Another involution principle-free bijective proof of Stanley's hook-content formula

C. Krattenthaler

Institut für Mathematik der Universität Wien,
Strudlhofgasse 4, A-1090 Wien, Austria.
e-mail: KRATT@Pap.Univie.Ac.At
WWW: http://radon.mat.univie.ac.at/People/kratt

Abstract. Another bijective proof of Stanley's hook-content formula for the generating function for semistandard tableaux of a given shape is given that does not involve the involution principle of Garsia and Milne. It is the result of a merge of the modified jeu de taquin idea from the author's previous bijective proof ("An involution principle-free bijective proof of Stanley's hook-content formula", Discrete Math. Theoret. Computer Science **3** (1998), 011–032) and the Novelli-Pak-Stoyanovskii bijection (Discrete Math. Theoret. Computer Science **1** (1997), 53–67) for the hook formula for standard Young tableaux of a given shape. This new algorithm can also be used as an algorithm for the random generation of tableaux of a given shape with bounded entries. An appropriate deformation of this algorithm gives an algorithm for the random generation of plane partitions inside a given box.

1. Introduction. There are two recent major contributions to bijective combinatorics in regard of bijective proofs of hook formulas for various types of tableaux: First, Novelli, Pak and Stoyanovskii [8] (announced in [9]) came up with a new bijective proof of the celebrated Frame–Robinson–Thrall hook formula [2] for the number of standard Young tableaux of a given shape, which is probably *the* bijective proof of the hook formula that everybody was looking for. This bijection can also be used as an algorithm for the random generation of standard Young tableaux of a given shape. Second, in [5] the author of this paper gave the first bijective proof of Stanley's hook-content formula [17, Theorem 15.3] for the generating function for semistandard tableaux of a given shape with bounded entries which does not rely on the involution principle of Garsia and Milne [3].

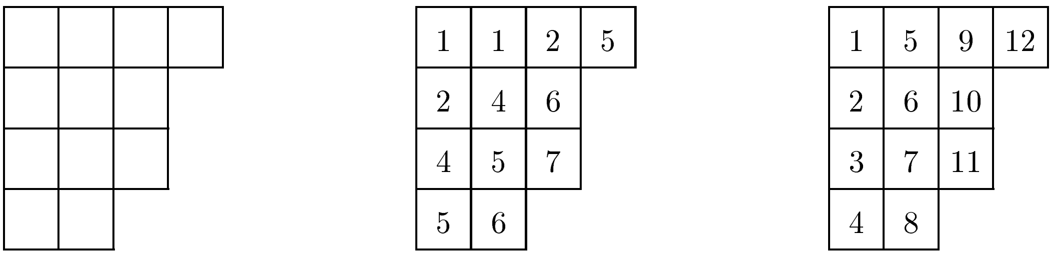
The purpose of this paper is to merge the ideas from the aforementioned bijections and thus obtain another bijective proof of Stanley's hook-content formula. Unlike the previous bijection, the new bijection can also be used as an algorithm for the random generation of semistandard tableaux of a given shape with bounded entries. A simple deformation of

1991 *Mathematics Subject Classification*. Primary 05A15; Secondary 05A17 05A30 05E10 05E15 11P81.

Key words and phrases. tableaux, plane partitions, hook-content formula, hook formula, bijection, jeu de taquin.

this algorithm in the rectangular shape case yields, as well, an algorithm for the random generation of plane partitions inside a given box.

In order to be able to state the formula, we have to recall some basic definitions. A *partition* is a sequence $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_r)$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$, for some r. The *Ferrers diagram* of λ is an array of cells with r left-justified rows and λ_i cells in row i. Figure 1.a shows the Ferrers diagram corresponding to $(4, 3, 3, 2)$. The *conjugate of* λ is the partition $(\lambda'_1, \ldots, \lambda'_{\lambda_1})$ where λ'_j is the length of the j-th column in the Ferrers diagram of λ. We label the cell in the i-th row and j-th column of (the Ferrers diagram of) λ by the pair (i, j). Also, if we write $\rho \in \lambda$ we mean 'ρ is a cell of λ'. The *hook length* h_ρ of a cell $\rho = (i, j)$ of λ is $(\lambda_i - j) + (\lambda'_j - i) + 1$, the number of cells in the *hook* of ρ, which is the set of cells that are either in the same row as ρ and to the right of ρ, or in the same column as ρ and below ρ, ρ included. The hook of a cell $\rho = (i, j)$ consists of three parts, the cell itself, the *arm*, which is the set of cells to the right of ρ in the same row as ρ, and the *leg*, which is the set of cells below ρ in the same column as ρ. The number of cells in the arm of ρ (which is $\lambda_i - j$) is denoted by a_ρ, and the number of cells in the leg of ρ (which is $\lambda'_j - i$) is denoted by l_ρ. The *content* c_ρ of a cell $\rho = (i, j)$ of λ is $j - i$.



a. Ferrers diagram b. semistandard tableau c. order of the cells

Figure 1

Given a partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_r)$, a *semistandard tableau* (*tableau* for short) *of shape* λ is a filling T of the cells of λ with integers such that the entries along rows are weakly increasing and such that the entries along columns are stricly increasing. Figure 1.b displays a (semistandard) tableau of shape $(4, 3, 3, 2)$. We write T_ρ for the entry in cell ρ of T. We call the sum of all the entries of a tableau T the *norm* of T, and denote it by $n(T)$.

Now we are in the position to state Stanley's hook-content formula [17, Theorem 15.3].

Theorem 1 (Stanley). *Let $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_r)$ be a partition and b be an integer $\geq r$. The generating function $\sum q^{n(T)}$, where the sum is over all semistandard tableaux T of shape λ with entries between 1 and b, is given by*

$$q^{\sum_{i=1}^r i\lambda_i} \prod_{\rho \in \lambda} \frac{1 - q^{b + c_\rho}}{1 - q^{h_\rho}}. \tag{1.1}$$

We refer the reader to the Introduction of [5] for a detailed exposition on proofs of this formula. The bijective proofs which are known for (1.1) are the rather involved proof by

Remmel and Whitney [13], which makes use of the involution principle, and the proof by the author [5], which makes use of the Hillman-Grassl algorithm [4] and a modified jeu de taquin (the jeu de taquin being due to Schützenberger [16]). The latter bijection is actually a bijection for a rewriting of (1.1), where the "content product" is brought to the other side,

$$\Bigg(\sum_{\substack{T \text{ a tableau of shape } \lambda\\ \text{with } 1\le\text{entries}\le b}} q^{n(T)}\Bigg)\cdot\prod_{\rho\in\lambda}\frac{1}{1-q^{b+c_\rho}}=q^{\sum_{i=1}^{r} i\lambda_i}\prod_{\rho\in\lambda}\frac{1}{1-q^{h_\rho}}. \tag{1.2}$$

(It is shown in Section 3 of [5] that the same ideas also yield a bijective proof for (1.1), directly.) On the other hand, Remmel and Whitney's bijection is one for a rewriting of (1.1), where the "hook product" is brought to the other side, and where each factor of the form $(1-q^\alpha)$ is divided by $(1-q)$, so, basically for

$$\Bigg(\sum_{\substack{T \text{ a tableau of shape } \lambda\\ \text{with } 1\le\text{entries}\le b}} q^{n(T)}\Bigg)\cdot\prod_{\rho\in\lambda}(q^{-a_\rho}+q^{-a_\rho+1}+\cdots+q^{l_\rho})=\prod_{\rho\in\lambda}(q^{1-c_\rho}+q^{2-c_\rho}+\cdots+q^{b}). \tag{1.3}$$

In [5, Sec. 3] a bijection for this rewriting was constructed which is simpler than the bijection by Remmel and Whitney, using the basic algorithm from the bijection for (1.2) from the same paper. However, we did have to make use of the involution principle.

In the next section, we describe a new bijection for (1.3), which does not rely on the involution principle. It is a merge of the modified jeu de taquin idea from [5] and the Novelli-Pak-Stoyanovskii bijection [8] for the hook formula for standard Young tableaux of a given shape. A few properties of our algorithm, which are crucial in showing that it is indeed the desired bijection, are stated and proved separately in Section 3. Finally, in Section 4, we argue that our bijection can be used as an algorithm for the random generation of tableaux of a given shape with bounded entries, and we also work out the appropriate deformation explicitly which gives an algorithm for the random generation of plane partitions inside a given box.

2. Bijective proof of Stanley's hook-content formula. In this section we present a bijective proof of Theorem 1, by constructing a bijection for the equivalent identity (1.3). In all that follows, we assume that we are given a fixed partition $\lambda=(\lambda_1,\lambda_2,\dots,\lambda_r)$ and a fixed positive integer b which is at least as large as r, the number of rows of the Ferrers diagram of λ.

In order to formulate (1.3) combinatorially, it is convenient to introduce a few terms. We call a filling of the cells of λ with integers such that the entry in cell ρ is at least $-a_\rho$ and at most l_ρ, for any cell ρ of λ, a *hook tabloid of shape* λ. Furthermore, we call a filling of the cells of λ with integers such that the entry in cell ρ is at least $1-c_\rho$ and at most b, for any cell ρ of λ, a *content tabloid of shape* λ. We use the terms that we introduced for tableaux also for (either kind of) tabloids. In particular, the sum of the entries of some tabloid T is again called the *norm* of T and denoted by $n(T)$.

