A BIJECTIVE PROOF OF THE HOOK-CONTENT FORMULA FOR SUPER SCHUR FUNCTIONS AND A MODIFIED JEU DE TAQUIN

C. Krattenthaler†

Institut für Mathematik der Universität Wien,
Strudlhofgasse 4, A-1090 Wien, Austria.
e-mail: KRATT@Pap.Univie.Ac.At

Submitted: February 6, 1995; Accepted: July 25, 1995

Dedicated to Dominique Foata

Abstract. A bijective proof of the product formula for the principal specialization of super Schur functions (also called hook Schur functions) is given using the combinatorial description of super Schur functions in terms of certain tableaux due to Berele and Regev. Our bijective proof is based on the Hillman–Grassl algorithm and a modified version of Schützenberger's jeu de taquin. We then explore the relationship between our modified jeu de taquin and a modified jeu de taquin by Goulden and Greene. We define a common extension and prove an invariance property for it, thus discovering that both modified jeu de taquins are, though different, equivalent.

1. Introduction. Let λ be a partition, i.e. $\lambda=(\lambda_1,\lambda_2,\dots,\lambda_r)$ is a sequence of integers with $\lambda_1\ge\lambda_2\ge\dots\ge\lambda_r>0$. The *super Schur function* $\mathcal{S}_\lambda(\mathbf{x},\mathbf{y})$ (or *hook Schur function*), where $\mathbf{x}=(x_1,x_2,\dots)$ and $\mathbf{y}=(y_1,y_2,\dots)$ are two sequences of indeterminates, is defined by

$$\mathcal{S}_\lambda(\mathbf{x},\mathbf{y})=\det_{1\le i,j\le r}(\mathcal{S}_{\lambda_i-i+j}(\mathbf{x},\mathbf{y})), \tag{1.1}$$

where the series $\mathcal{S}_m(\mathbf{x},\mathbf{y})$ are determined by their generating function

$$\sum_{m\ge0}\mathcal{S}_m(\mathbf{x},\mathbf{y})z^m=\prod_{i\ge1}\frac{1+y_iz}{1-x_iz}. \tag{1.2}$$

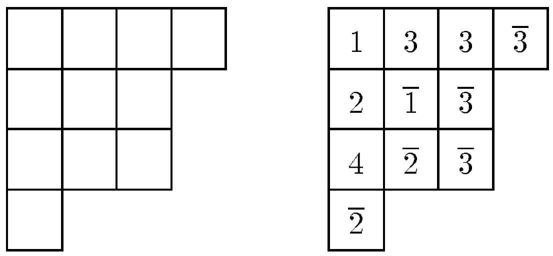
1991 *Mathematics Subject Classification.* Primary 05A15; Secondary 05A17, 05A30, 05E10, 05E15, 11P81.

Key words and phrases. Plane partitions, tableaux, hook-content formula, Hillman–Grassl algorithm, jeu de taquin, super Schur function.

†Supported in part by EC's Human Capital and Mobility Program, grant CHRX-CT93-0400 and the Austrian Science Foundation FWF, grant P10191-PHY

Super Schur functions appear naturally in the representation theory of Lie superalgebras (cf. [2, 3, 9, 10, 11, 17]), and were also discovered independently under the name of supersymmetric polynomials [14].

In order to be able to state the formula alluded to in the title, we have to recall some basic notions from partition theory. The *Ferrers diagram* of a partition $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_r)$ 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 $(4, 3, 3, 1)$. The *conjugate of* λ is the partition $(\lambda'_1, \dots, \lambda'_{\lambda_1})$ 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 *content* c_ρ of a cell $\rho = (i, j)$ of λ is $j - i$.



a. Ferrers diagram b. super semistandard tableau

Figure 1

Now we are in the position to state the hook-content formula for the principal specialization of super Schur functions.

Theorem 1. *Let $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_r)$ be a partition. For the specialization $x_i = aq^i$, $y_i = bq^i$, $i = 1, 2, \dots$, in the super Schur function $\mathcal{S}_\lambda(\mathbf{x}, \mathbf{y})$ there holds*

$$\mathcal{S}_\lambda((aq, aq^2, aq^3, \dots), (bq, bq^2, bq^3, \dots)) = q^{\sum_{i=1}^r i\lambda_i} \prod_{\rho \in \lambda} \frac{a + bq^{c_\rho}}{1 - q^{h_\rho}}. \tag{1.3}$$

Theorem 1 can be found, as everything about symmetric functions, in a slightly different, but equivalent form, in Macdonald's book [13, p. 27, Ex. 5, p. 45, Ex. 3].

