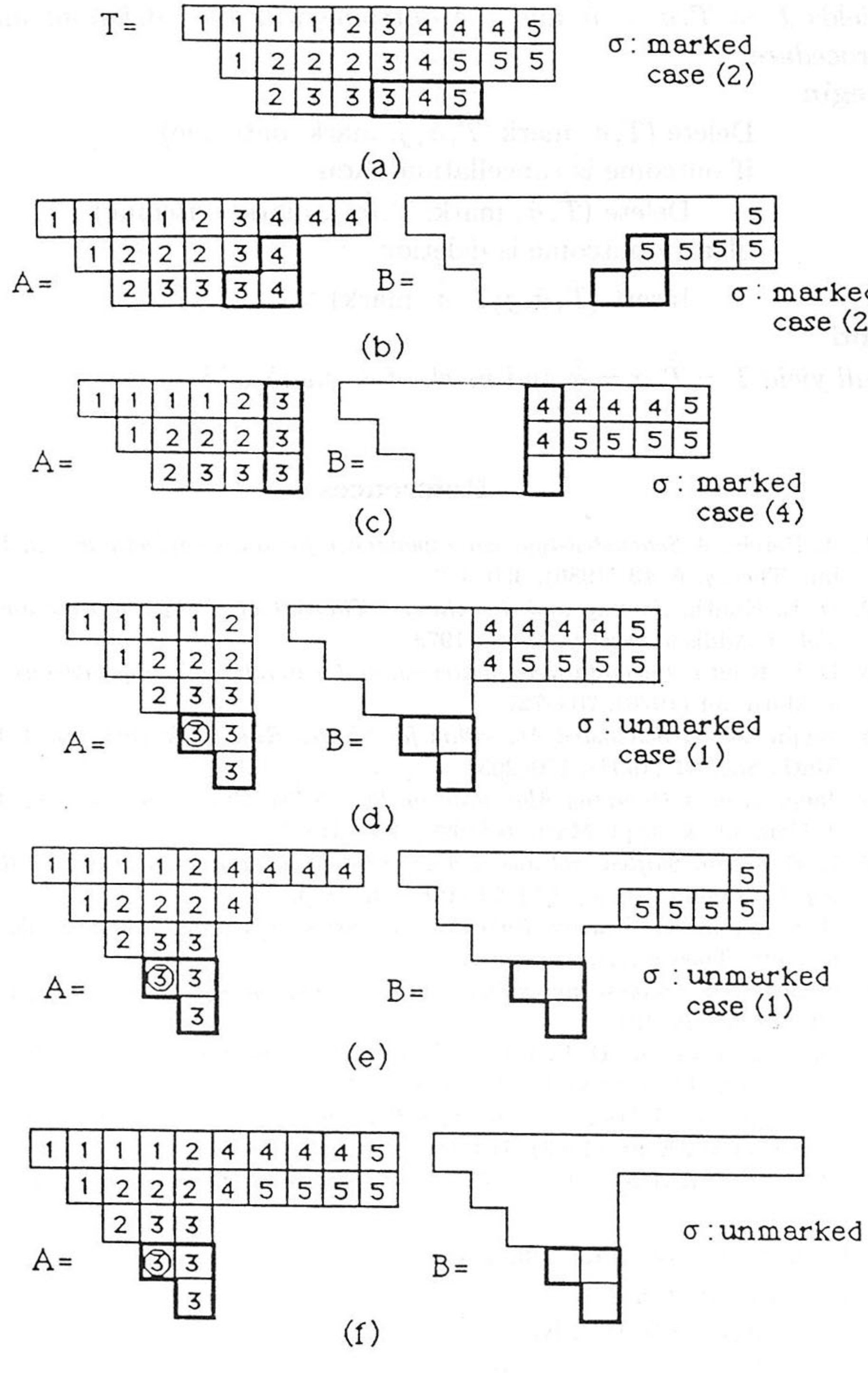

T=
σ: marked
case (2)
(a)
A=
B=
σ: marked
case (2)
(b)
A=
B=
σ: marked
case (4)
(c)
A=
B=
σ: unmarked
case (1)
(d)
A=
B=
σ: unmarked
case (1)
(e)
A=
B=
σ: unmarked
(f)

Figure 3.3

yields $T = \hat{\hat{T}}, \sigma = \hat{\hat{\sigma}}$ *and the outcome will be a deletion; and the procedure:*

begin

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

if outcome is cancellation **then**

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

else $(*$ outcome is deletion $*)$

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

end.

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

References

1. A. Berele, *A Schensted-type correspondence for the symplectic group*, J. Combin. Theory. A **43** (1986), 320–328.
2. D. E. Knuth, *Sorting and Searching, "The Art of Computer Programming," Vol. 3*, Addison-Wesley, Mass., 1973.
3. D. E. Knuth, *Permutations, matrices, and generalized Young tableaux*, Pacific J. Math. **34** (1970), 709–727.
4. Jaejin Lee, *A Schensted Algorithm for Shifted Rim Hook Tableaux*, J. Korean Math. Soc. **31** (1994), 179–203.
5. Jaejin Lee, *A Bumping Algorithm on the Shifted Rim Hook Tableaux*, Korean J. Comput. & Appl. Math. **6** (1999), 901–914.
6. B. E. Sagan, *Shifted tableaux, Schur Q-functions and a conjecture of R. Stanley*, J. Combin. Theory (A) **45** (1987), 62–103.
7. B. Sagan and R. Stanley, *Robinson-Schensted algorithms for skew tableaux*, J. Combin. Theory (A), to appear.
8. C. Schested, *Longest increasing and decreasing sucsequences*, Canad. J. Math. **13** (1961), 179–191.
9. D. W. Stanton and D. E. White, *A Schensted algorithm for rim hook tableaux*, J. Combin. Theory (A) **40** (1985), 211–247.
10. D. E. White, *A bijection proving orthogonality of the characters of S_n*, Advances in Math. **50** (1983), 160–186.
11. D. R. Worley, *A theory of Shifted Young Tableaux*, Ph. D. thesis, M.I.T., 1984.

Department of Mathematics
Hallym University
Chunchon, 200–702, Korea
E-mail: `jjlee@hallym.ac.kr`