In terms of these notions, identity (1.3) can be rewritten as

$$\sum_{(T_L,H_L)} q^{n(T_L)}q^{n(H_L)}=\sum_{C_R} q^{n(C_R)},$$

where the left-hand side sum is over all pairs (T_L, H_L), with T_L varying over all tableaux of shape λ with entries between 1 and b and H_L varying over all hook tabloids of shape λ, and where the right-hand side sum is over all content tabloids C_R of shape λ. So the task is to set up a bijection that maps a right-hand side content tabloid C_R to a left-hand side pair (T_L, H_L), such that

$$n(T_L) + n(H_L) = n(C_R). \tag{2.1}$$

We claim that Algorithm HC, which is described below, performs this task. For the formulation of the algorithm, it is convenient to fix, once and for all, a particular total order of the cells of λ. In this order, the cells in each column are ordered from top to bottom, and cells in column 1 come before cells in column 2, etc. Figure 1.c illustrates this particular order of the cells for $\lambda = (4, 3, 3, 2)$.

The building blocks of Algorithm HC are jeu de taquin-like moves, which are explained in Definition JT below.

Definition JT. Consider a filling of shape λ with integers, and a cell ω filled with entry s. Suppose that the entries which are in the region consisting of all the successors of ω, in the fixed total order of the cells, are weakly increasing along rows and strictly increasing along columns.

We say that we *move s to the right/bottom by modified jeu de taquin*, when we perform the algorithm:

Call s the special entry.

(JT) Compare the special entry s with its right neighbour, x say (if there is no right neighbour, then, by convention, we set $x = \infty$), and its bottom neighbour, y say (if there is no bottom neighbour, then, by convention, we set $y = \infty$), see (2.2). If $s \le x$ and $s < y$, then stop.

If not, i.e. if the special entry violates weak increase along rows or strict increase along columns, then we have the following situation,

$$\begin{array}{|c|c|}\hline s & x \\ \hline y & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}\ , \tag{2.2}$$

where at least one of $s > x$ and $s \ge y$ holds. If $x + 1 < y$ then do the move

$$\begin{array}{|c|c|}\hline x+1 & s \\ \hline y & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}\ . \tag{2.3}$$

If $x \ge y - 1$ then do the move

$$\begin{array}{|c|c|}\hline y-1 & x \\ \hline s & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}\ . \tag{2.4}$$

The "new" special entry in either case is s again. Repeat (JT). (Note that always after either type of move the only possible violations of increase along rows or strict increase along columns in the region consisting of cell ω and all its successors in the fixed total order of the cells involve the special entry and the entry to the right or/and below.) $\square$

Clearly, aside from adding/subtracting 1 to/from the special entry, what happens from (2.2) to (2.3), respectively (2.4), is a *jeu de taquin forward move* (cf. [16, Sec. 2; 15, pp. 120/169]). Because of that, we call the way the special entry takes during the performance of (JT) the *forward (jeu de taquin) path of* ω, the cell where s starts.

Example JT. Figure 2 shows an example for the algorithm defined in Definition JT, with $\lambda = (6, 5, 5, 4)$ and $\omega = (2, 2)$. The array in the left half of Figure 2 displays a filling of $(6, 5, 5, 4)$ which does have the property that entries in the region consisting of all the successors of $(2, 2)$ in the fixed total order of cells are weakly increasing along rows and strictly increasing along columns. The entry in cell $(2, 2)$ is circled. The right half of Figure 2 displays the array which is obtained after application of the algorithm. The path which the special entry takes is marked by arrows, and the special entry is circled in the cell where it finally stops.

$$\begin{array}{|c|c|c|c|c|c|}\hline 4&1&2&2&3&4\\\hline 4&6&3&4&4\\\cline{1-5} 8&5&5&5&6\\\cline{1-5} 6&6&7&7\\\cline{1-4}\end{array} \quad \xrightarrow{(\mathrm{JT})} \quad \begin{array}{|c|c|c|c|c|c|}\hline 4&1&2&2&3&4\\\hline 4&4&4&4&4\\\cline{1-5} 8&5&6&6&6\\\cline{1-5} 6&6&7&7\\\cline{1-4}\end{array}$$

Figure 2

$\square$

Now, the idea of Algorithm HC is to start with a content tabloid C_R, and convert it, step by step, into a tableau T_L. This conversion uses the above modifed jeu de taquin. The moves of the entries during this modified jeu de taquin are recorded, in an appropriate way, by a hook tabloid H_L.

Algorithm HC. The input for the algorithm is a content tabloid C_R of shape λ.

(HC0) Set $(T, H) := (C_R, \mathbf{0})$, where $\mathbf{0}$ denotes the hook tabloid of shape λ with 0 in each cell. Call the last cell of λ in the fixed total order of the cells (i.e., it is the bottommost cell in the last column) the *distinguished cell.* Continue with (HC1).

(HC1) Denote the distinguished cell by ω. Move the entry in the distinguished cell to the right/bottom by modified jeu de taquin according to Definition JT. Say that it stops in cell ω'. Continue with (HC2).

(HC2) Let T be the array just obtained. Let $\omega = (i, j)$ and $\omega' = (i', j')$. The "new" H is obtained from the "old" H by moving all the entries of the old H in ω's column between row $i+1$ and i' up by one row, thereby also adding 1 to each such entry, and by setting the entry in cell (i', j) equal to $(j - j')$, the negative of the number of columns we moved to the right during (HC1). In symbols: For $k = i, i+1, \ldots, i'-1$, the entry $H_{(k,j)}$ is replaced by $H_{(k+1,j)} + 1$, and $H_{(i',j)}$ is replaced by $(j - j')$. All other entries are left unchanged.

If the distinguished cell is the first cell of λ (i.e., the topmost cell in the first column), then stop. The output of the algorithm is (T, H).

Otherwise, we choose as the "new" distinguished cell the predecessor of the "old" distinguished cell in the fixed total order of the cells. Continue with (HC1). $\square$

Example HC. A complete example for Algorithm HC can be found in the appendix. There we choose $b = 7$ and map the content tabloid of shape $(4, 3, 3, 2)$ on the left of Figure 3 to the pair on the right of Figure 3, consisting of a tableau of shape $(4, 3, 3, 2)$ with entries between 1 and 7 and a hook tabloid of shape $(4, 3, 3, 2)$, such that the weight property (2.1) is satisfied. In fact, the norm of the content tabloid on the left of Figure 3 is 46, while the norm of the tableau on the right is 48 and the norm of the hook tabloid is -2.

$$
\begin{array}{cccc} 7 & 3 & 5 & -2 \\ 7 & 3 & 2 & \\ 5 & 4 & 2 & \\ 4 & 6 & & \end{array}
\quad \longleftrightarrow \quad
\left(\begin{array}{cccc} 1 & 1 & 2 & 5 \\ 2 & 4 & 6 & \\ 4 & 5 & 7 & \\ 5 & 6 & & \end{array} \; , \; \begin{array}{cccc} 3 & 0 & -1 & 0 \\ -1 & -1 & 1 & \\ -2 & -1 & 0 & \\ 0 & 0 & & \end{array} \right)
$$

$$n(.) = 46 \qquad\qquad n(.) = 48 \;,\; n(.) = -2$$

Figure 3

The appendix has to be read in the following way. What the left columns show is the pair (T, H) that is obtained after each loop (HC1)-(HC2). Together with the pair (T, H) a picture of the shape $(4, 3, 3, 2)$ is displayed, containing certain paths. This will be important for understanding the inverse algorithm (Algorithm HC* below) but can be ignored for the moment. At each stage, the entry in the distinguished cell is circled. Then each modified jeu de taquin move during the execution of (JT) in (HC1) is displayed in the right columns. The special entry is always underlined. Once the special entry stops, the filling of $(4, 3, 3, 2)$ that is obtained is displayed again in the left column of the row below, with the special entry doubly circled. In the filling, the portion consisting of all the successors of the distinguished cell in the fixed total order of the cells (this is the portion which is "already ordered") is separated from the rest by a thick line. The hook tabloid is changed according to the rules described in (HC2) and is displayed next to the filling. $\square$

We have to prove that Algorithm HC is well-defined, that the weight property (2.1) is satisfied, and that it is a bijection between "right-hand side objects" and "left-hand side objects" of (1.3).

For Algorithm HC being well-defined, we have to show that the output is always a pair (T_L, H_L), where T_L is a tableau of shape λ with entries between 1 and b, and where H_L is a hook tabloid of shape λ.

That T_L is indeed a tableau follows from the observation that after each loop (HC1)-(HC2) the portion of the array T, obtained after that loop, consisting of cell ω and all cells which are after ω in the fixed total order of the cells, is a *skew tableau*, i.e., the entries are weakly increasing along rows and strictly increasing along columns in that portion. This observation is based on the easily verified fact that always after either type of move, (2.2)-(2.3) or (2.2)-(2.4), the only possible violations of increase along rows or strict increase along columns in that portion involve the special entry and the entry to the right or/and below, as we already mentioned.

Furthermore, the entries of T_L are at most b. For, in the beginning we start with $T = C_R$, which is an array of integers all of which are at most b. Then, in order to finally obtain T_L, we perform moves (2.2)-(2.3) and (2.2)-(2.4). We claim that at any stage, the entries will be at most b. The only possible problem could arise from the move (2.2)-(2.3), where x is increased by 1. But, by induction, we may assume that all entries are $\leq b$ before such a move. In particular, we have $s \leq b$ and $y \leq b$. Now, the move (2.2)-(2.3) is applied only if $x+1 < y$, or if y is not there and, hence, $s > x$. In either case it follows that $x + 1 \leq b$.