For our bijective proof of this hook-content formula we use the combinatorial description of super Schur functions in terms of what I call *super semistandard tableaux* (see section 2 for the definition), originally introduced by Berele and Regev [2] under the name of (k, l)-*semistandard tableaux*, a name however which does not make sense in our context. Our proof is based on the Hillman–Grassl algorithm [8] and on Schützenberger's [18] jeu de taquin. It is inspired by our bijective proof [12] of Stanley's hook-content formula for the generating function for column-strict reverse plane partitions of a given shape with bounded entries. Though there are quite a few similarities with [12], it turns out that the algorithms and arguments in [12] have to be adapted quite a bit in order to provide a bijective proof of (1.3).

We recall all the needed tableau and plane partition definitions in section 2. Then, in section 3, we present our bijective proof of the hook-content formula (1.3) for super Schur functions. In the appendix we carry out a complete example for our bijection. After a first version of this article was written, I discovered that a jeu de taquin procedure similar to ours already appeared in a paper by Goulden and Greene [7]. In fact, they are different *algorithms*, but it turns out that as *mappings* they are the same. We explore the relationship between the two algorithms in section 4. In order to prove the equivalence of the two algorithms, we define a joint extension of the algorithms, and establish an invariance property of the underlying modified jeu de taquin.

2. Super semistandard tableaux. Super semistandard tableaux are hybrids of column-strict reverse plane partitions and row-strict reverse plane partitions. We are going to define the latter first. Let $\lambda=(\lambda_1,\lambda_2,\dots,\lambda_r)$ be a partition. A *reverse plane partition of shape* λ is a filling of the cells of (the Ferrers diagram of) λ with entries from an ordered alphabet such that the entries along rows and along columns are weakly increasing. A reverse plane partition is called *column-strict* if in addition columns are *strictly* increasing. Likewise, a reverse plane partition is called *row-strict* if rows are *strictly* increasing. We will also consider *skew shapes* for row-strict reverse plane partitions and column-strict reverse plane partitions. If λ and μ are two partitions with $\lambda\ge\mu$ (i.e. the Ferrers diagram of μ is contained in the Ferrers diagram of λ), then the *skew shape* λ/μ consists of all cells that are contained in λ but not in μ. As expected, by a *row-strict* (*respectively column-strict*) *reverse plane partition of shape* λ/μ we mean a filling of the cells of λ/μ with entries from some ordered alphabet such that columns (respectively rows) are weakly increasing and rows (respectively columns) are strictly increasing.

Now, a *super semistandard tableau of shape* λ is a filling S of the cells of λ with entries from the ordered alphabet $1<2<3<\cdots<\overline{1}<\overline{2}<\overline{3}<\cdots$ such that

(SST1) the unbarred entries form a column-strict reverse plane partition of some shape ν, where ν is a partition,

(SST2) the barred entries form a row-strict reverse plane partition of shape λ/ν.

Figure 1.b shows a super semistandard tableau of shape $(4,3,3,1)$. We call the part of a super semistandard tableau S that is occupied by the unbarred entries the *inner part* of S and denote it by S^I. Likewise, we call the part of S that is occupied by the barred entries the *outer part* of S and denote it by S^O.

The combinatorial interpretation of the super Schur function $\mathcal{S}_\lambda(\mathbf{x},\mathbf{y})$ is (here SST is short for 'super semistandard tableau')

$$\mathcal{S}_\lambda(\mathbf{x},\mathbf{y})=\sum_{\substack{S\text{ a SST}\\\text{of shape }\lambda}}\prod_{i\ge1}x_i^{\#i\text{'s in }S}y_i^{\#\bar{i}\text{'s in }S},\tag{2.1}$$

which can be proven combinatorially by the Gessel–Viennot method [5, 6] of nonintersecting lattice paths as shown by Remmel [15, sec. 5] (see also Brenti [4, Theorem 3.4]).

For convenience, we introduce the following two terms. Given any filling P (a super semistandard tableau, or a reverse plane partition, or ...) of some (possibly skew) shape, we write P_ρ, respectively $\overline{P_\rho}$, for the entry in cell ρ of P, depending on whether we mean the unbarred or barred integer, respectively. Also, we call the sum of all the entries of P (forgetting about all the bars when taking the sum) the *norm* of P, and denote it by $n(P)$. For example, the norm of the super semistandard tableau in Figure 1.b is 27.

