

A Bijective Proof of the Hook-Length Formula

D. S. FRANZBLAU

*Department of Mathematics, Massachusetts Institute of Technology, Cambridge,
Massachusetts 02139*

AND

DORON ZEILBERGER

*Department of Theoretical Mathematics, The Weizmann Institute of Science, Rehovot
76100, Israel*

Received February 1, 1981; revised January 21, 1982

A well-known theorem of Frame, Robinson, and Thrall states that if λ is a partition of n , then the number of Standard Young Tableaux of shape λ is $n!$ divided by the product of the hook-lengths. We give a new combinatorial proof of this formula by exhibiting a bijection between the set of unsorted Young Tableaux of shape λ , and the set of pairs (T, S) , where T is a Standard Young Tableau of shape λ and S is a "Pointer" Tableau of shape λ .

1. INTRODUCTION

Let $\lambda = \{\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m\}$ be a partition of n . The Ferrers diagram of shape λ is an array of cells indexed by $\{(i, j) \mid 1 \leq i \leq m; 1 \leq j \leq \lambda_i\}$. If A is a set of positive integers such that $|A| = n$, then a Young Tableau (YT) of shape λ on A is any arrangement of A in the cells of the Ferrers diagram of λ . Such an arrangement is a Standard Young Tableau (SYT), if the rows and columns are increasing sequences (see Fig. 1).

Let $[n] = \{1, 2, \dots, n\}$. Then the number of Standard Young Tableaux on $[n]$ of shape λ is denoted f_λ . For each cell (i, j) in the Ferrers diagram of λ , define the hook to be the set of cells $H_{ij} = \{(k, l) \mid k = i \text{ and } l \geq j \text{ or } k \geq i \text{ and } l = j\}$. The hook-length of (i, j) , h_{ij} , is $|H_{ij}|$.

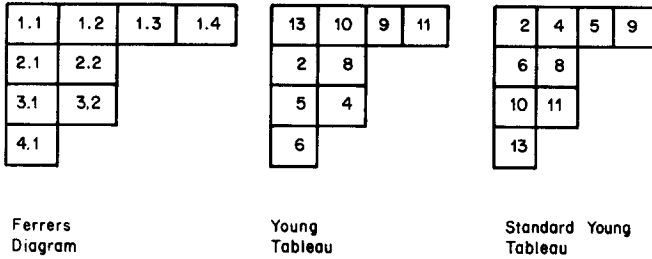


FIG. 1. $\lambda = (4, 2, 2, 1)$, $n = 9$. $A = \{2, 4, 5, 6, 8, 9, 10, 11, 13\}$.

THEOREM (Frame–Robinson–Thrall) [1]. *If λ is a partition of n , then*

$$f_\lambda = \frac{n!}{\prod_{(i,j)} h_{ij}}, \tag{1}$$

where the product is taken over all cells in the Ferrers diagram of λ .

An excellent introduction to Young Tableaux, and a classical proof of this formula, can be found in Knuth [4]. Another nice proof was given by Hillman and Grassl [3]. More recently, Greene *et al.* [2] found a probabilistic interpretation of the formula, and provided a procedure for constructing an SYT uniformly at random. Remmel [5] has given a very different sort of bijective proof, using a lattice path interpretation of Young Tableaux. So far, however, no direct and simple bijective proof has been found.

We show that there is a simple bijection establishing the formula. However, it is nontrivial to verify that the mapping is a bijection.

We first rewrite (1) as

$$n! = f_\lambda \cdot \prod h_{ij} \tag{1}'$$

and note that $n!$ is the number of Young Tableaux on $[n]$ of shape λ . To analyze the right side of (1)' we define a *Pointer Tableau* of shape λ to be an assignment of a cell index (i', j') to each cell (i, j) of the Ferrers diagram of λ , such that $(i', j') \in H_{ij}$. The number of Pointer Tableaux of shape λ is simply $\prod h_{ij}$. Let $P = \{(T, S) \mid T \text{ is an SYT of shape } \lambda \text{ on } [n]; S \text{ is a Pointer Tableau of shape } \lambda\}$. Then $|P| = f_\lambda \cdot \prod h_{ij}$. We call $(T, S) \in P$ a *Pointed Young Tableau (PYT)*.

Our bijection consists of an algorithm for sorting a Young Tableau R , to produce a unique pair (T, S) in P . The Pointer Tableau S records each step of the sorting algorithm, storing just enough information to enable us to reverse the procedure.

The algorithm makes repeated use of a procedure called IC (Insert Column), which we describe in detail in Section 2. The input is (T, S) , a

PYT, where T is an SYT on some set $A \subseteq [n]$, and a column vector

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}$$

of distinct numbers not already in T . IC first inserts \mathbf{a} into T , obtaining an augmented Tableau (not necessarily an SYT). IC then sorts T and constructs a column vector \mathbf{d} of pointers so that $(T, \mathbf{d}S)$ is a PYT. Starting with a YT R , the final PYT is built up one column at a time, starting with the rightmost (shortest) column, as described in the following procedure.

R-SORT

Input: Young Tableau R on $[n]$ with shape $\lambda = \{\lambda_1 \geq \lambda_2 \cdots \geq \lambda_l\}$.

Output: PYT (T, S) of shape λ on $[n]$.

1. *Initialize:* $(T, S) \leftarrow (\phi, \phi); K \leftarrow \lambda_1$.

2. *While* $K \neq 0$ *do*

$\mathbf{a} \leftarrow K$ th column of R

Call IC (\mathbf{a}, T, S) to obtain an augmented PYT (T, S) .

$K \leftarrow K - 1$

endwhile

exit \square

In Section 3 we present DC (Delete Column), the inverse procedure for IC. DC begins with a PYT (T, S) , where T has m rows, deletes a column vector of length m from T and deletes the first column of S , obtaining a diminished PYT. Applying DC successively, we eventually recover R . More formally, we define:

FIND-R

Input: PYT (T, S) of shape $\lambda = \{\lambda_1 \geq \cdots \geq \lambda_l\}$.

Output: YT R of shape λ on $[n]$.

1. *Initialize:* $R \leftarrow \phi; K \leftarrow 1$.

2. *While* $K \leq \lambda_1$ *do*

Call DC (T, S) to obtain smaller PYT (T, S) and column vector \mathbf{a}

K th column of $R \leftarrow \mathbf{a}$

$K \leftarrow K + 1$

endwhile

exit \square

In Sections 4 and 5 we prove that algorithms IC and DC are effective, and, in Section 6, show that they are inverses of each other. Together, these results are sufficient to show that R-SORT provides a bijection between the set of PYT and the set of Young Tableaux of shape λ . In Section 7 we show that the Pointer Tableau can be viewed as a generalization of the familiar inversion table for a permutation.

2. INSERTING A COLUMN INTO A TABLEAU

We show how to insert each column of R into T and how to construct S . First, some notation.

DEFINITION 2.1. A *Pseudo Tableau* is a Young Tableau whose rows are increasing.

DEFINITION 2.2. Let $a_{k1} < a_{k2} < \dots < a_{ky}$ be the k th row of a Pseudo Tableau. To *INSERT* the number b into row k means to insert b so that the row is still increasing. For example, inserting 7 into 13568 yields 135678. If $a_{ki} < b < a_{ki+1}$ then the new row is renumbered so that $a'_{ki+1} \leftarrow b$, $a'_{ki+2} \leftarrow a_{ki+1}$, and so on.

DEFINITION 2.3. Let $a_{k-11} < a_{k-12} \dots < a_{k-1y}$

$$a_{k1} < \dots < a_{ky'},$$

with $y' \leq y$, be rows $\binom{k-1}{k}$ of a Pseudo Tableau. To *EXCHANGE* a_{k-1u} and a_{kv} means to delete these entries from their original positions and *INSERT* them into rows k and $k - 1$, respectively. For instance, exchanging 3 and 6 in $\binom{1256}{3478}$ yields $\binom{1235}{4678}$. The entries are then renumbered.

(Note: from now on, “insert” and “exchange” will be used interchangeably with *INSERT* and *EXCHANGE*.)