We claim that the entries of T_L are also at least 1. Since we already know that T_L is a tableau, it suffices to show that the entry, e say, in cell $(1,1)$ is at least 1. Now, the type of moves (2.2)-(2.3) and (2.2)-(2.4) imply that the origin of the entry e must be some entry e' in cell ρ of C_R, which was moved to the left and up according to (2.3) and (2.4), thereby adding/subtracting 1, in a way such that $e = e' + c_\rho$. Since C_R is a content tabloid, we have $e' \geq 1 - c_\rho$. Thus, it follows that $e = e' + c_\rho \geq 1$, as desired.

This proves that T_L is indeed a tableau of shape λ with entries between 1 and b.

Next, H_L is a hook tabloid, because the way the new array H is formed in (HC2) implies directly that it is a hook tabloid at any stage, and so must be H_L at the end.

Similarly, the weight property (2.1) holds because the algorithm is constructed in a way that it holds at any stage. To be precise, whenever we performed (HC2), we have

$$n(T) + n(H) = n(C_R). \tag{2.5}$$

For, equation (2.5) trivially holds at the very beginning (where we set $T = C_R$ and $H = \mathbf{0}$). Then, during each loop (HC1)-(HC2), the special entry s is moved from cell $\omega = (i,j)$ to cell $\omega' = (i',j')$. In each loop, while applying (JT) (possibly repeatedly), we add 1 to $j'-j$ entries of T, and we subtract 1 from $i'-i$ entries of T. On the other hand, in (HC2) we add 1 to $i'-i$ entries of H, and we introduce a new entry $j-j'$ into H. Obviously, these alterations add up to zero. This establishes (2.5), and thus, by spezializing (2.5) to the final output (T_L, H_L) of Algorithm HC, equation (2.1) as well.

What remains is to demonstrate that our algorithm is actually a *bijection* between right-hand side and left-hand side objects of (1.3). This will be accomplished by constructing an algorithm, Algorithm HC* below, that will turn out to be the inverse of Algorithm HC. That Algorithm HC* is well-defined will be argued after the statement of the algorithm, with two critical properties being established separately in Lemmas HC* and OC* in Section 3, respectively. That Algorithm HC and Algorithm HC* are inverses of each other will follow from the construction of the algorithms and from Lemma HC in Section 3.

As the reader may anticipate, the building blocks of Algorithm HC* are again jeu de taquin-like moves, which are, of course, inverse to those in Definition JT. We explain them in Definition JT* below.

Definition JT*. Consider a filling of shape λ with integers, a cell ω, and a cell ω', filled with entry s, which is located weakly to the right and below of ω. Suppose that the entries which are in the region consisting of cell ω and all its successors in the fixed total order of the cells are weakly increasing along rows and strictly increasing along columns.

We say that we *move s to the left/top until ω by modified jeu de taquin*, when we perform the algorithm:

Call s the special entry.

(JT*) If the special entry s is located in ω, then stop.

If not, then we have the following situation,

$$\begin{array}{|c|c|} \cline{2-2} \multicolumn{1}{c|}{} & y \\ \hline x & s \\ \hline \end{array}\ , \tag{2.6}$$

(If s should be in the same column as ω, by convention we set $x = \infty$. If y is actually not there, by convention we set $y = \infty$.) If $x - 1 > y$ then do the move

$$\begin{array}{|c|c|} \cline{2-2} \multicolumn{1}{c|}{} & y \\ \hline s & x-1 \\ \hline \end{array}\ . \tag{2.7}$$

If $x \le y + 1$ then do the move

$$\begin{array}{|c|c|} \cline{2-2} \multicolumn{1}{c|}{} & s \\ \hline x & y+1 \\ \hline \end{array}\ . \tag{2.8}$$

The "new" special entry in either case is s again. Repeat (JT*). $\square$

Again, it should be noticed that what happens from (2.6) to (2.7), respectively from (2.6) to (2.8), are *jeu de taquin backward moves* (cf. [16, Sec. 2; 15, pp. 120/169]), which reverse the forward moves (2.2)-(2.3) and (2.2)-(2.4), respectively. Because of that, we call the way that the special entry s takes during the execution of (JT*) the *backward (jeu de taquin) path of ω'* (the cell where s starts) *until ω*.

Example JT*. Figure 4 shows an example for the algorithm defined in Definition JT*, with $\lambda = (6, 5, 5, 4)$, $\omega = (2, 2)$ and $\omega' = (3, 4)$. The array in the left half of Figure 4 displays a filling of $(6, 5, 5, 4)$ which does have the property that entries in the region consisting of cell $(2, 2)$ and all its successors in the fixed total order of cells are weakly increasing along rows and strictly increasing along columns. The entry in cell $(2, 2)$ is doubly circled, the entry in cell $(3, 4)$ is circled. The right half of Figure 4 displays the array which is obtained after application of the algorithm. The path which the special entry takes is marked by arrows, and the special entry is circled in the cell where it finally stops.

$$\begin{array}{|c|c|c|c|c|c|} \hline 4 & 1 & 2 & 2 & 3 & 4 \\ \hline 4 & 4 & 4 & 4 & 4 \\ \cline{1-5} 8 & 5 & 6 & 6 & 6 \\ \cline{1-5} 6 & 6 & 7 & 7 \\ \cline{1-4} \end{array} \quad \xrightarrow{(\mathrm{JT}^*)} \quad \begin{array}{|c|c|c|c|c|c|} \hline 4 & 1 & 2 & 2 & 3 & 4 \\ \hline 4 & 6 & 3 & 4 & 4 \\ \cline{1-5} 8 & 5 & 5 & 5 & 6 \\ \cline{1-5} 6 & 6 & 7 & 7 \\ \cline{1-4} \end{array}$$

Figure 4

□

The reader should observe that there are actually *two* possible outcomes in Definition JT*: We may actually reach ω (then the algorithm stops there), or we may miss ω. As it turns out, in the situations in which the above algorithm is applied in Algorithm HC* below we do always reach ω. This is a nontrivial fact that is proved in Lemma HC*.

In order to state Algorithm HC* we need one more definition. In Algorithm HC* we need to compare different backward (jeu de taquin) paths that end in the same cell ω (and start in cells weakly to the right and below of ω). For that purpose we impose a total order on (certain) backward paths as follows.

Definition P. Given two backward paths P and Q ending in ω, which have the property that they do not cross each other (i.e., they may touch each other, but never change sides), we say that P comes before Q if one of the following three conditions is satisfied:

(a) P is neither contained in Q nor contains Q and is "to the right and above of Q", as exemplified in Figure 5, Case (a).
(b) P is contained in Q, and Q enters P from below, as exemplified in Figure 5, Case (b).
(c) P contains Q and enters Q from the right, as exemplified in Figure 5, Case (c).

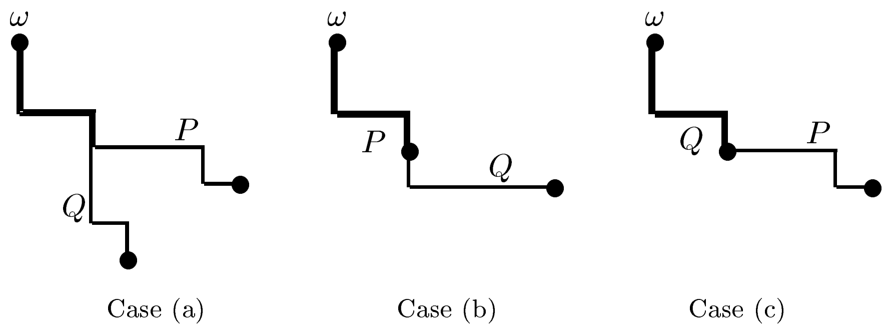


Figure 5

□

Now we are in the position to formulate Algorithm HC*.

Algorithm HC*. The input for the algorithm is a pair (T_L, H_L), where T_L is a tableau of shape λ with entries between 1 and b, and where H_L is a hook tabloid of shape λ.

(HC*0) Set $(T, H) := (T_L, H_L)$. Call the first cell of λ in the fixed total order of the cells (i.e., it is the topmost cell in the first column) the *distinguished cell.* Continue with (HC*1).

(HC*1) Let the distinguished cell be $\omega = (i, j)$. For each *nonpositive* entry, e say, of H in cell (k, j), $k \geq i$ (i.e., e is located in the same column as ω and in a row weakly below ω), mark the cell $(k, j - e)$ as a *candidate cell*.

For each candidate cell determine its backward path until ω according to Definition JT*. (In Lemma HC* below we prove that any such backward path does indeed terminate in

ω.) Let $\omega' = (i', j')$ be the candidate cell whose backward path is the first in the total order of backward paths defined in Definition P. (Note that this makes indeed sense, since, clearly, two such backward paths can never cross each other.)

Move the entry in ω' to the left/top until ω by modified jeu de taquin according to Definition JT*. Continue with (HC*2).

(HC*2) Let T be the array just obtained. The "new" H is obtained from the "old" H by moving all the entries of the old H which are in the same column as ω, i.e., in column j, and between row i and $i' - 1$ down by one row, thereby also subtracting 1 from each such entry, and by setting the entry in cell (i, j) equal to 0. In symbols: For $k = i+1, i+2, \ldots, i'$, the entry $H_{(k,j)}$ is replaced by $H_{(k-1,j)} - 1$, and $H_{(i,j)}$ is replaced by 0. All other entries are left unchanged.

If the distinguished cell is the last cell of λ (i.e., the bottommost cell in the last column), then stop. The output of the algorithm is T.