3. Bijective proof of the hook-content formula for super Schur functions. In view of (2.1), our bijective proof of the hook-content formula (1.3) will consist of proving

$$\sum_{\substack{S \text{ a SST}\\ \text{of shape}\lambda}} a^{|S^I|} b^{|S^O|} q^{n(S)} = q^{\sum_{i=1}^r i\lambda_i} \prod_{\rho\in\lambda} \frac{1}{1-q^{h_\rho}} \prod_{\rho\in\lambda} (a+bq^{c_\rho}) \tag{3.1}$$

bijectively, where, as expected, $|S^I|$ and $|S^O|$ denote the numbers of entries in the inner part and outer part of S, respectively.

While it is obvious what the combinatorial interpretation of the left-hand side of (3.1) is, we need to introduce yet a few more terms for being able to conveniently describe the right-hand side of (3.1) in combinatorial terms. We call an *arbitrary* filling of the cells of λ with *nonnegative* integers a *tabloid of shape* λ. We define the *hook weight* $w_h(T)$ of a tabloid T of shape λ by $\sum_{\rho\in\lambda} T_\rho \cdot h_\rho$. Furthermore, we call a filling of the cells of λ with only 0's and 1's a (0–1)-*tabloid*. We define the *content weight* $w_c(T)$ of a (0–1)-tabloid T by $\sum_{\rho\in\lambda} T_\rho \cdot c_\rho$. Then the right-hand side of (3.1) is the generating function

$$\sum q^{n(P_0)} q^{w_h(T_R)} q^{w_c(U_R)} a^{\#0\text{'s in } U_R} b^{\#1\text{'s in } U_R},$$

where the sum is over all triples (P_0, T_R, U_R), with P_0 being the "minimal" column-strict reverse plane partition of shape λ with entries from $1<2<3<\cdots$, i.e. the column-strict reverse plane partition with all entries in row i equal to i for all i, with T_R varying over all tabloids of shape λ, and with U_R varying over all (0–1)-tabloids of shape λ. So the task is to set up a bijection that maps a super semistandard tableau S_L of shape λ to such a triple (P_0, T_R, U_R), such that

$$n(S_L) = n(P_0) + w_h(T_R) + w_c(U_R) \tag{3.2}$$

and

$$|(S_L)^I| = \text{number of 0's in } U_R, \qquad \text{and} \qquad |(S_L)^O| = \text{number of 1's in } U_R. \tag{3.3}$$

One step in our bijection was already done much earlier. In their celebrated paper [8], Hillman and Grassl constructed an algorithmic bijection between tabloids T_R of shape λ and reverse plane partitions $\tilde{P}_R$ of shape λ with entries from $0<1<2<\cdots$ such that $n(\tilde{P}_R) = w_h(T_R)$. If we add such a reverse plane partition $\tilde{P}_R$ to P_0 cell-wise, then we obtain a *column-strict* reverse plane partition P_R of shape λ with entries from $1<2<3<\cdots$, and we have $n(P_R) = n(P_0) + n(\tilde{P}_R) = n(P_0) + w_h(T_R)$. Therefore the new task is to set up a bijection that maps a super semistandard tableau S_L of shape λ to a pair (P_R, U_R), where

P_R is a column-strict reverse plane partition of shape λ with entries from $1 < 2 < 3 \cdots$, and where U_R is a (0–1)-tabloid of shape λ, such that

$$n(S_L) = n(P_R) + w_c(U_R) \tag{3.4}$$

and (3.3) hold.

We claim that the following algorithm performs this task.

Algorithm S. The input for the algorithm is a super semistandard tableau S_L of shape λ.

(S0) Set $(S, U) := (S_L, \mathbf{0})$, where $\mathbf{0}$ denotes the (0–1)-tabloid of shape λ with 0 in each cell.

(S1) If S does not contain any barred entry then stop. The output of the algorithm is (S, U).

Otherwise, consider all *inward corner cells* of the outer part S^O of S (which are the cells in S^O with left and upper neighbour cells in S^I). Choose all such inward corner cells that contain the minimal barred entry of S, and among all these pick the top-most, cell ω say. (Note that the minimal barred entry of S *must* appear in an inward corner cell of S^O since S^O is a row-strict reverse plane partition.) Replace the entry $\overline{S_\omega}$ in cell ω by $S_\omega - c_\omega$. (Note that we "unbarred" the entry in this cell. Also note that $S_\omega - c_\omega$ might be nonpositive, which does not bother us at this point.) Call this entry *special*. Continue with (S2).