We also need a convenient way to represent Pointer Tableaux. Let S_{ij} be the (i, j) th entry in a PT, S . If $S_{ij} = Cx$ ($x \geq 1$) then S_{ij} points to cell $(i, j + x - 1)$. If $S_{ij} = Ry$ ($y \geq 2$) then S_{ij} points to cell $(i + y - 1, j)$. For example, if $\lambda = (3, 2)$ then the following Pointer Tableaux are equivalent:

$$\begin{array}{ccc|cc} (1, 3) & (2, 2) & (1, 3) & C3 & R2 & C1 \\ (2, 1) & (2, 2) & & C1 & C1 & \end{array}$$

We can now give:

Procedure IC (Insert Column)

Input: (T, S) PYT with l rows;
 a_1

Column vector $\mathbf{a} = \begin{pmatrix} \vdots \\ a_m \end{pmatrix}$; $m \geq l$, $a_i \notin T \forall i$.

Output: Augmented PYT (T, S) , where now $a_i \in T \forall i$. T and S have the same shape as on input, except that an additional column of m cells has been adjoined to each.

$IC(\mathbf{a}, T, S)$

(0) Initialize: For $j = 1$ to m do

INSERT a_j into row j of T
 let (j, z) be the new position of a_j
 $d_j \leftarrow Cz$
endfor

(1) While [T is not a Standard Young Tableau] there is an entry $a_{lt} \in T$ such that $a_{lt} < a_{l-1t}$
 do begin:

(2) $a_{kx} \leftarrow \min\{a_{lt} \mid a_{lt} < a_{l-1t}\}$
 $[a_{kx}$ is the smallest entry out of order]

(3) [Claim: $d_{k-1} = Cy$ for some y]
EXCHANGE a_{kx} and a_{k-1y}
 Let (k, y') be the new position of a_{k-1y}

(4) [Update pointers]

$$d_{k-1} \leftarrow \begin{cases} Ru & \text{if } d_k = Ru - 1, \\ R2 & \text{if } d_k = Cx, \\ Cv & \text{if } d_k = Cv, \quad v \neq x, \end{cases}$$

$$d_k \leftarrow Cy'$$

endwhile

(5) [T is now an SYT]

$$S \leftarrow \mathbf{d}S$$

exit \square

EXAMPLE 1. Application of R-SORT; $\lambda = (4, 3, 3, 2)$.

(R)		(T)		(S)									
9	12	8	1	→	1	3	4	8		C1	R3	C2	C1
2	5	4			2	7	9			C3	C1	C1	
11	3	7			5	10	12			R2	C2	C1	
10	6				6	11				C2	C1		

We use IC to insert in succession:

$$(1), \begin{pmatrix} 8 \\ 4 \\ 7 \end{pmatrix}, \begin{pmatrix} 12 \\ 5 \\ 3 \\ 6 \end{pmatrix}, \text{ and } \begin{pmatrix} 9 \\ 2 \\ 11 \\ 10 \end{pmatrix}$$

$$[1] \begin{array}{c|l} (T) & (d) \\ 1 & C1 \rightarrow \text{exit} \end{array}$$

$$[2] \begin{array}{c|l} 1 & 8 & C2 \\ 4 & & C1 \rightarrow \text{No entry is out of order so we exit.} \\ 7 & & C1 \end{array}$$

$$[3] \begin{array}{c|l} 1 & 8 & 12 & C3 \\ 4 & 5 & & C2 \\ 3 & 7 & & C1 \\ 6 & & & C1 \end{array} \rightarrow \begin{array}{c|l} 1 & 8 & 12 & C3 \\ 3 & 4 & & R2 \\ 5 & 7 & & C1 \\ 6 & & & C1 \end{array}$$

(a) (b)

$$\begin{array}{c|l} 1 & 4 & 8 & R3 \\ 3 & 12 & & C2 \\ 5 & 7 & & C1 \\ 6 & & & C1 \end{array} \rightarrow \begin{array}{c|l} 1 & 4 & 8 & R3 \\ 3 & 7 & & C1 \\ 5 & 12 & & C2 \\ 6 & & & C1 \end{array} \rightarrow \text{exit}$$

(c)

(a) $a_{kx} = 3; k = 3, x = 1. d_2 = C2, a_{22} = 5$ so 3 and 5 are exchanged, $d_3 = Cx$ so $d_2 \leftarrow R2$ and $d_3 \leftarrow C1$.

(b) Now $a_{kx} = 4, k = 2, x = 2. 4$ must be exchanged with $a_{13} = 12. d_2 = R2$ so $d_1 \leftarrow R3, d_2 \leftarrow C2$.

(c) $a_{kx} = 7; a_{k-1y} = 12. d_k = d_3 = C1 \neq Cx$ so $d_2 \leftarrow C1, d_3 \leftarrow C2$.

$$[4] \begin{array}{c|l} 1 & 4 & 8 & 9 & C4 \\ 2 & 3 & 7 & & C1 \\ 5 & 11 & 12 & & C2 \\ 6 & 10 & & & C2 \end{array} \rightarrow \begin{array}{c|l} 1 & 3 & 4 & 8 & C1 \\ 2 & 7 & 9 & & C3 \\ 5 & 11 & 12 & & C2 \\ 6 & 10 & & & C2 \end{array}$$

$$\rightarrow \begin{array}{c|l} 1 & 3 & 4 & 8 & C1 \\ 2 & 7 & 9 & & C3 \\ 5 & 10 & 12 & & R2 \\ 6 & 11 & & & C2 \end{array} \rightarrow \text{exit}$$

EXAMPLE 2. $\lambda = (3, 3, 3).$

$$\begin{array}{c|l} 6 & 4 & 8 & 1 & 4 & 7 & R3 & R2 & R3 \\ 3 & 1 & 9 & \rightarrow & 2 & 5 & 8 & C1 & C1 & C1 \\ 2 & 5 & 7 & & 3 & 6 & 9 & C2 & C1 & C1 \end{array}$$

Remark . Throughout IC, T remains a Pseudo Tableau; this fact will be used repeatedly in the proof of correctness.

3. DELETING A COLUMN FROM A TABLEAU

We now construct DC, the inverse of IC. At each stage, DC finds the last two entries exchanged by IC and exchanges them back to their original positions.

To motivate DC, we recall that if IC has just exchanged entries a_{kx} and a_{k-1y} between $\binom{k-1}{k}$, then

- (1) $d_k = Cy$ for some y , and
- (2) either
 - (a) $d_{k-1} = Ru$ for some u , or
 - (b) $d_{k-1} = Cv$ and (as we shall prove in Section 4) a_{kx} is in position $(k-1, x)$ with $v < x$.

We shall also prove that the following hold (see Figure 2):

- (1) $a_{k-1t} < a_{kt} \quad \forall t,$
- (2) $a_{k-1t} < a_{kt-1} \quad x < t \leq y,$
- (3) $a_{k-1x} > a_{kx-1}.$

This suggests the following algorithm.

Procedure DC (Delete Column)

Input: (T, S) , a PYT with m rows

Output: Column vector $\mathbf{a} = \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} a_i \in T \quad \forall i$

(T, S) a new, smaller PYT where $a_i \notin T \quad \forall i.$ (T, S) have the

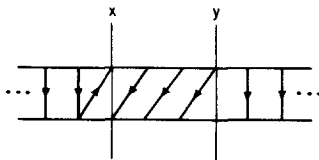


FIG. 2. Rows $\binom{k-1}{k}$ immediately after IC has exchanged a_{kx} and a_{k-1y} . Notation convention: there is a directed arrow from a to b if $a < b$.

same shape as the original pair except that their leftmost columns have been deleted.

$DC(T, S)$

(0) *Initialize:* For $j = 1$ to m do

$$d_j \leftarrow S_{j1}$$

endfor

[$\mathbf{d} \leftarrow$ leftmost column of S].

(1) *While* there is an entry a_{jy_j} in T satisfying:

(i) $d_j = Cy_j$

(ii) $d_{j-1} = Ru$ for some u or

$$d_{j-1} = Cv \text{ for some } v < y_j$$

where $a_{jt-1} < a_{j-1t}$

for some $v < t \leq y_j$

[Rows $j-1$ and j could have exchanged entries during IC].

do begin:

(2) For each such j define s_j :

If $a_{j-1t} < a_{jt-1}$ for $2 \leq t \leq y_j$

or $d_j = C1$

Then $s_j \leftarrow 1$.

Else let r be the largest column index such that

$r \leq y_j$ and $a_{jr-1} < a_{j-1r}$;

$s_j \leftarrow r$.

[If $\binom{j-1}{j}$ were the last rows to exchange entries during IC then a_{j-1s_j} is the new entry in row $j-1$].