Otherwise, we choose as the "new" distinguished cell the successor of the "old" distinguished cell in the fixed total order of cells. Continue with (HC*1). □

Example HC*. A complete example for Algorithm HC* can be found in the appendix. There we choose $b = 7$ and map the pair on the right of Figure 3, consisting of a tableau of shape $(4, 3, 3, 2)$ with entries between 1 and 7 and a hook tabloid of shape $(4, 3, 3, 2)$, to the content tabloid of shape $(4, 3, 3, 2)$ on the left of Figure 3, such that the weight property (2.1) is satisfied. It is simply the inverse of the example for Algorithm HC given in Example HC. Therefore, here the appendix has to be read in the reverse direction, and in the following way. What the left columns show is the pair (T, H) that is obtained after each loop (HC*1)-(HC*2), together with a copy of the Ferrers board $(4, 3, 3, 2)$ in which the candidate cells are marked by bold dots, the distinguished cell is marked by a circle, and which contains the backward paths of all the candidate cells. The entry in the candidate cell with first backward path is doubly circled in the tableau. Then each modified jeu de taquin move during the execution of (JT*) in (HC*1) is displayed in the right columns in the row above. The special entry is always underlined. Once the special entry reaches the distinguished cell, the filling of $(4, 3, 3, 2)$ that is obtained is displayed again in the left column, with the special entry circled. In the filling, the portion consisting of all the successors of the distinguished cell in the fixed total order of the cells (this is the portion which is "still ordered") is separated from the rest by a thick line. The hook tabloid is changed according to the rules described in (HC*2) and is displayed next to the filling. □

We have to prove that Algorithm HC* is well-defined, that the weight property (2.1) is satisfied, and that it is exactly the inverse of Algorithm HC.

In order to show that Algorithm HC* is always well-defined, we have to show that whenever we are in (HC*1) the backward path of *any* candidate cell reaches ω, that when performing the changes to H in (HC*2) we always obtain a hook tabloid, and, finally, that the output is a content tabloid.

The first of these three assertions is proved in Lemma HC* in Section 3, the third is proved in Lemma OC* in Section 3.

To see that H is always a hook tabloid while performing Algorithm HC*, we argue by induction on the number of loops (HC*1)-(HC*2). We have to prove that for the entry H_ρ in cell ρ we have $-a_\rho \le H_\rho \le l_\rho$. In view of the definition of transformations on H in (HC*2), the only problem that can occur is that the lower bound, $-a_\rho$, is violated for some entry H_ρ. We will therefore concentrate on that problem.

Trivially, H is a hook tabloid at the very beginning because in (HC*0) we set $H = H_L$. Now assume that we are at some stage during the execution of Algorithm HC*, that we obtained (T, H) so far, where H is a hook tabloid, and that we have to perform (HC*1) with distinguished cell $\omega = (i, j)$ next. There we determine the candidate cells, and the candidate cell, $\omega' = (i', j')$ say, with first backward path. Then the entry in ω' is moved to the left/top until ω by modified jeu de taquin. Let P denote the backward path it takes. Next comes the application of (HC*2), where the transformations on the entries of H are performed, so that we obtain a "new" H. By induction, we do not have to worry about the entries of the "new" H that are the same as the corresponding entries in the "old" H. The only changes concerned the entries in column j between rows i and i'. Now we move to (HC*1) and determine candidate cells again. The entries of the "new" H will obey the required lower bounds if and only if all these "new" candidate cells are well-defined, i.e., if and only if they lie *inside* the Ferrers diagram of λ. The only "new" candidate cells that we have to worry about are the ones between rows i and i'. However, it follows immediately from Lemma HC* that these "new" candidate cells are located weakly to the left of the backward path P, so, at worst, these "new" candidate cells are located *on* P. Consequently, they are still located inside the Ferrers diagram. This establishes that H is indeed a hook tabloid at any stage during Algorithm HC*.

What remains is to demonstrate that Algorithm HC* and Algorithm HC are inverses of each other. We would like to show that the two algorithms reverse each other step by step. That Algorithm HC reverses Algorithm HC* is obvious. For the converse, we would need the following: Consider the pair (T, H) obtained after some loop (HC1)-(HC2) during the performance of Algorithm HC. In particular, suppose that in the last application of (HC1) we moved from cell ω to cell ω' to obtain T. If we now, with this pair (T, H) and with distinguished cell equal to ω, jump into (HC*1) of Algorithm HC*, then ω' is the candidate cell with first backward jeu de taquin path in the total order of backwards paths. (It is a candidate cell, of course.) This fact is demonstrated in Lemma HC in Section 3.

Consequently, under the assumption of the truth of the lemmas in Section 3, it is abundantly clear that Algorithms HC and HC* are well-defined and inverses of each other. This finishes the bijective proof of (1.3).

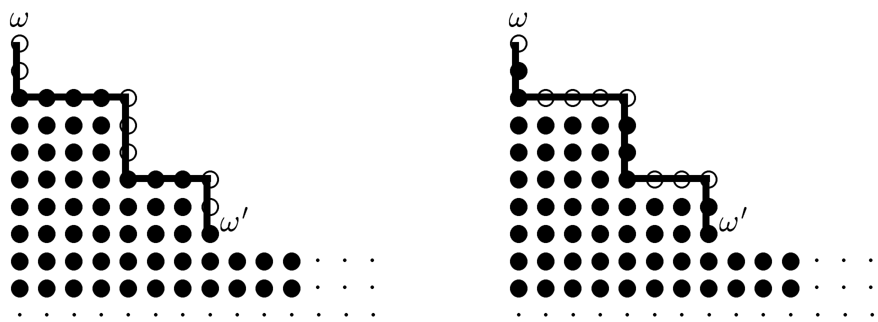
3. The crucial lemmas and their proofs.

Lemma HC*. *Let ω be a cell of λ which is not at the bottom of some column, and let $\bar\omega$ be its successor in the fixed total order of the cells of λ. Furthermore, at some stage during the execution of Algorithm HC*, let P be the backward path of modified jeu de taquin which is performed in (HC*1), connecting the candidate cell, ω' say, with first backward path in the total order of paths that was defined in Definition P with cell ω. Let $\bar\omega'$ be a candidate cell in the* subsequent *execution of (HC*1). Then the backward path of modified jeu de taquin, $\bar P$ say, which starts in $\bar\omega'$ reaches $\bar\omega$. Moreover, the backward paths*

P and $\bar{P}$ do not cross each other, and, in addition, horizontal pieces of P and $\bar{P}$ do not overlap.

Proof. This claim follows, as we shall see, from the way the modifications of H in (HC*2) are defined and from a standard property of jeu de taquin.

The statement of the lemma talks about two executions of (HC*1). Suppose we are in the first of these two, i.e., we determine candidate cells, of which ω' is the one with first backward path in the total order of backward paths. The definition of the path ordering implies that any candidate cell either is located in a lower row than ω', and if so, its backward path must enter the row which contains ω' weakly to the left of ω', or, if not, is located weakly to the left of the backward path P, but is allowed to be on the backward path only on its horizontal pieces, with their rightmost cells excluded (these are the cells on the backward path where it changes direction from north to west), see Figure 6.a (there, possible candidate cells are indicated by bold dots). Later, we will refer to these candidate cells as the "old" candidate cells.



a. possible "old" candidate cells   b. possible "new" candidate cells

Figure 6

Next comes the application of (HC*2), where the transformations on the entries of H are performed. Then we move to (HC*1) and determine candidate cells again. Because of the definition of the transformations on H in (HC*2), these "new" candidate cells are related to the "old" ones as follows: Those which were located in a lower row than ω' are still candidate cells. All the other "old" candidate cells, with the exception of ω', moved one unit to the right and one unit down to form the corresponding "new" candidate cells. Therefore, a "new" candidate cell is either located in a lower row than ω', and if so, its backward path must enter the row which contains ω' weakly to the left of ω', or, if not, is located weakly to the left of the backward path P, but is allowed to be on the backward path only on its vertical pieces, with their topmost cells excluded (these are again the cells on the backward path where it changes direction from north to west), see Figure 6.b (possible "new" candidate cells are indicated by bold dots).

The assertions of the lemma now follow from the following property of our modified jeu de taquin: If the second backward path is below of the first backward path somewhere, then it has to stay below from thereon. To make this precise, suppose that during the first execution of (HC*1) the special entry, s_1 say, went to the left by the move (2.6)-(2.7), see the left half of Figure 7. (The arrows mark the direction of move of the special entry.)

$$\begin{array}{|c|c|}\hline y & s_1 \\ \hline z & * \\ \hline\end{array} \xrightarrow{\text{during first (HC*1)}} \begin{array}{|c|c|}\hline * \leftarrow & y-1 \\ \hline z & * \\ \hline\end{array} \qquad \begin{array}{|c|c|}\hline * & y-1 \\ \hline z & s_2 \\ \hline\end{array} \xrightarrow{\text{during second (HC*1)}} \begin{array}{|c|c|}\hline * & y-1 \\ \hline * \leftarrow & z-1 \\ \hline\end{array}$$

Figure 7

Since columns are strictly increasing, we have $y < z$. Suppose that during the second execution of (HC*1) we reach the cell neighbouring z and $y-1$ with a special entry s_2, see the right half of Figure 7. Then the definition of (JT*) forces us to move left in the next step, i.e., to apply the move (2.6)-(2.7).

This concludes the proof of the lemma. $\square$

Lemma OC*. *The output of Algorithm HC* is a content tabloid.*

PROOF. Let the output of Algorithm HC* be the array C. We have to prove that the entry C_ρ of C in cell ρ satisfies

$$1 - c_\rho \le C_\rho \le b. \tag{3.1}$$

Observe first that once an entry is moved into the distinguished cell during the execution of (HC*1)-(HC*2), it stays there for the rest of Algorithm HC*, and will therefore be the corresponding entry in the output C.