(S2) If the special entry, s say, does not violate (SST1), i.e. if s does not violate weak increase along rows and strict increase along columns (in the unbarred entries, of course), then continue with (S3).

If it does, then we have the following situation,

$$\begin{array}{cc} & \boxed{y} \\ \boxed{x} & \boxed{s} \end{array}\ , \tag{3.5}$$

where at least one of $x > s$ and $y \geq s$ holds. (One of x or y is also allowed to be actually not there.) If $x > y$ then do the move

$$\begin{array}{cc} & \boxed{y} \\ \boxed{s+1} & \boxed{x} \end{array}\ . \tag{3.6}$$

If $x \leq y$ then do the move

$$\begin{array}{cc} & \boxed{s-1} \\ \boxed{x} & \boxed{y} \end{array}\ . \tag{3.7}$$

The new special entry in (3.6) is $s + 1$, the new special entry in (3.7) is $s - 1$. Repeat (S2). (Note that always after either type of move the only possible violations of increase along rows or strict increase along columns involve the new special entry and the entry to the left or/and above.)

(S3) Let S be the super semistandard tableau just obtained. (The fact that indeed a super semistandard tableau is obtained will be proved right after Example S.) If we ended up with the special entry in cell ρ then add 1 to the entry in cell ρ of U. The tabloid thus obtained is the new U. (It will be proved in Lemma S below that U is in fact a (0–1)-tabloid.) Continue with (S1). $\square$

Example S. A complete example for Algorithm S can be found in the appendix. There we map the super semistandard tableau of shape $(4,3,3,1)$ on the left of Figure 2 to the pair on the right of Figure 2, consisting of a column-strict reverse plane partition of shape $(4,3,3,1)$ with entries from $1<2<3<\cdots$ and a (0–1)-tabloid of shape $(4,3,3,1)$, such that the weight properties (3.3) and (3.4) hold. In fact, the norm of the semistandard tableau on the left of Figure 2 is 27, while the norm of the column-strict reverse plane partition on the right is 31 and the content weight of the (0–1)-tabloid is -4, which verifies (3.4) in this case. That (3.3) holds in our example is obvious.

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

$$\begin{array}{r}
n(.)=27 \\
|\text{inner part}|=5 \\
|\text{outer part}|=6
\end{array}
\qquad\qquad
\begin{array}{r}
n(.)=31\ ,\ w_c(.)=-4 \\
\#0\text{'s}=5 \\
\#1\text{'s}=6
\end{array}$$

Figure 2

The appendix has to be read in the following way. First of all, ignore all double circles, and all even rows in the right columns. What the left columns show is the pair (S,U) that is obtained after each loop (S1)-(S2)-(S3). Together with the pair (S,U) a filling of the shape $(4,3,3,1)$ is displayed that shows all values $S_\rho+c_\rho$ for all cells ρ with $U_\rho\neq 0$. This will be important for understanding Lemma S but can be ignored for the moment. At each stage, the entry that is chosen by (S1) is circled. Then each intermediate step during the loop (S1)-(S2)-(S3) is displayed in the odd rows of the right columns. The special entry is always underlined. When a super semistandard tableau is reached, the special entry is boxed. The entry in the corresponding cell of the (0–1)-tabloid is subsequently increased by 1 in step (S3). $\square$

It should be noticed that, aside from adding/subtracting 1 to/from the special entry, what happens from (3.5) to (3.6), respectively (3.7), is a *jeu de taquin forward move* (cf. [18, sec. 2; 16, pp. 120/169]).

We have to justify the claim in (S3) that when we arrive at (S3) we obtained a super semistandard tableau. We do this by induction on the number of loops (S1)-(S2)-(S3). For the induction hypothesis assume that we enter (S1) with a pair (S,U), where S is a super semistandard tableau. (This is true for our initial pair $(S_L,\mathbf{0})$). It is obvious that when we arrive at (S3) that S was transformed into a filling that satisfies (SST1) (otherwise the step (S2) was not finished) and (SST2) (because the only change in the outer part, the part containing the barred entries, was to remove an entry in an inward corner cell from it and make the entry into an unbarred entry). Therefore the only problem that could arise is when the special entry in the end is ≤ 0. However, when we arrive at (S3), a special entry ≤ 0 can only occur in cell $(1,1)$, because otherwise step (S2) was not finished. Each loop (S1)-(S2)-(S3) of the algorithm starts with some entry $\overline{S_\omega} \ge \overline{1}$ in an inward corner cell ω of the outer part S^O of S. It is replaced by $S_\omega - c_\omega$. Then it is (possibly) moved according to (3.6) and (3.7). It is easy to check that at each stage during performing the steps (S2), the special entry, if located in cell ρ, will equal $S_\omega - c_\rho$. This is a property so important that it has to be recorded for later use,

$$(\text{special entry in } \rho) = S_\omega - c_\rho. \tag{3.8}$$

Suppose we reach cell $(1,1)$. When we arrive at $(1,1)$, by (3.8) and since $c_{(1,1)} = 0$, our special entry has become S_ω. But this is ≥ 1 since $\overline{S_\omega} \ge \overline{1}$.