(3) $a_{k-1s} \leftarrow \max\{a_{j-1s_j} \mid a_{jy_j} \text{ satisfies (i) and (ii) above}\}$

$$a_{ky} \leftarrow a_{ky_k}$$

[These were the last entries exchanged by IC]

(4) *EXCHANGE* a_{k-1s} and a_{ky} .

Let $(k-1, y')$ be the new position of a_{ky}

(5) [Update Pointers]

$$d_{k-1} \leftarrow \begin{cases} Ru - 1 & \text{if } d_{k-1} = Ru, u > 2 \\ Cs & \text{if } d_{k-1} = R2 \\ Cv & \text{if } d_{k-1} = Cv \end{cases}$$

$$d_{k-1} \leftarrow Cy'$$

endwhile

(6) [T now looks like the pseudo tableau obtained by initializing IC:

$d_j = Cy_j \forall j$, and deleting a_{jy_j} from T for all j leaves an SYT].

For $1 \leq j \leq m$ $a_j \leftarrow a_{jy_j}$, where $d_j = Cy_j$

$$\mathbf{a} \leftarrow \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}$$

$T \leftarrow T$ with $\{a_1, \dots, a_m\}$ deleted

$$S \leftarrow S \text{ with } \begin{pmatrix} S_{11} \\ S_{21} \\ \vdots \\ S_{m1} \end{pmatrix} \text{ deleted.}$$

exit \square

Again we note that T remains a Pseudo Tableau during the execution of DC.

EXAMPLE. Application of FIND-R to Pointed Tableau of Example 1, Section 2.

(T)	(S)	(R)
$\begin{array}{cccc} 1 & 3 & 4 & 8 \\ 2 & 7 & 9 & \\ 5 & 10 & 12 & \\ 6 & 11 & & \end{array}$	$\begin{array}{ccc} C1 & R3 & C2 \\ C3 & C1 & C1 \\ R2 & C2 & C1 \\ C2 & C1 & \end{array}$	$\begin{array}{cccc} 9 & 12 & 8 & 1 \\ 2 & 5 & 4 & \\ 11 & 3 & 7 & \\ 10 & 6 & & \end{array}$

[1]	$\begin{array}{cccc} 1 & 3 & 4 & 8 \\ 2 & 7 & 9 & \\ 5 & 10 & 12 & \\ 6 & 11 & & \end{array}$	$\begin{array}{c} C1 \\ C3 \\ R2 \\ C2 \end{array}$	\rightarrow	$\begin{array}{cccc} 1 & 3 & 4 & 8 \\ 2 & 7 & 9 & \\ 5 & 11 & 12 & \\ 6 & 10 & & \end{array}$	$\begin{array}{c} C1 \\ C3 \\ C2 \\ C2 \end{array}$
	(a)			(b)	

(R)	(T)
$\begin{array}{cccc} 1 & 4 & 8 & 9 \\ 2 & 3 & 7 & \\ 5 & 11 & 12 & \\ 6 & 10 & & \end{array}$	$\begin{array}{ccc} C4 & 9 & \\ C1 & 2 & \\ C2 & 11 & \\ C2 & 10 & \end{array}$

\rightarrow	$\begin{array}{ccc} 1 & 4 & 8 \\ 3 & 7 & \\ 5 & 12 & \\ 6 & & \end{array}$	\rightarrow exit
---------------	--	--------------------

(a) There are two eligible pairs of rows:

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix}: a_{1s_2} = 3, a_{23} = 9, s_2 = 2, d_1 = C1,$$

$$\begin{pmatrix} 3 \\ 4 \end{pmatrix}: a_{3s_4} = 10, a_{42} = 11,$$

$10 > 3$ so 10 and 11 are exchanged.

$d_3 = R2$, 10 is in cell (4, 2), so

$$d_4 \leftarrow C2, d_3 \leftarrow C2.$$

(b) $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ is the only eligible pair. $s_2 = 2 = s$. $d_1 = C1$ so $d_2 \leftarrow C1$,
 $d_1 \leftarrow C4$ since 9 is in cell (1, 4).

$$[2] \quad \begin{array}{ccc|c} 1 & 4 & 8 & R3 \\ 3 & 7 & & C1 \\ 5 & 12 & & C2 \\ 6 & & & C1 \end{array} \rightarrow \begin{array}{ccc|c} 1 & 4 & 8 & R3 \\ 3 & 12 & & C2 \\ 5 & 7 & & C1 \\ 6 & & & C1 \end{array}$$

$$\rightarrow \begin{array}{ccc|c} 1 & 8 & 12 & C3 \\ 3 & 4 & & R2 \\ 5 & 7 & & C1 \\ 6 & & & C1 \end{array} \rightarrow \begin{array}{ccc|c} 1 & 8 & 12 & C3 \\ 4 & 5 & & C2 \\ 3 & 7 & & C1 \\ 6 & & & C1 \end{array}$$

$$\rightarrow \begin{array}{c|c} 12 & 1 \ 8 \\ 5 & 4 \\ 3 & 7 \\ 6 & \end{array} \rightarrow \text{exit}$$

$$[3] \quad \begin{array}{c|c} 1 \ 8 & C2 \\ 4 & C1 \\ 7 & C1 \end{array} \rightarrow \begin{array}{c|c} 8 & 1 \\ 4 & \\ 7 & \end{array} \rightarrow \text{exit}$$

$$[4] \quad 1 | C1 \rightarrow 1 \rightarrow \text{exit}$$

4. PROOF OF EFFECTIVENESS FOR IC

We prove that IC is effective by induction on the number of exchanges made. It is sufficient to show: (1) The algorithm does not abort at step 3, i.e., if a_{kx} is chosen in step 2, then $d_{k-1} = Cy$ for some y ; (2) $a_{kx}^{(1)}, a_{kx}^{(2)}, \dots$, the sequence of entries chosen to exchange upward in step 2, is nondecreasing; this ensures that the algorithm will halt, since the number of times an entry can move upward is bounded by the number of rows in the Tableau; (3) at all times during IC, the d_j point to existing hook cells. Step (3) is immediately apparent, so we only prove (1) and (2).

To facilitate the proof we introduce a bookkeeping vector \mathbf{e} . We shall use the notation $e_j^{(i)}$ ($i \geq 0$) to denote the value of e_j after the i th iteration. When the context is clear we use e_j to denote $e_j^{(n)}$ or e_j and e'_j to denote $e_j^{(n-1)}, e_j^{(n)}$. We use a similar convention for $a_{ju}^{(i)}$ and $d_j^{(i)}$.

DEFINITION 4.1. For $i \geq 0$.

$e_j^{(i)} =$ The former column index of the last entry to move into row j , if row j has exchanged with $j - 1$ or $j + 1$;

$= z$, where $d_j^{(i)} = Cz, 3$ otherwise.

In effect, \mathbf{e} stores the information lost during an exchange.

EXAMPLE.

$$\mathbf{a} = \begin{pmatrix} 7 \\ 2 \\ 6 \end{pmatrix} T = \begin{pmatrix} 1 & 5 \\ 3 & 8 \\ 4 \end{pmatrix}$$

We insert \mathbf{a} using IC:

$$\begin{array}{ccc|c|c} & T & & \mathbf{d} & \mathbf{e} & & & & & & & \\ 1 & 5 & 7 & C3 & 3 & \rightarrow & 1 & 3 & 5 & C1 & 2 \\ 2 & 3 & 8 & C1 & 1 & & 2 & 7 & 8 & C2 & 3 \\ 4 & 6 & & C2 & 2 & & 4 & 6 & & C2 & 2 \\ & & & & & & & & & & & \\ & & & & & & 1 & 3 & 5 & C1 & 2 \\ & & & & & & \rightarrow & 2 & 6 & 8 & R2 & 2 \\ & & & & & & & 4 & 7 & C2 & 2 \end{array}$$

THEOREM 1. For $n \geq 0$, the following hold after the n th exchange of IC:

H1. Let a_{kx} and a_{k-1y} be the entries exchanged during the n th exchange, so that $d_k = d_k^{(n)} = Cz$ for some z .

Then $x = e_{k-1} \leq z \leq e_k = y$ after the exchange.

H2. For each pair of rows $\binom{j-1}{j}$

(A) If $d_{j-1} = Ru$ then $e_{j-1} \leq e_j$.

(B) If $d_{j-1} = Cv$ then $v \leq e_{j-1}$.