It is easy to check that the moves (2.6)-(2.7) and (2.6)-(2.8) guarantee that after each loop (HC*1)-(HC*2) the portion of the array T, obtained after that loop, consisting of all cells which are after the distinguished cell in the fixed total order of the cells, is a *skew tableau*, i.e., the entries are weakly increasing along rows and the entries are strictly increasing along columns in that portion. Besides, if the distinguished cell is the bottommost cell in column j_0, say, then any entry in the first row and to the right of the distinguished cell is at least $1 - j_0$. This follows, by induction on j_0, from the definition of T in (HC*0) as a tableau with entries at least 1 and the fact that it is only the move (2.6)-(2.7) where something (namely 1) is subtracted from an entry. If we combine both facts, we obtain that if the distinguished cell is the bottommost cell in column j_0, then any entry in row i_0 and to the right of the distinguished cell is at least $i_0 - j_0$.

Now, suppose that we are in (HC*1) and are about to move an entry, s say, into the distinguished cell, $\omega = (i, j)$ say. If $i = 1$, i.e., if ω is in the first row, then the observation of the preceding paragraph (with $j_0 = j - 1$) immediately implies that this entry s is at least $1 - (j-1) = 1 - c_{(1,j)}$, which establishes the lemma in that case. If ω is not in the first row, then let $\widehat{\omega} = (i-1, j)$ be the preceding cell in the fixed total order of the cells, let $\widehat{P}$ be the backward path into $\widehat{\omega}$ of the preceding application of (HC*1), and let $\widehat{\omega}'$ be the cell where $\widehat{P}$ starts. Lemma HC* implies that the entry s comes from a cell, $(\widehat{i}, \widehat{j})$ say, which is located in a lower row than $\widehat{\omega}'$, or, if not, is located weakly to the left of the backward path $\widehat{P}$, but is allowed to be on the backward path only on its vertical pieces, with their topmost cells excluded. Furthermore, that cell $(\widehat{i}, \widehat{j})$ is not on a horizontal piece of any other "previous" backward path which terminated in column j. Consequently, again by the observation of the preceding paragraph (with $i_0 = \widehat{i}$ and $j_0 = j - 1$) we have $s \ge \widehat{i} - (j-1) \ge i - (j-1) = 1 - c_{(i,j)}$, which is exactly the assertion of the lemma. $\square$

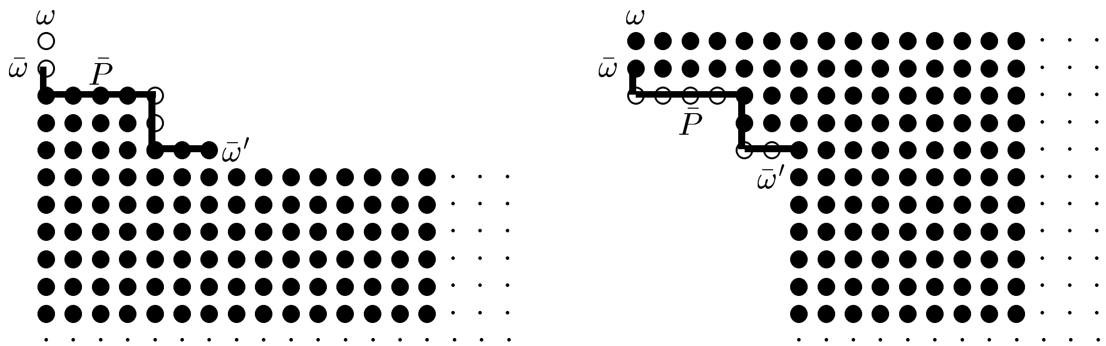
Lemma HC. *Suppose that we are at the end of a loop (HC1)-(HC2) during the execution of Algorithm HC. In particular, suppose that in the last application of (HC1) we moved from cell ω to cell ω', and that we obtained the pair (T,H) in (HC2). If we now, with this pair (T,H) and with distinguished cell equal to ω, jump into (HC*1) of Algorithm HC*, then ω' is the candidate cell with first backward jeu de taquin path in the total order of backwards paths that was defined in Definition P in Section 2.*

Proof. We prove the assertion of the lemma by *reverse* induction on the number of the cell ω in the fixed total order of the cells.

There is nothing to prove if ω is the bottommost cell in some column. In particular, the assertion is true if ω is the very last cell in the total order of cells, i.e., the bottommost cell in the last column of λ. This allows to start the induction.

Now, let us assume that ω is not the bottommost cell in some column. Then the successor of ω, $\bar\omega$ say, in the fixed total order of cells is in the same column immediately below of ω. Let $(\bar T,\bar H)$ be the outcome of the preceding loop (HC1)-(HC2), i.e., we obtain (T,H) by applying (HC1)-(HC2) to $(\bar T,\bar H)$ with distinguished cell equal to ω. On the other hand, $\bar T$ was obtained as the outcome of the previous execution of (HC1), when some entry was moved by (JT) from $\bar\omega$ to $\bar\omega'$, say. Let $\bar P$ be this foward path. By induction, we may assume that, when we jump into (HC*1) with $(\bar T,\bar H)$ and distinguished cell $\bar\omega$, the backward path of $\bar\omega'$ is the first among all backward paths of candidate cells. From now on, we will refer to these candidate cells as the "old" candidate cells.

In particular, as we already observed in the proof of Lemma HC*, this says that any "old" candidate cell either is located in a lower row than $\bar\omega'$, and if so, its backward path must enter the row which contains $\bar\omega'$ weakly to the left of $\bar\omega'$, or, if not, is located weakly to the left of the forward path $\bar P$, but is allowed to be on the forward path only on its horizontal pieces, with their rightmost cells excluded, see Figure 8.a. (There, possible "old" candidate cells are indicated by bold dots.)



a. possible "old" candidate cells b. possible territory for P

Figure 8

Next, in the execution of Algorithm HC, comes the application of (HC1)-(HC2) to obtain (T,H) out of $(\bar T,\bar H)$. Let P be the forward path from ω to ω' performed in (HC1). We claim that P has to be "above" $\bar P$ always. To make this precise, suppose that during the first execution of (HC1) the special entry, s_1 say, on its way along $\bar{P}$ went to the right by the move (2.2)-(2.3), see the left half of Figure 9. (The arrows mark the direction of move of the special entry.)

$$\begin{array}{|c|c|}\hline * & y \\ \hline s_1 & z \\ \hline\end{array} \xrightarrow{\substack{\text{during}\\ \text{first (HC1)}}} \begin{array}{|c|c|}\hline * & y \\ \hline z+1 & * \\ \hline\end{array} \qquad \begin{array}{|c|c|}\hline s_2 & y \\ \hline z+1 & * \\ \hline\end{array} \xrightarrow{\substack{\text{during}\\ \text{second (HC1)}}} \begin{array}{|c|c|}\hline y+1 & * \\ \hline z+1 & * \\ \hline\end{array}$$

Figure 9

Since columns are strictly increasing, we have $y < z$. Suppose that during the second execution of (HC1), when moving along P, we reach the cell neighbouring $z+1$ and y with a special entry s_2, see the right half of Figure 9. Then the definition of (JT) forces us to move right in the next step, i.e., to apply the move (2.2)-(2.3). As a corollary, we obtain that P must be located in the region which is above and to the right of $\bar{P}$, with no overlap of horizontal pieces of P and $\bar{P}$, as indicated in Figure 8.b. (There, the cells through which P may run are indicated by bold dots.)

Furthermore, we claim that P has to be "above" the backward path, of *any* "old" candidate cell. To make this precise, suppose that we jump into (HC*1) with $(\bar{T}, \bar{H})$ and distinguished cell $\bar{\omega}$ and determine the backward path, Q say, of an "old" candidate cell, ζ say. Let us further suppose that, while determining Q, the special entry, s_1 say, would go to the left by the move (2.6)-(2.7), see the left half of Figure 10. (The arrows mark the direction of (potential) move of the special entry.)

$$\substack{\text{during}\\ \text{(HC*1)}}\ \begin{array}{|c|c|}\hline * & y \\ \hline z & s_1 \\ \hline\end{array} \qquad \begin{array}{|c|c|}\hline s_2 & y \\ \hline z & * \\ \hline\end{array} \xrightarrow{\substack{\text{during}\\ \text{(HC1)}}} \begin{array}{|c|c|}\hline y+1 & * \\ \hline z & s_1 \\ \hline\end{array}$$

Figure 10

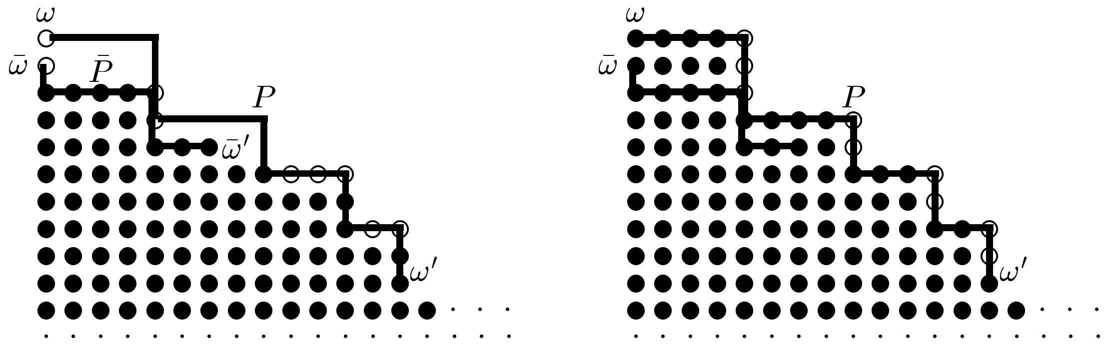
If this is the case then we must have $z-1 > y$. Now suppose that we apply (HC1)-(HC2) to $(\bar{T}, \bar{H})$, with distinguished cell $\bar{\omega}$, thus obtaining (T, H), and now apply (HC1), with distinguished cell ω. Suppose further that during the execution of (HC1), when moving along P, we reach the cell neighbouring z and y with a special entry s_2, see the right half of Figure 10. Then the definition of (JT) forces us to move to the right in the next step, i.e., to apply the move (2.2)-(2.3).

Combining the above with the already observed fact that backward paths of "old" candidate cells in a lower row than $\bar{\omega}'$ must enter the row of $\bar{\omega}'$ weakly to the left of $\bar{\omega}'$, we obtain that "old" candidate cells are located in one of the following three regions (see Figure 11.a; there, possible "old" candidate cells are indicated by bold dots):

(I) In a row strictly below ω'; then the backward path of the "old" candidate cell must enter the row of ω' weakly to the left of ω'.

(II) In a row strictly below $\bar{\omega}'$ and weakly above ω', weakly to the left of P, but on P only on its vertical pieces, with their topmost cells excluded.

(III) In a row weakly above $\bar{\omega}'$ and weakly to the left of $\bar{P}$, but on $\bar{P}$ only on its horizontal pieces, with their rightmost cells excluded.



a. possible "old" candidate cells, revisited b. possible "new" candidate cells

Figure 11

Since P is "above" $\bar{P}$, with no overlap of horizontal pieces, we may weaken the above by saying that "old" candidate cells are located in one of the following *two* regions:

(I) In a row strictly below ω'; then the backward path of the "old" candidate cell must enter the row of ω' weakly to the left of ω'.

(II) In a row weakly above ω', and weakly to the left of P, but on P only on its vertical pieces, with their topmost cells excluded.

In (HC2) the transformations on the entries of $\bar{H}$ are performed to obtain H. Because of the definition of these transformations on H in (HC2), the "new" candidate cells, i.e., those which arise if we jump into (HC*1) with (T, H) and distinguished cell ω, are related to the "old" ones, i.e., those which arise if we jump into (HC*1) with $(\bar{T}, \bar{H})$ and distinguished cell $\bar{\omega}$, as follows: Those "old" candidate cells which were located in a lower row than ω' are "new" candidate cells as well. All the other "old" candidate cells moved one unit to the left and one unit up to form the corresponding "new" candidate cells. Therefore, "new" candidate cells are located in one of the following two regions (see Figure 11.b; as before, possible "new" candidate cells are indicated by bold dots):

(I) In a row strictly below ω'; then the backward path of the "new" candidate cell must enter the row of ω' weakly to the left of ω'.

(II) In a row weakly above ω', and weakly to the left of P, but on P only on its horizontal pieces, with their rightmost cells excluded.

Consequently, the backward path of a "new" candidate cell is necessarily later then P in the total order of backward paths.

This completes the proof of the lemma. □

4. Random generation of semistandard tableaux and of plane partitions. The purpose of this section is to demonstrate that Algorithm HC of Section 2 can be used as a device for the random generation of semistandard tableaux of a given shape and with bounded entries, and for the random generation of plane partitions inside a given box.

Let $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_r)$ be a partition and b an integer $\geq r$. We claim that semistandard tableaux of shape λ with entries between 1 and b can be randomly generated as follows:

Choose a content tabloid C of shape λ at random. This is, of course, easy, as we may choose the entry in each cell ρ of C between $1 - c_\rho$ and b independently and at random. Now apply Algorithm HC to C. This gives a pair (T, H). Discard the hook tabloid H, and define the output of this procedure to be the semistandard tableau T. (Since we discard H at the end, we actually do not have to worry about H during the execution of Algorithm HC. I.e., the procedure may be streamlined further by not assigning anything to H in (HC0) and by disregarding the operations on H every time (HC2) is performed.)

T is indeed a random semistandard tableaux of shape λ with entries between 1 and b, because, as demonstrated before, Algorithm HC is a bijection between content tabloids C and pairs (T, H) where T is a semistandard tableaux of shape λ with entries between 1 and b, and where H is a hook tabloid of shape λ. This implies in particular that any hook tabloid H of shape λ is equally likely as second component of the pair (T, H). Therefore, when discarded, any semistandard tableau T of shape λ with entries between 1 and b is equally likely as first component.

Finally we turn to the random generation of plane partitions. Recall that, given a partition λ, a *plane partition of shape* λ is a filling P of the cells of λ with integers such that the entries along rows and columns are weakly decreasing. We are interested here in plane partitions of shape (c^a) (this is an $a \times c$ rectangle, i.e., a Ferrers diagram with a rows of length c) with entries between 0 and b. Because plane partitions can also be viewed geometrically as certain piles of unit cubes (cf. [14], for instance), these plane partitions are also called *plane partitions inside an* $a \times b \times c$ *box.*

Plane partitions inside an $a \times b \times c$ box are in bijection with semistandard tableaux of shape (c^a) with entries between 1 and $a + b$. This is seen by replacing each entry, e say, in the i-th row of such a plane partition by $b - e + i$. Therefore Algorithm HC, or rather a proper deformation of it, can also be used for the random generation of plane partitions inside a given box. We give this deformation explicitly as Algorithm PP below.

Algorithm PP. The input for the algorithm is a filling F of the shape (c^a) with the entry in cell (i, j) between $i - a$ and $b + j - 1$, $i = 1, 2, \ldots, a$, $j = 1, 2, \ldots, c$.

(PP0) Set $P := F$. Call the last cell of λ in the fixed total order of cells (i.e., the cell (a, c)) the *distinguished cell*, and call the entry it contains *special.* Continue with (PP1).

(PP1) Compare the special entry s with its right neighbour, x say (if there is no right neighbour, then, by convention, we set $x = \infty$), and its bottom neighbour, y say (if there is no bottom neighbour, then, by convention, we set $y = \infty$), see (4.1). If $s \geq x$ and $s \geq y$, then continue with (PP2).

If not, i.e. if the special entry violates weak decrease along rows or columns, then we have the following situation,

$$\begin{array}{|c|c|}\hline s & x \\ \hline y & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}\ , \tag{4.1}$$

where at least one of $s < x$ and $s < y$ holds. If $x-1 \ge y$ then do the move

$$\begin{array}{|c|c|}\hline x-1 & s \\ \hline y & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}\ . \tag{4.2}$$

If $x \le y$ then do the move

$$\begin{array}{|c|c|}\hline y & x \\ \hline s+1 & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}\ . \tag{4.3}$$

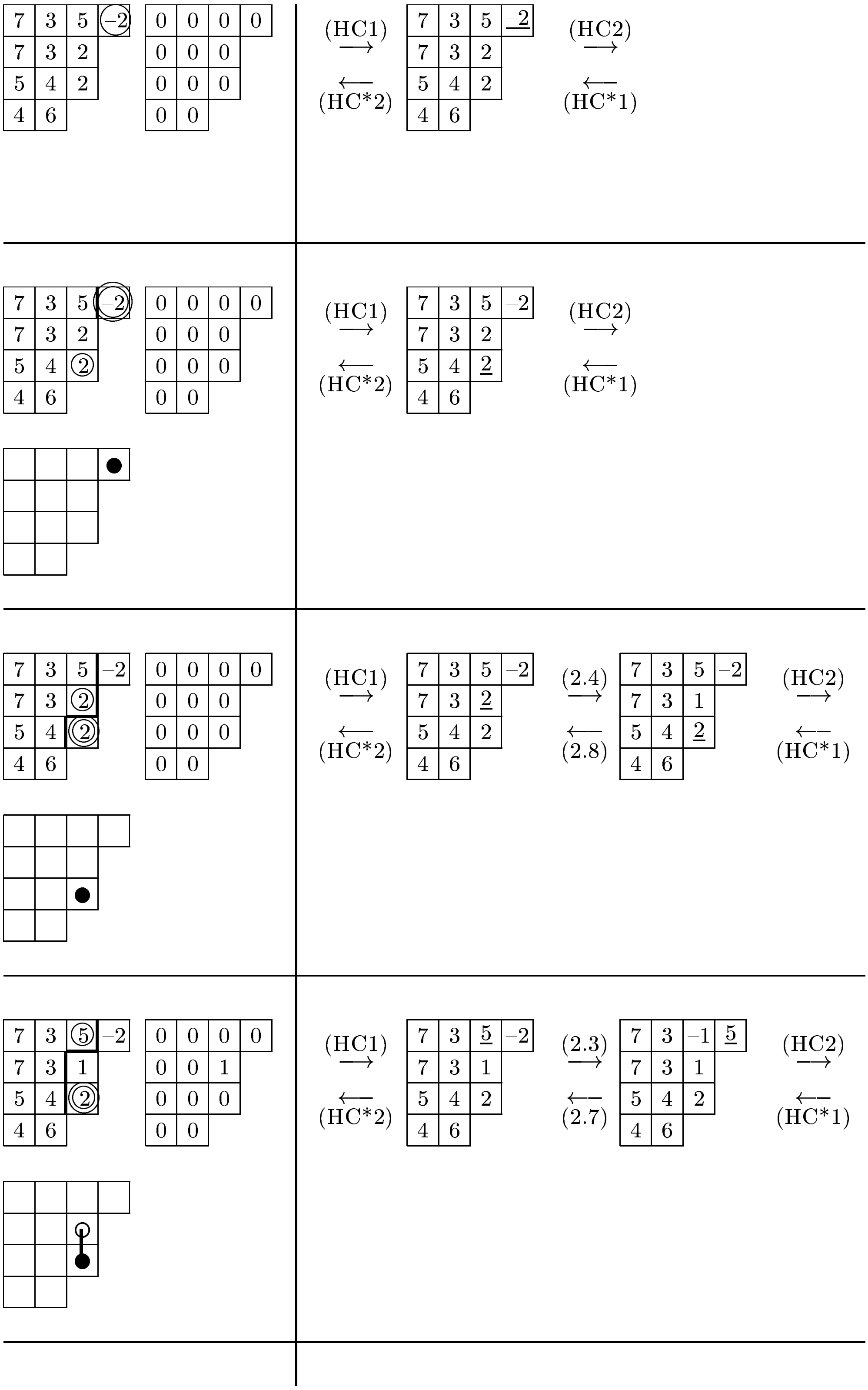
The "new" special entry in (4.2) is s again, the "new" special entry in (4.3) is $s+1$. Repeat (PP1).