(C) If $d_{j-1} = Ru$

or $d_{j-1} = Cv$ and $v < e_j$,

then $a_{j-1t} < a_{jt} \forall t$;

if $e_{j-1} < e_j$,

then also $a_{j-1t} < a_{jt-1}$, $e_{j-1} < t \leq e_j$.

(D) Else, if $d_{j-1} = Cv, e_j \leq v$,

then $a_{j-1t} < a_{jt+1}$, $e_j \leq t \leq v$,

$a_{j-1t} < a_{jt}$ otherwise.

[Observe that H2 implies that if $a_{jt} < a_{j-1t}$, then $d_{j-1} = Cv$.]

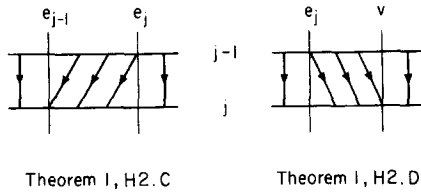


FIGURE 3

H3. For each pair $\binom{j-1}{j}$, if there is a t such that $a_{jt} < a_{j-1t}$, then $a_{kx} \leq a_{jt}$.
 [H2 and H3 are sufficient to prove (1) and (2) above.]

Proof of Theorem 1. For $n = 0$ (after initialization), H1 and H3 hold trivially. Since $d_j = Cv_j$ and $e_j = v_j$ for all j , H2.A and H2.B also hold. H2.C and H2.D hold because T was a Standard Young Tableau before a was inserted (see Figure 4).

Now assume H1, H2, and H3 hold through the $n - 1$ st exchange, and let a_{kx} be the entry chosen in step 2 of IC (we assume T is not yet an SYT).

We first verify that H1 holds after the n th exchange. By induction (H2), $d_{k-1} = Cy$ for some y , and $e_k \leq y \leq e_{k-1}$. Let a_{kz} be the largest entry in row k such that $a_{kz} < a_{k-1y}$. It is clear from Fig. 3 and the definition of a_{kx} , that $e_k \leq x \leq z \leq y \leq e_{k-1}$. One can also see that the new positions of a_{kx} and a_{k-1y} are $(k - 1, x)$ and (k, z) respectively. H1 follows from the definition of $e^{(n)}$: $e_{k-1}^{(n)} = x, e_k^{(n)} = y$.

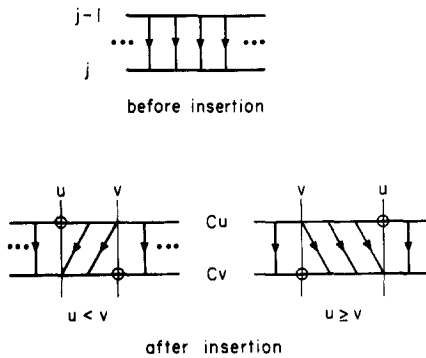


FIG. 4. Rows $\binom{j-1}{j}$ before and after the first iteration of IC.

To verify H3 we use the following lemma, which is an immediate consequence of the definition of a_{kx} and that fact that T is a Pseudo Tableau (see Figure 5).

LEMMA 4.1. *Before a_{kx} and a_{k-1y} are exchanged*

$$a_{k-2t} < a_{k-1t} < a_{kt} < a_{k+1t} \quad \text{for } t < x,$$

$$a_{kx} < a_{k+1x}.$$
□

Since $x \leq z \leq y$, exchanging a_{kx} and a_{k-1y} shifts the entries $a_{kx+1}, a_{kx+2}, \dots, a_{kz}$ one cell to the left, and shifts the entries $a_{k-1x}, a_{k-1x+1}, \dots, a_{k-1y-1}$ one cell to the right. No other entries are moved. Thus, if $j = k - 1$ or $k + 1$, and $a_{jt} < a_{j-1t}$ after the exchange, $x \leq t$ by Lemma 4.1, and so $a_{kx} \leq a_{jt}$. If $j \neq k - 1$ or $k + 1$, $a_{kx} < a_{jt}$ by the choice of a_{kx} . So H3 holds after the n th exchange.

This leaves us with H2 to verify. If $\binom{j-1}{j} \neq \binom{k-2}{k-1}, \binom{k-1}{k}, \binom{k}{k+1}$ then H2 holds by induction, since rows $\binom{j-1}{j}$ were not affected by the exchange.

Rows $\binom{k-1}{k}$

H2.A follows from H1. If $d'_{k-1} = Cv$ after exchange, then $d_k = Cv$ ($v \neq x$) before exchange. By induction (H2.B), $v \leq e_k$. But $e_k \leq x$ (shown in verifying H1), so $v < x = e'_{k-1}$, which verifies H2.B. An examination of Fig. 6 shows that H2.C holds for $\binom{k-1}{k}$.

Rows $\binom{k}{k+1}$

$d'_k = Cz$ with $z \leq y = e'_k$, by H1, so H2.A is vacuous, and H2.B holds in $\binom{k}{k+1}$. To show that H2.C and H2.D hold, we note that there are three possibilities: (a) $x \leq z < e_{k+1}$, (b) $x < e_{k+1} \leq z$, or (c) $e_{k+1} \leq x \leq z$. In (a) and (b) we have $e_k < e_{k+1}$ before exchange, since $e_k \leq x$. So $a_{kt} < a_{k+1t-1}$ for $x < t \leq e_{k+1}$ by induction (H2.C). Since only entries a_{kx+1}, \dots, a_{kz} are shifted left, if $z < e_{k+1}$ then we still have $a_{kt} < a_{k+1t-1}$ for $z < t \leq e_{k+1}$ and $a_{kt} < a_{k+1t}$ for all other t , so H2.C holds. If $e_{k+1} \leq z$, then, at worst, $a_{kt} < a_{k+1t+1}$ for $e_{k+1} \leq t \leq z$, $a_{kt} < a_{k+1t}$, otherwise, so H2.D holds. In (c) before the exchange, $a_{kt} < a_{k+1t}$ for all t . If not, then $a_{kt} > a_{k+1t}$ implies $t \leq x$ by H1 (recall that we would have $d_k = Cu$ and $u \leq x$), but this violates Lemma 4.1. Thus, H2.D still holds in $\binom{k}{k+1}$.

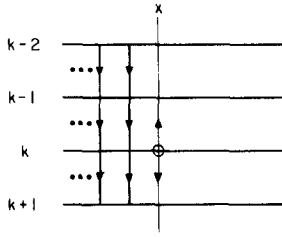


FIG. 5. Lemma 4.1.

Rows $\binom{k-2}{k-1}$

We need two additional lemmas.

LEMMA 4.2. Suppose the newest entry in row $k - 2$ was exchanged up from row $k - 1$. Then $e_{k-2} \leq x$.

Proof. Let $w = e_{k-2}$, and let $b = a_{k-2w}$ be the entry exchanged up from $k - 1$. Figure 6 shows that $a_{k-1w-1} < b$ just after $\binom{k-2}{k-1}$ exchange. Let c be the next entry exchanged up from k into row $k - 1$. By H3, $c > b$, so $c > a_{k-1w-1}$. Thus c must be exchanged into position $(k - 1, x')$ with $w \leq x'$. Either $c = a_{kx}$, or a_{kx} is exchanged into $k - 1$ after c . Then, since $a_{kx} > c$ by H3, $w \leq x' < x$. \square

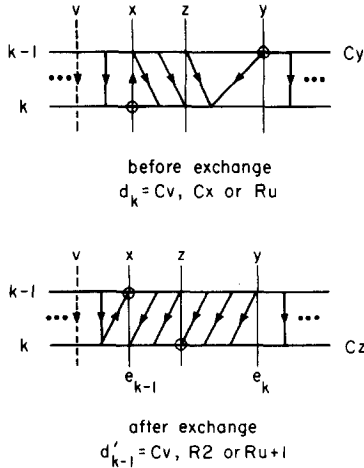


FIG. 6. Rows $\binom{k-1}{k}$ during the exchange of a_{kx} and $a_{k-1,y}$ into positions $(k - 1, x)$ and (k, z) by IC. Relabel the rows with $\binom{k-2}{k-1}$ to prove Lemma 4.2.

LEMMA 4.3. Just before $\binom{k-1}{k}$ exchange, assume that the newest entry in row $k-2$ moved down from $k-3$. (So $d_{k-2} = Cv$ for some v .) Then $\binom{k-1}{k}$ have not exchanged entries since the last time $\binom{k-3}{k-2}$ exchanged.