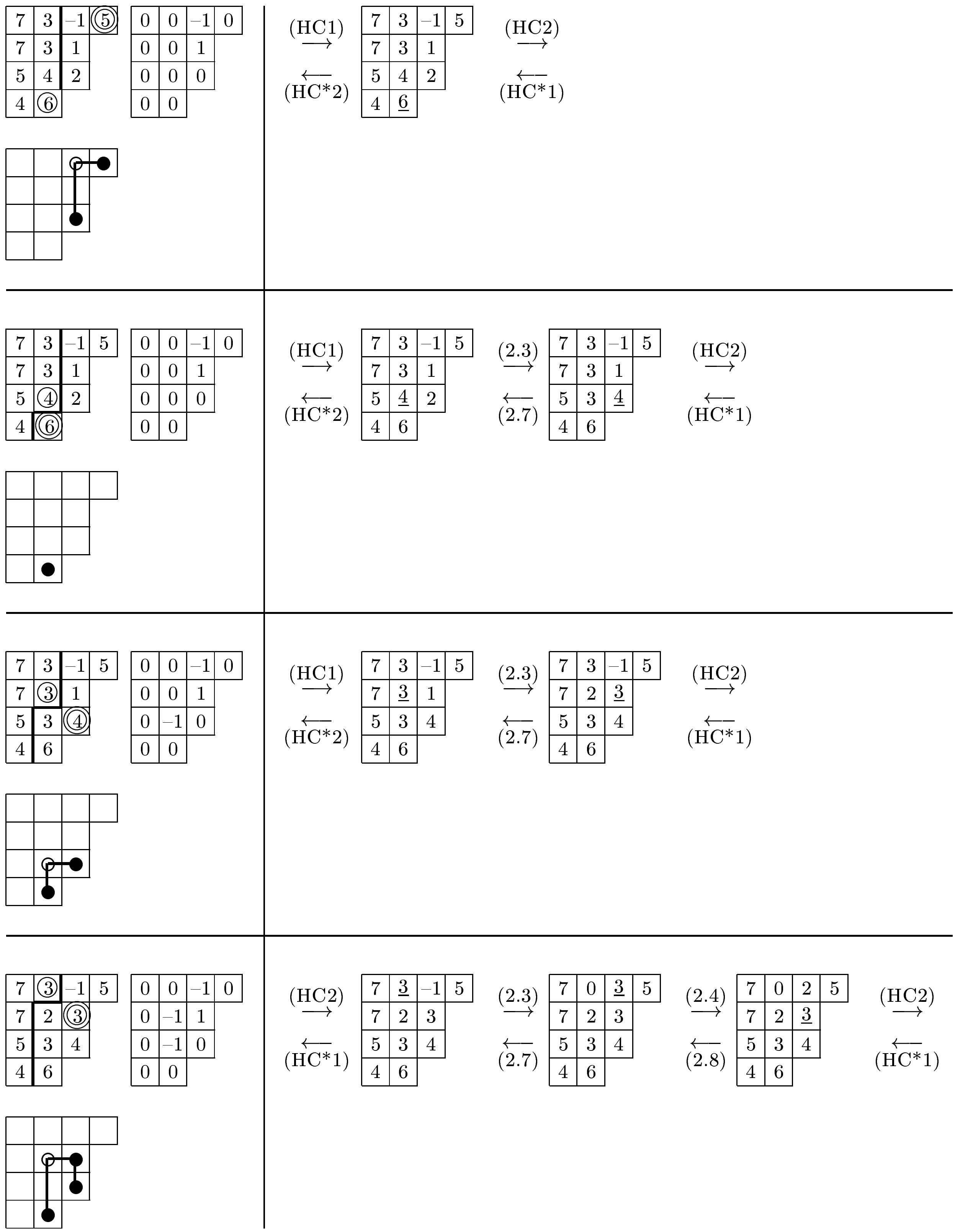
(PP2) Let P be the array just obtained. If ω is the first cell of (c^a) (i.e., the cell $(1,1)$), then stop. The output of the algorithm is P.

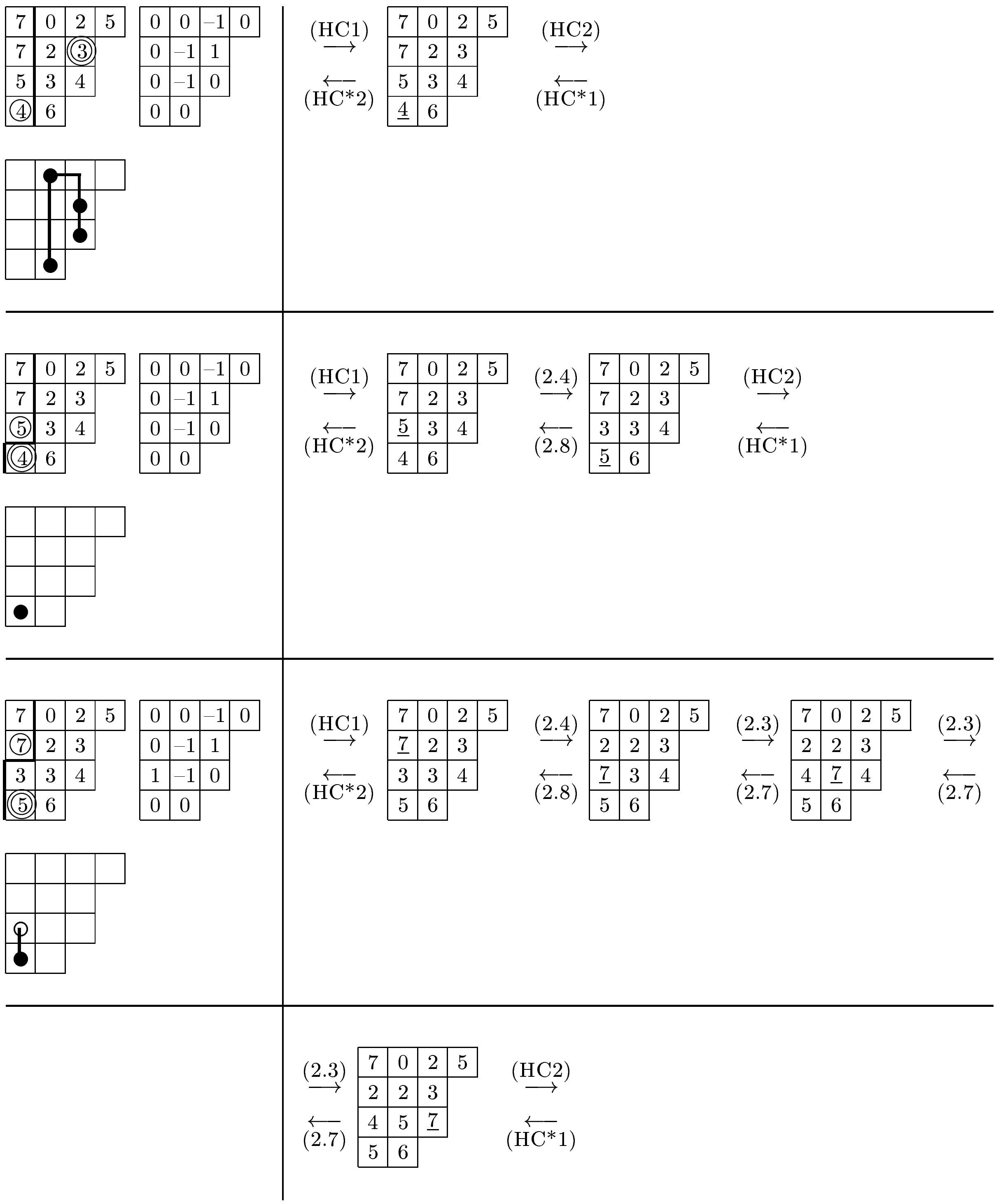
Otherwise, we choose as the "new" distinguished cell the predecessor of the "old" distinguished cell in the fixed total order of the cells, and as the "new" special entry the entry it contains. Continue with (PP1). $\square$

There are already algorithms for the random generation of semistandard tableaux of a given shape with bounded entries and of plane partitions inside a given box in the literature. These are the appropriate specializations of algorithms by Luby, Randall and Sinclair [6], and by Propp and Wilson [10, 11, 18]. The approaches of these authors are however quite different. In [6, 10, 11] they propose a method which is based on a Markov chain approach called "coupling from the past" (see also [12]). Wilson's algorithms [18] are based on a determinant algorithm originally developed by Colbourn et al. [1]. (There are two algorithms in [18], one is based on Kasteleyn determinants, the other on Gessel–Viennot determinants.) It is then interesting to compare the performance of our algorithms with the algorithms by Propp and Wilson. For example, the complexity of Algorithm PP, measured in terms of the number of modified jeu de taquin moves (4.1)-(4.2) and (4.1)-(4.3) which has to be performed, is at worst $O\big(\sum_{i=a}^{a+c-1}\binom{i}{2}\big) = O\big(c(3a^2+3ac+c^2)\big)$. (The average case complexity is expected to be much better, probably quadratic in a and c.) The complexity of Wilson's algorithm based on Kasteleyn determinants [18, Sec. 3] on the other hand is between $O\big((ab+bc+ca)^{3/2}\big)$ and $O\big((ab+bc+ca)^{5/2}\log^{\alpha}(ab+bc+ca)\big)$, depending on the size of a, b, c, while the complexity of Wilson's algorithm based on Gessel–Viennot determinants [18, Sec. 5] is $O(b^{1.688}(ab+bc+ca))$. The complexity of the "coupling from the past" algorithms [6, 10, 11] is $O\big((a+b)^2(ab+bc+ca)\log(ab+bc+ca)\big)$ (this is partially conjectural; see [19, Sec. 5]). So, the performance of our algorithm is (at least) in the range of the others if a, b, c are roughly of the same magnitude, but should be better if one of the quantities a, b, c is much larger than the other two, because there is no dependence on b in the performance of our algorithm. It has to be emphasized, however, that our algorithm is rather specialized, and does not extend to cases that are still covered by the algorithms by Luby, Randall, and Sinclair, respectively by Propp and Wilson.

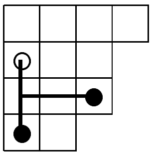
Appendix. The appendix contains a complete example for Algorithms HC and HC* for $\lambda = (4,3,3,2)$ and $b = 7$, setting up a mapping between the two sides of Figure 3. See the specific descriptions given in Examples HC and HC* of how to read the following tables.


(HC1)
(HC*2)
(HC2)
(HC*1)
(HC1)
(HC*2)
(HC2)
(HC*1)
(HC1)
(HC*2)
(2.4)
(2.8)
(HC2)
(HC*1)
(HC1)
(HC*2)
(2.3)
(2.7)
(HC2)
(HC*1)




(HC1)
(HC*2)
(HC2)
(HC*1)
(2.4)
(2.8)
(2.3)
(2.7)

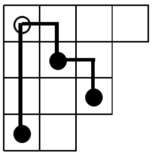
$$\begin{array}{|c|c|c|c|}\hline 7&0&2&5\\ \hline 2&2&3&\multicolumn{1}{c}{}\\ \cline{1-3} 4&5&7&\multicolumn{1}{c}{}\\ \cline{1-3} 5&6&\multicolumn{2}{c}{}\\ \cline{1-2}\end{array} \quad \begin{array}{|c|c|c|c|}\hline 0&0&-1&0\\ \hline 2&-1&1&\multicolumn{1}{c}{}\\ \cline{1-3} -2&-1&0&\multicolumn{1}{c}{}\\ \cline{1-3} 0&0&\multicolumn{2}{c}{}\\ \cline{1-2}\end{array}$$



$$\underset{(HC^*2)}{\overset{(HC1)}{\rightleftarrows}} \begin{array}{|c|c|c|c|}\hline \underline{7}&0&2&5\\ \hline 2&2&3&\multicolumn{1}{c}{}\\ \cline{1-3} 4&5&7&\multicolumn{1}{c}{}\\ \cline{1-3} 5&6&\multicolumn{2}{c}{}\\ \cline{1-2}\end{array} \underset{(2.7)}{\overset{(2.3)}{\rightleftarrows}} \begin{array}{|c|c|c|c|}\hline 1&\underline{7}&2&5\\ \hline 2&2&3&\multicolumn{1}{c}{}\\ \cline{1-3} 4&5&7&\multicolumn{1}{c}{}\\ \cline{1-3} 5&6&\multicolumn{2}{c}{}\\ \cline{1-2}\end{array} \underset{(2.8)}{\overset{(2.4)}{\rightleftarrows}} \begin{array}{|c|c|c|c|}\hline 1&1&2&5\\ \hline 2&\underline{7}&3&\multicolumn{1}{c}{}\\ \cline{1-3} 4&5&7&\multicolumn{1}{c}{}\\ \cline{1-3} 5&6&\multicolumn{2}{c}{}\\ \cline{1-2}\end{array} \underset{(2.7)}{\overset{(2.3)}{\rightleftarrows}}$$

$$\underset{(2.7)}{\overset{(2.3)}{\rightleftarrows}} \begin{array}{|c|c|c|c|}\hline 1&1&2&5\\ \hline 2&4&\underline{7}&\multicolumn{1}{c}{}\\ \cline{1-3} 4&5&7&\multicolumn{1}{c}{}\\ \cline{1-3} 5&6&\multicolumn{2}{c}{}\\ \cline{1-2}\end{array} \underset{(2.8)}{\overset{(2.4)}{\rightleftarrows}} \begin{array}{|c|c|c|c|}\hline 1&1&2&5\\ \hline 2&4&6&\multicolumn{1}{c}{}\\ \cline{1-3} 4&5&\underline{7}&\multicolumn{1}{c}{}\\ \cline{1-3} 5&6&\multicolumn{2}{c}{}\\ \cline{1-2}\end{array} \underset{(HC^*1)}{\overset{(HC2)}{\rightleftarrows}}$$

$$\begin{array}{|c|c|c|c|}\hline 1&1&2&5\\ \hline 2&4&6&\multicolumn{1}{c}{}\\ \cline{1-3} 4&5&7&\multicolumn{1}{c}{}\\ \cline{1-3} 5&6&\multicolumn{2}{c}{}\\ \cline{1-2}\end{array} \quad \begin{array}{|c|c|c|c|}\hline 3&0&-1&0\\ \hline -1&-1&1&\multicolumn{1}{c}{}\\ \cline{1-3} -2&-1&0&\multicolumn{1}{c}{}\\ \cline{1-3} 0&0&\multicolumn{2}{c}{}\\ \cline{1-2}\end{array}$$



References

1. C. J. Colbourn, W. J. Myrvold and E. Neufeld, *Two algorithms for unranking arborescences*, J. Algorithms **20** (1996), 268–281.
2. J. S. Frame, G. B. Robinson and R. M. Thrall, *The hook graphs of the symmetric group*, Canad. J. Math. **6** (1954), 316–325.
3. A. M. Garsia and S. C. Milne, *Method for constructing bijections for classical partition identities*, Proc. Nat. Acad. Sci. U.S.A. **78** (1981), 2026–2028.
4. A. P. Hillman and R. M. Grassl, *Reverse plane partitions and tableau hook numbers*, J. Combin. Theory Ser. A **21** (1976), 216–221.
5. C. Krattenthaler, *An involution principle-free bijective proof of Stanley's hook-content formula*, Discrete Math. Theoret. Computer Science **3** (1998), 011–032.
6. M. Luby, D. Randall and A. Sinclair, *Markov chain algorithms for planar lattice structures*, (extended abstract), 36th Annual Symposium on Foundations of Computer Science, 1995, pp. 150–159.
7. P. A. MacMahon, *Combinatory Analysis*, Cambridge University Press, 1916; reprinted by Chelsea, New York, 1960.
8. J. C. Novelli, I. M. Pak and A. V. Stoyanovsii, *A direct bijective proof of the hook-length formula*, Discrete Math. Theoret. Computer Science **1** (1997), 53–67.
9. I. M. Pak and A. V. Stoyanovskii, *A bijective proof of the hook-length formula and its analogues*, Funkt. Anal. Priloz. **26**, No. 3 (1992), 80–82, English translation in Funct. Anal. Appl. **26** (1992), 216–218.
10. J. Propp, *Generating random elements of finite distributive lattices*, Electron. J. Combin. **4** (no. 2, "The Wilf Festschrift") (1997), #R15, 12 pp.
11. J. Propp and D. B. Wilson, *Exact sampling with coupled Markov chains and applications to statistical mechanics*, Random Structures Algorithms **9** (1996), 223–252.
12. J. Propp and D. B. Wilson, *Coupling from the past: a user's guide*, Microsurveys in Discrete Probability (D. Aldous, J. Propp, eds.), DIMACS Ser. Discrete Math. Theoret. Comput. Sci., vol. 41, Amer. Math. Soc.,, Providence, R.I., 1998, pp. 181–192.
13. J. B. Remmel and R. Whitney, *A bijective proof of the hook formula for the number of column-strict tableaux with bounded entries*, Europ. J. Combin. **4** (1983), 45–63.
14. D. P. Robbins, *The story of* $1, 2, 7, 42, 429, 7436, \dots$, The Math. Intelligencer **13** (1991), 12–19.
15. B. E. Sagan, *The symmetric group*, Wadsworth & Brooks/Cole, Pacific Grove, California, 1991.
16. M.-P. Schützenberger, *La correspondance de Robinson*, Combinatoire et Représentation du Groupe Symétrique, Lecture Notes in Math., vol. 579, Springer–Verlag, Berlin–Heidelberg–New York, 1977, pp. 59–113.
17. R. P. Stanley, *Theory and applications of plane partitions: Part 2*, Stud. Appl. Math **50** (1971), 259–279.
18. D. B. Wilson, *Determinant algorithms for random planar structures*, Proceedings of the Eighth Annual ACM–SIAM Symposium on Discrete Algorithms, 1997, pp. 258–267.
19. D. B. Wilson, *Mixing times of lozenge tiling and card shuffling Markov chains*, manuscript in preparation.

Institut für Mathematik der Universität Wien, Strudlhofgasse 4, A-1090 Wien, Austria.