Proof. Suppose to the contrary that $\binom{k-1}{k}$ have exchanged. Let b be the last entry exchanged up into $k-1$, and assume b is in column x' . We have assumed that $\binom{k-2}{k-1}$ do not exchange subsequently, so $e_{k-1} = x'$. By H3, $a_{kx} > b$, so $e_{k-1} = x' < x$. But $x \leq y \leq e_{k-1}$ by H1 and H2.B, a contradiction. \square

COROLLARY. Under the assumption of Lemma 4.3, where $d_{k-2} = Cv$, if $v < e_{k-1}$ then $a_{k-2t} < a_{k-1t-1}$ for $v < t \leq e_{k-1}$. [Note that this is stronger than H2.C.] (See Figure 7.)

Proof. Suppose $\binom{k-3}{k-2}$ last exchanged on the i th iteration. Let $a_{k-2x}^{(i-1)}$ and $a_{k-3v}^{(i-1)}$ be the entries exchanged. Since $k-2$ does not exchange again, $d_{k-2} = d_{k-2}^{(i)} = Cv$. $k-1$ does not exchange after the $i-1$ st iteration, by Lemma 4.3, so $e_{k-1} = e_{k-1}^{(i)}$. We have $e_{k-2}^{(i-1)} \leq x' \leq v < e_{k-1}^{(i-1)}$ by H2.D (and by assumption $v < e_{k-1}$). Thus $a_{k-2t}^{(i-1)} < a_{k-1t-1}^{(i-1)}$ for $e_{k-2}^{(i-1)} < t \leq e_{k-1}^{(i-1)}$ by H2.C. Then, after $\binom{k-3}{k-2}$ exchange, we have $a_{k-2t}^{(i)} = a_{k-2t} < a_{k-1t-1} = a_{k-1t-1}^{(i)}$ for $v < t \leq e_{k-1}$. \square

We can now verify H2 for $\binom{k-2}{k-1}$. If $d_{k-2} = Cv$, then H2.B holds by induction (d_{k-2} was not affected by the n th exchange). If $d_{k-2} = Ru$ then

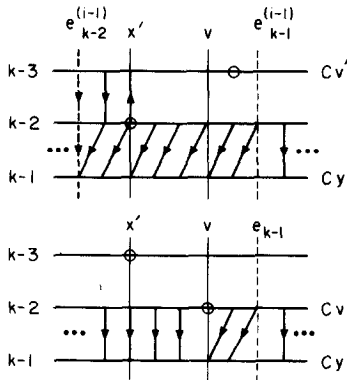


FIG. 7. The effect of the last exchange between rows $\binom{k-3}{k-2}$ on rows $\binom{k-2}{k-1}$ as described in the corollary to Lemma 4.3.

the newest entry in row $k - 2$ must have moved up from row $k - 1$, so Lemma 4.2 shows that $e_{k-2} \leq x = e_{k-1}$ after exchange, so H2.A holds.

There are three cases to consider for H2.C and H2.D: (a) $d_{k-2} = Ru$, (b) $d_{k-2} = Cv$, $v < e_{k-1}$, and (c) $d_{k-2} = Cv$, $e_{k-1} \leq v$. In all three cases, $d_{k-1} = Cy$, $y \leq e_{k-1}$, and $x = e'_{k-1}$.

In (a), Lemma 4.2 applies, and $e_{k-2} \leq x \leq y \leq e_{k-1}$. By induction (H2.C) $a_{k-2t} < a_{k-1t-1}$ for $e_{k-2} < t \leq e_{k-1}$, $a_{k-2t} < a_{k-1t}$ otherwise. Recall that the exchange only shifts $a_{k-1x}, \dots, a_{k-1y-1}$ one cell to the right. So, if $e_{k-2} < x$ then $a_{k-2t} < a_{k-1t-1}$ for $e_{k-2} < t \leq x = e'_{k-1}$, $a_{k-2t} < a_{k-1t}$ otherwise, so H2.C still holds. If $e_{k-2} = x$, then if b is in position $(k - 2, x)$, b was exchanged upward by definition of e_{k-2} . So $b < a_{kx}$ by H3. Then, after exchange, $a_{k-2t} < a_{k-1t}$ for all t , and H2.C holds.

In case (b), ($d_{k-2} = Cv$, $v < e_{k-1}$), first suppose that $k - 2$ last exchanged with $k - 3$. Then $a_{k-2t} < a_{k-1t-1}$ for $v < t \leq e_{k-1}$, $a_{k-2t} < a_{k-1t}$ otherwise, by the corollary to Lemma 4.3. If $x \leq v$ then, since $y \leq e_{k-1}$, $a_{k-2t} < a_{k-1t+1}$ for $x \leq t \leq v$, $a_{k-2t} < a_{k-1t}$ otherwise ($x = e'_{k-1}$, so H2.D holds). If $v < x$ then $\binom{k-2}{k-1}$ are in order after the exchange, and since $v \leq e_{k-2}$, if $e_{k-2} < e'_{k-1} = x$ then H2.C holds. If $k - 2$ has never exchanged entries, then $e_{k-2} = v$ and the analysis is similar. Otherwise, row $k - 2$ exchanged last with row $k - 1$ and so $e_{k-2} \leq x$ by Lemma 4.2, and the analysis is the same as in case (a). (Note: $v < e_{k-2}$.)

Finally, in case (c) $d_{k-2} = Cv$, and $e_{k-1} \leq v \leq e_{k-2}$. H2.D applies, and since $y \leq e_{k-1}$, after exchange we have $a_{k-2t} < a_{k-1t+1}$ for $e'_{k-1} = x \leq t \leq v$ and $a_{k-2t} < a_{k-1t}$ otherwise, so H2.D still holds. \square

This completes the proof of Theorem 1, and shows that IC (and hence R-SORT) is effective.

5. PROOF OF EFFECTIVENESS FOR DC

The proof is similar to that in the previous section. It is sufficient to show (1) either there is an entry satisfying the conditions of Step 1 in DC, or if all the a_j are deleted from T' , then the result is a Standard Young Tableau; (2) $a_{k-1s}^{(1)}, a_{k-1s}^{(2)}, \dots$, the sequence of entries exchanged downward is non-increasing; and (3) the pointers remain legitimate throughout DC.

We define another bookkeeping vector \mathbf{f} , similar to \mathbf{e} . Again, $f_j^{(i)}$ denotes the value of f_j after the i th iteration.

DEFINITION 5.1. For all j and $i \geq 0$.

$f_j^{(i)}$ = The former column index of the newest entry in row j , if row j has exchanged with $j + 1$ or $j - 1$;
 = 0 otherwise

DEFINITION 5.2. Rows $\binom{j-1}{j}$ will be called *eligible* (for exchange) if they satisfy the conditions in Step 1 of DC.

We first state our Theorem 2 then show how (1), (2), and (3) follow from the theorem.

THEOREM 2. For $n \geq 0$ the following hold after the n th exchange of DC:

H1. Let a_{k-1s} and a_{ky} be the entries exchanged during the n th exchange, so $d_{k-1} = Cz$ after exchange, $d_k = Cy$ before exchange. Then,

$$s = f_k \leq f_{k-1} = y \leq z.$$

If $d_{k-1} = Cv$ before exchange then also $v < s$.

H2. For each pair of rows $\binom{j-1}{j}$

(A) If

$$d_{j-1} = Cu, d_j = Rv, \quad 0 < f_j \leq u,$$

Then $a_{j-1t} < a_{jt+1}, \quad f_j \leq t \leq u,$

$$a_{j-1t} < a_{jt}, \quad \text{otherwise.}$$

(B) If

$$d_{j-1} = Cu, d_j = Cv, \quad v \leq u,$$

Then $a_{j-1t} < a_{jt+1}, \quad v \leq t \leq u,$

$$a_{j-1t} < a_{jt}, \quad \text{otherwise.}$$

(C) Otherwise

$$a_{j-1t} < a_{jt} \quad \text{for all } t.$$

H3. If $\binom{j-1}{j}$ are eligible after the exchange then

$$a_{j-1s}^{(n)} \leq a_{k-1s} = a_{k-1s}^{(n-1)}.$$

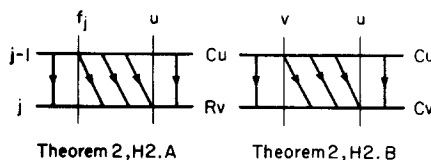


FIGURE 8

It is clear that (2) follows from H3, and (3) follows from H1 ($v < s$) and the pointer updating rules. The next lemma will be used to show that (1) follows from H2. The proof is immediate.

LEMMA 5.1. *Let $\binom{j-1}{j}$ be two rows in a Pseudo Tableau. Deleting a_{j-1x} and a_{jy} from $\binom{j-1}{j}$ leaves $\binom{j-1}{j}$ in order ($a'_{j-1t} < a'_{jt} \forall t$) iff*

(a) $x < y$ and

$$\begin{cases} a_{j-1t} < a_{jt-1}, & x < t \leq y, \\ a_{j-1t} < a_{jt} & \text{otherwise,} \end{cases}$$

or

(b) $x \geq y$ and

$$\begin{cases} a_{j-1t} < a_{jt+1}, & y \leq t \leq x \\ a_{j-1t} < a_{jt} & \text{otherwise.} \end{cases} \quad \square$$

EXAMPLE.

$$\binom{j-1}{j} = \begin{matrix} 1 & 3 & 6 & 9 \\ 4 & 5 & 7 & \end{matrix}$$

4 and 6 can be deleted by (b) $\rightarrow \begin{matrix} 1 & 3 & 9 \\ 5 & 7 & \end{matrix}$.

3 and 7 cannot be deleted $\rightarrow \begin{matrix} 1 & 6 & 9 \\ 4 & 5 & \end{matrix}$.

We now claim that (1) follows from H2. Suppose there is no a_{jy} satisfying Step 1 of DC. Then $d_j \neq Ru$ for any j (otherwise there would exist $\binom{j-1}{j}$ with $d_{j-1} = Ru, d_j = Cv$). Let $j \geq 2$, so $d_{j-1} = Cu, d_j = Cv$ for some u and v . If $u \geq v$, then a_{j-1u} and a_{jv} can be deleted by H2.B and Lemma 5.1(b). If $u < v$ then $a_{j-1t} < a_{jt-1}$ for $u < t \leq v$ otherwise $\binom{j-1}{j}$ would be eligible. So a_{ju} and a_{jv} can be deleted by Lemma 5.1(a).

Proof of Theorem 2. For $n = 0$, H1 and H3 are vacuous. T is initially an SYT, so $\binom{j-1}{j}$ are in order for all j , and H2 holds.

Assume H1–H3 hold through the $n - 1$ st exchange and assume there is a pair of eligible rows. Let a_{k-1s} and a_{ky} be the entries chosen in Step 3.

We first verify H1. From Step 1 of DC, if $d_{k-1} = Cu$, then $u < s \leq y$. Let z be the largest column index such that $a_{k-1z} < a_{ky}$. Since $\binom{k-1}{k}$ are in order by H2.C, $y \leq z$. Also, by the definition of $s, a_{k-1t} < a_{kt-1}$ for $s < t \leq y$, and $a_{k-1s} > a_{ks-1}$. Thus, the new positions of a_{k-1s} and a_{ky} are (k, s) and $(k - 1, z)$, respectively. After exchange, $f_k = s, f_{k-1} = y$ by definition, so H1 holds.

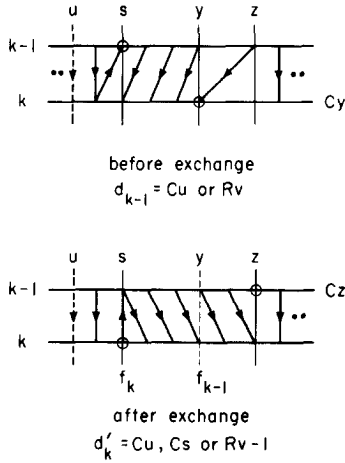


FIG. 9. Rows $\binom{k-1}{k}$ during the exchange of a_{k-1s} and a_{ky} into positions (k, s) and $(k-1, z)$ by DC.

The next lemma, which is analogous to Lemma 4.1, will help establish H2 and H3.

LEMMA 5.2. *Before (and hence after) a_{k-1s} and a_{ky} are exchanged:*

- (1) $a_{k-2t} < a_{k-1t} \quad t < s.$
- (2) $a_{kt} < a_{k+1t}$

Proof. The result is certainly true if $\binom{k-2}{k-1}$ and $\binom{k}{k+1}$ satisfy the conditions of H2.C so we only consider the other cases.

Rows $\binom{k}{k+1}$

Case 1. $d_{k+1} = Rv, 0 < f_{k+1} < s.$

(If $s \leq f_{k+1}$ then (2) holds by H2.A.) The newest entry in row $k+1, b$, came from row k and is in position $(k+1, f_{k+1})$. By H2.A, $a_{kt} < a_{k+1t}$ for $t < f_{k+1}$. By definition of s , $a_{ks-1} < a_{k-1s}$, and $a_{k-1s} < b$ by H3. So $a_{kt} < b$ for $f_{k+1} \leq t \leq s-1$ and (2) holds.

Case 2. $d_{k+1} = Cv, v < s \leq y.$

Observe that if any entries in $\binom{k}{k+1}$ are out of order, then $\binom{k}{k+1}$ must have exchanged at least once, because, $\binom{k}{k+1}$ are in order at $n = 0$, and (by H1) exchanging entries in $\binom{k-1}{k}$ or $\binom{k+1}{k+2}$ only causes entries in row k to shift right or entries in row $k+1$ to shift left.

Let c be the entry which moved down the last time $\binom{k}{k+1}$ exchanged. First, suppose that $\binom{k+1}{k+2}$ do not exchange subsequently, so c is in position $(k+1, f_{k+1})$. Note that $a_{kt} < a_{k+1t}$ for $t < f_{k+1}$ (Fig. 9) immediately after c moves down, and this is still true after any exchange of $\binom{k-1}{k}$. For $f_{k+1} \leq t < s$, $a_{kt} < a_{k+1t}$ holds by the argument in Case 1.

Now we assume $\binom{k+1}{k+2}$ do exchange. Let d be the next entry to move into row $k+1$. After this exchange, $d_{k+1} = Cw$ and d is in position $(k+1, w)$. Observe that $\binom{k+1}{k+2}$ cannot exchange a second time. Otherwise, let c' be the entry moving down into $k+2$, in column s' . Then, $w < s'$ by H1, and $c' > d$, which violates H3, hence $w = v$. By H2.B, $a_{kt} < a_{k+1t}$ for $t < v$. The same argument in Case 1 shows that since $a_{k-1s} < d$ (H3), $a_{kt} < a_{k+1t}$ for $v \leq t \leq s-1$, and (2) still holds.

Rows $\binom{k-2}{k-1}$

Again $\binom{k-2}{k-1}$ have exchanged at least once or they are in order. The argument in Case 2 above shows that $\binom{k-1}{k}$ have not exchanged after the last exchange of $\binom{k-2}{k-1}$, otherwise they would not be eligible now. Let c be the last entry exchanged down into $k-1$, so c is in cell $(k-1, f_{k-1})$. From Figure 9, $a_{k-2t} < a_{k-1t}$ for $t < f_{k-1}$ just after the exchange, and this is still true if $\binom{k-3}{k-2}$ exchange. $a_{k-1s} \leq c$ by H3, so $s \leq f_{k-1}$ and (1) holds. \square

We now verify H2. It is necessary only to consider rows $\binom{k-2}{k-1}, \binom{k-1}{k}, \binom{k}{k+1}$. Figure 9 shows that H2.A or H2.B holds in $\binom{k-1}{k}$.

Rows $\binom{k-2}{k-1}$

If H2.C holds before exchange, then it also holds afterward. So we assume $d_{k-2} = Cv$, and, before exchange, either (a) $d_{k-1} = Cu, (u < s)$ and $u \leq v$ or (b) $d_{k-1} = Ru$ and $0 < f_{k-1} \leq v$. In (a) we have (after exchange) $a_{k-2t} < a_{k-1t}$ for $t < s$ by Lemma 5.2, and for $s \leq t < z$ by H2.B and H1. If $v < z$ then H2.C holds; if $z \leq v$ then H2.B holds (recall $d'_{k-1} = Cz$).

In (b) we showed that $s \leq f_{k-1}$ in the proof of Lemma 5.2, and $a_{k-2t} < a_{k-1t}$ for $t < f_{k-1}$. Again, if $v < z$ then H2.C holds (since $a_{k-2t} < a_{k-1t}$ for $v < t$ before exchange); if $z \leq v$ then H2.B holds.

Rows $\binom{k}{k+1}$

$d_k = Cy$ before exchange. We again assume (a) $d_{k+1} = Cv, v \leq y$ or (b) $d_{k+1} = Rv, 0 < f_{k+1} \leq y$. If $s < f_{k+1}$ in (b) or $s < v$ in (a), then $\binom{k}{k+1}$ will be in order after the exchange, so we assume $f_{k+1} \leq s$ (or $v \leq s$).

Then $a'_{kt} < a'_{k+1t}$ (after exchange) for $t < s$, by Lemma 5.2, and for $t > s$ by H2.A or B.

For $t = s$ we must consider the location of the newest entry in row $k + 1$. In (b), this entry moved down from k and is in cell $(k + 1, f_{k+1})$. Thus $a_{k-1s} = a'_{ks} < a'_{k+1f_{k+1}} \leq a'_{k+1s}$ by H3. In (a), if $f_{k+1} = 0$ then $\binom{k}{k+1}$ are in order before and after exchange, so assume $f_{k+1} > 0$. If the newest entry in $k + 1$ came from k , the analysis is the same as in (b). Otherwise, the newest entry moved up from $k + 2$ and is in cell $(k + 1, v)$. The analysis is again the same, with v substituted for f_{k+1} . Thus H2 holds for all rows after exchange.

Finally, we consider H3. Again we need only consider $\binom{k-2}{k-1}$ and $\binom{k}{k+1}$ by the definition of a_{k-1s} .

Rows $\binom{k}{k+1}$

It is sufficient to show that if $\binom{k}{k+1}$ become eligible, then $s_{k+1} \leq s$. Assume $\binom{k}{k+1}$ eligible, so that $d'_{k+1} = d_{k+1} = Cv$ and $d'_k = Ru$ or $d'_k = Cu$ ($u \leq s$) and $u < v$. Assume $s < v$ otherwise $s_{k+1} \leq v \leq s$ by definition of s_{k+1} . By H2.B, $a_{kt} < a_{k+1t}$ for $t < v$ ($d_k = Cy$), so $a'_{kt} < a'_{k+1t-1}$ for $s < t \leq v$ after exchange and $s_{k+1} \leq s$.

Rows $\binom{k-2}{k-1}$

Recall that $d'_{k-1} = Cz$ and that $a_{k-1s}, \dots, a_{k-1z-1}$ are shifted left. Assume $\binom{k-2}{k-1}$ eligible. If $d'_{k-2} = d_{k-2} = Rv$ for some v , then $\binom{k-2}{k-1}$ are in order before exchange by H2.C. Thus $a'_{k-2t} < a'_{k-1t-1}$ for $s < t \leq z$ so $s_{k-1} \leq s$ and so $a'_{k-2s_{k-1}} < a_{k-1s}$.

Else, $d'_{k-2} = Cv$ with $v < z$. In verifying H2, we showed that $\binom{k-2}{k-1}$ must be in order after exchange (H2.C). By H2.A or H2.B, we have $a_{k-2t} < a_{k-1t}$ for $v < t$ before the exchange, so $a'_{k-2t} < a'_{k-1t-1}$ for $\max(v, s) < t \leq z$, after exchange. Then, if $s \leq v$, $\binom{k-2}{k-1}$ are not eligible, and if $v < s$ then $s_{k-1} \leq s$. \square

Theorem 2 is now established, and we conclude that FIND-R is effective.

6. PROOF THAT IC AND DC ARE INVERSES

The proofs will follow easily by induction from the following two lemmas.

LEMMA 6.1. *Let $(T, \mathbf{d})^{(m)}$ be a Pseudo Tableau and pointer vector obtained by $m \geq 1$ iterations of IC. If a_{kx} and a_{k-1y} were chosen to exchange in*

$(T, \mathbf{d})^{(m-1)}$ then $a'_{k-1x} = a_{kx}$ is the largest entry in the set defined in Step 3 of DC.

Proof. $d'_k = Cz$ and Fig. 6 shows that a'_{kz} satisfies (i) and (ii) of Step 1 of DC, with $s_k = x$, so a'_{k-1x} is a member of this set.

We first consider the cases in which $\binom{k}{k+1}$ or $\binom{k-2}{k-1}$ are eligible in $(T, \mathbf{d})^{(m)}$. Assume $d_{k+1} = Cv$ with $z < v$. Then $z < e_{k+1}$ by Theorem 1, H2.B. $e_k \leq x \leq z$ before exchange (H2.D), so $a_{kt} < a_{k+1t-1}$, $e_k < t \leq e_{k+1}$. After exchange, $a'_{kt} < a'_{k+1t-1}$ for $z < t \leq e_{k+1}$ so $\binom{k}{k+1}$ are not eligible.

If $d'_{k-1} = Cu$ after exchange then $u < x$ and $a'_{k-1u} < a'_{k-1x}$. If $\binom{k-2}{k-1}$ are eligible, then $s_{k-1} \leq u$ and so $a'_{k-2s_{k-1}} < a'_{k-1x}$.

Now assume $j \neq k-1, k, \text{ or } k+1$, and $\binom{j-1}{j}$ are eligible. Then $d'_j = Cv$ for some v . If $d'_{j-1} = Rw$ then b , the newest entry in $j-1$, came from j and is in $(j-1, e_{j-1})$. $v \leq e_j$, and either $v \leq e_{j-1}$ or $a_{j-1t} < a_{jt-1}$ for $e_{j-1} \leq t \leq v$, so $a_{j-1s_j} \leq b$ (Theorem 1, H2.C). But $b < a'_{k-1x}$ by H3.

Now assume $d'_{j-1} = Cw$ with $w < v$ ($v \leq e_j$). Theorem 1, H2.C applies; if $w = e_{j-1}$ then $\binom{j-1}{j}$ are not eligible, so $w < e_{j-1}$, and $j-1$ has exchanged at least once. Suppose the newest entry in $j-1$ came from row j , then the argument above (for $d'_{j-1} = Rw$) holds. Otherwise, b , the newest entry, came from $j-2$ and is in $(j-1, w)$. It follows from Lemma 4.3 that $\binom{j}{j+1}$ can exchange entries at most once after b moves down. If $\binom{j}{j+1}$ do not exchange, then the corollary to Lemma 4.3 shows that $a_{j-1t} < a_{jt-1}$ for $w < t \leq e_j$, so $\binom{j-1}{j}$ are not eligible. If $\binom{j}{j+1}$ do exchange, then c , the entry which moved up is in (j, e_j) . The exchange does not affect a_{jt} for $t < e_j$, so $\binom{j-1}{j}$ are still not eligible since $w < e_j$. \square

LEMMA 6.2. *Now let $(T, \mathbf{d})^{(n)}$ be a Pseudo Tableau and pointer vector obtained by $n \geq 1$ iterations of DC. If a_{k-1s} and a_{ky} were chosen to exchange in $(T, \mathbf{d})^{(n-1)}$ then $a'_{ks} = a_{k-1s}$ is the smallest entry out of order in $(T, \mathbf{d})^{(n)}$.*

Proof. From Figure 9, $a'_{ks} < a'_{k-1s}$, so a'_{ks} is out of order. From Lemma 5.2, $a'_{k-2t} < a'_{k-1t}$ for $t < s$. In verifying H2 (Theorem 2) we showed that $\binom{k}{k+1}$ are in order for all t .

If a'_{jr} is out of order, with $j \neq k-1, k, \text{ or } k+1$, then $\binom{j-1}{j}$ must have exchanged at least once. If b was the last entry exchanged down, then $b \leq a'_{jr}$, and $a'_{ks} < b$ by Theorem 2, H3. \square

THEOREM 3. *Let (T, S) be a PYT with l rows, and*

$$\mathbf{a} = \begin{pmatrix} a_1 \\ \vdots \\ a_l \end{pmatrix}$$

a column of positive integers not in T . Then

$$DC[IC(\mathbf{a}, T, S)] = (\mathbf{a}, T, S).$$

Proof. By Lemma 6.1, if DC is applied to $(T, S)^{(m)}$ with $m \geq 1$, then a'_{k-1x} and a'_{kz} are exchanged on the next iteration. From Fig. 6 it is easy to see that the pointers will be reversed correctly, and that the result will be $(T, S)^{(m-1)}$.

If $(T, S)^{(0)}$ is the Tableau obtained by initializing IC, then $DC[(T, S)^{(0)}] = (\mathbf{a}, T, S)$ since step 6 will be executed (see Fig. 4 and Lemma 5.1). The theorem follows by induction on m . \square

THEOREM 4. *Let (T, S) be a PYT. Then $IC[DC(T, S)] = (T, S)$.*

Proof. Figure 9 and Lemma 6.2 show that $IC[(T, S)^{(n)}] = (T, S)^{(n-1)}$ for $n \geq 1$. If $n = 0$, then $(T, S)^{(0)}$ is a PYT, so $IC[(T, S)^{(0)}] = (T, S)$ since Step 5 will be executed. \square

7. POINTER TABLEAUX AND INVERSION TABLES

Let σ be a permutation of $[n]$, and let $a_1 < a_2 < \dots < a_n$ be a sequence of positive integers. The *inversion table* for the induced permutation $a_{\sigma_1}, a_{\sigma_2}, \dots, a_{\sigma_n}$ is the sequence b_1, b_2, \dots, b_n , where, for all j , b_j is the number of a_{σ_i} which are to the left of a_j and greater than a_j . For example, the inversion table for 92758 is 12110.

It is easy to check that if we apply R-SORT to a single column, then the resulting pointer column, S , gives us the inversion table directly. However, we can make a few stronger statements.

We can also define another type of inversion table, which we shall call the *involuted inversion table*. This is the sequence C_1, C_2, \dots, C_n , where for each j , C_j is the number of entries to the right of a_{σ_j} which are smaller than a_{σ_j} (the entry in the j th position). The involuted inversion table for 92758 is 40100. [N.B. The name ‘‘involuted inversion table’’ comes from the fact that $\sigma^2 = 1$ iff $b_j = C_j$ for all j . (See Knuth [1], p. 19, Ex. 7 and 9)]

Suppose that R-SORT is applied to a Tableau $R = (a_{ij})$ whose columns are already arranged in increasing order. It is not hard to see that the effect of R-SORT is to sort within the rows, since, at each stage, only the insertion step of IC will be executed. Hence, the final Pointer Tableau S has the form:

$$\begin{matrix} Ck_{11} & Ck_{12} & \cdots & Ck_{1\lambda_1} \\ Ck_{21} & \cdots & Ck_{2\lambda_2} & \\ \vdots & & \vdots & \\ Ck_{m1} & \cdots & Ck_{m\lambda_m} & \end{matrix}$$

where $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$. The reader can verify that, for each $i, k_{i1} - 1, k_{i2} - 1, \dots, k_{i\lambda_i} - 1$, is the involuted inversion table for the permutation $a_{i1}a_{i2} \cdots a_{i\lambda_i}$ (the i th row of R).

EXAMPLE.

R	T	S	Involuted inversion tables
2 1 5 4	1 2 4 5	C2 C1 C2 C1	1 0 1 0
7 3 6	3 6 7	C3 C1 C1	2 0 0
10 9 8	8 9 10	C3 C2 C1	2 1 0

A more surprising result is that if R-SORT is applied to a Pseudo Tableau R (one whose rows are increasing), then the effect is to sort R within columns. Furthermore, the Pointer Tableau S has the form

$$\begin{matrix}
 Q^{l_{11}} Q^{l_{12}} \cdots Q^{l_{1\lambda_1}} \\
 \vdots \\
 Q^{l_{m1}} \cdots Q^{l_{m\lambda_m}}
 \end{matrix}$$

where $Q = C$ if $l_{ij} = 1$, and $Q = R$ otherwise; and for all j , the column $l_{1j} - 1, l_{2j} - 1, \dots$ is the inversion table for the permutation determined by the j th column of R .

EXAMPLE.

R	T	S	Inversion tables
5 7 8	1 3 6	R2 R3 R2	1 2 1
1 4 6	2 4 8	R2 R2 C1	1 1 0
2 3	5 7	C1 C1	0 0

This result is a consequence of the following, more general lemma.

LEMMA 7.1. *Let*

$$\begin{matrix}
 T = t_{11} & t_{12} & t_{13} \cdots \\
 & t_{21} & t_{22} & \cdots \\
 & \vdots & & \\
 & t_{m1} & t_{m2} & \cdots
 \end{matrix}$$

be a Standard Young Tableau. Let

$$\begin{pmatrix} a'_1 \\ \vdots \\ a'_l \end{pmatrix}$$

with $l \geq m$ be a column vector of distinct positive numbers not contained in T . Let

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_l \end{pmatrix}$$

be the same vector sorted in increasing order ($a_1 < a_2 < \dots < a_l$). If $a_1 < t_{11}, a_2 < t_{21}, \dots, a_m < t_{m1}$, then the SYT obtained by applying IC to

$$\begin{pmatrix} a'_1 \\ a'_2 \\ \vdots \\ a'_l \end{pmatrix}$$

and T is

$$\begin{array}{cccc} \mathbf{a}T = a_1 & t_{11} & t_{12} & \cdots \\ & a_2 & t_{21} & t_{22} \cdots \\ & \vdots & & \\ & a_m & t_{m1} & \cdots \\ & a_{m+1} & & \\ & \vdots & & \\ & a_l & & \end{array}$$

Furthermore, if \mathbf{d} is the resulting pointer vector then $d_j = C1$ or Ru for all j .

Proof. The following facts can be established by induction on exchanges (we leave the details to the reader):

After the n th exchange ($n \geq 0$), let j be the smallest index such that a_j is in row k but $k > j$ (if $k \leq j$ for all j set $j = l + 1$); if $j < l + 1$ then

(1) a_j is in position $(k, 1)$;

(2) a_j is the smallest entry out of order (i.e., the entry chosen in Step 2 of IC to exchange upward);

(3) Each row, i , contains exactly one entry from \mathbf{a} ; if $d_i = Cx$ then this entry is in cell (i, x) .

We now show that the lemma follows from (1), (2), and (3). Suppose $j = l + 1$ for the first time after the n th exchange. Then either $n = 0$ or two entries from \mathbf{a} were just exchanged, by (3), so each row i still contains exactly one entry from \mathbf{a} .

Hence, by the definition of j , a_i must be in row i for all i . If $n = 0$, it is clear that IC will halt with output $\mathbf{a}T$, and

$$\mathbf{d} = \begin{pmatrix} C1 \\ C1 \\ \vdots \\ C1 \end{pmatrix}$$

(since $a_i < t_{i1} \forall i$). If $n > 0$, then (2) and (3) guarantee that none of the t_{ij} have been moved from their original rows. Thus IC must halt with output $\mathbf{a}T$ in this case as well. By (3), \mathbf{d} has the desired form. By (2), IC cannot halt until $j = l + 1$. This establishes the lemma. \square

Our result on inversion tables follows from the observation that each a_j is exchanged upward once for each a_k above a_j in

$$\begin{pmatrix} a'_1 \\ \vdots \\ a'_l \end{pmatrix}$$

such that $k > j$, and that the pointer for the row containing a_j becomes $R2$ after a_j is exchanged for the first time. \square

ACKNOWLEDGMENTS

We are especially grateful to Herb Wilf for his encouragement. His critical reading of a large part of the manuscript helped clarify several ambiguities. We also wish to thank D. Sturtevant and J. Joichi for pointing out an error in the original version of IC.

One of us (D. Franzblau) would like to thank the National Science Foundation, which provided support during this work through the Graduate Fellowship Program.

Dale Worley also deserves special thanks for his diligent checking and proofreading of the manuscript.

REFERENCES

1. J. S. FRAME, G. DE ROBINSON, AND R. M. THRALL, The hook graphs of the symmetric group, *Canad. J. Math.* **6** (1954), 316–324.

2. C. GREENE, A. NIJENHUIS, AND H. S. WILF, A probabilistic proof of a formula for the number of Young tableaux of a given shape, *Advances in Math.* **31** (1979), 104–109.
3. A. P. HILLMAN AND R. M. GRASSL, Reverse plane partitions and tableau hook numbers, *J. Combinatorial Theory Ser. A* **21** (1976), 216–221.
4. D. E. KNUTH, “The Art of Computer Programming, Vol. 3. Sorting and Searching.” Addison–Wesley, Reading, Mass., 1973.
5. J. B. REMMEL, Bijective proofs of formulae for the number of Standard Young Tableaux, submitted.