So we have an algorithm that maps a super semistandard tableau S_L to a pair (P_R, U_R), where P_R is a super semistandard tableau without any barred entry, or in plain words, a column-strict reverse plane partition, and where U_R, thanks to Lemma S (cf. the Remark after Lemma S), is a (0–1)-tabloid. Stated in different terms, the algorithm maps left-hand side objects of (3.1) to right-hand side objects of (3.1). Besides, this mapping satisfies the weight properties (3.3) and (3.4). While (3.3) is trivially satisfied, (3.4) follows from (3.8).

What remains is to establish that our algorithm is actually a *bijection* between left-hand side and right-hand side objects. This will be accomplished by constructing an algorithm, Algorithm S* below, that will turn out to be the inverse of Algorithm S. To motivate the definition of Algorithm S*, we note the following lemma.

Lemma S. *Let (S,U) be obtained after some loop* (S1)-(S2)-(S3) *during Algorithm S. Suppose that the loop terminated in cell γ when reaching (S3). Then γ cannot be reached by a special entry during succeeding loops (S1)-(S2)-(S3). Also, among all cells ρ with $U_\rho \ne 0$, γ is a cell for which the value $S_\rho + c_\rho$ attains its maximum, and if there are several cells ρ with $U_\rho \ne 0$ where the maximum is attained, then γ is the left-most and bottom-most of those. Besides, there holds*

$$\overline{S_\gamma + c_\gamma} \le \min\{\text{entries in } S^O\}. \tag{3.9}$$

Remark. Note that the property that γ cannot be reached by "later" special entries inductively implies that after each loop (S1)-(S2)-(S3) the tabloid U is in fact a (0–1)-tabloid.

Proof of the Lemma. We prove the assertions by induction on the number of loops (S1)-(S2)-(S3).

For the start of the induction it will suffice to recall that we start Algorithm S with the pair $(S_L, \mathbf{0})$, where S_L is a super semistandard tableau.

As induction hypothesis we assume that (S,U) is either the initial pair $(S_L, \mathbf{0})$, or obtained after some loop (S1)-(S2)-(S3) in which case all the assertions of the Lemma are assumed to be true. Let ω be the cell where the next loop starts, i.e. the inward corner cell of S^O chosen by (S1), and let ζ be the cell where this next loop terminates at (S3). Let $(\tilde{S}, \tilde{U})$ be the outcome after this loop. Note that by construction the cells ρ with $\tilde{U}_\rho \neq 0$ are ζ and the cells with $U_\rho \neq 0$. In particular, $U_\rho \neq 0$ implies $\tilde{U}_\rho \neq 0$.

First we prove (3.9) for $\tilde{S}$ and ζ. By (3.8), the entry of $\tilde{S}$ in cell ζ is

$$\tilde{S}_\zeta = S_\omega - c_\zeta. \tag{3.10}$$

Since by definition of (S1) we have $\overline{S_\omega} = \min\{\text{entries in } S^O\}$, this immediately implies

$$\overline{\tilde{S}_\zeta + c_\zeta} = \overline{S_\omega} = \min\{\text{entries in } S^O\} \leq \min\{\text{entries in } \tilde{S}^O\}. \tag{3.11}$$

The last inequality is due to the fact that the multiset of entries in $\tilde{S}^O$ comes out of the multiset of entries in S^O by removing $\overline{S_\omega}$. Obviously, (3.11) proves (3.9) for $\tilde{S}$ and ζ, as desired.

Next we prove that the cell ζ cannot be reached during a "later" loop (S1)-(S2)-(S3). We distinguish between two cases. Namely, by definition of (S1), a "later" loop (S1)-(S2)-(S3) starts either at a cell, κ say, strictly to the right of ω or weakly to the left and strictly below of ω.

In the former case, by definition of (S1) and the fact that the outer part S^O of S is a row-strict reverse plane partition, we have

$$\tilde{S}_\kappa = S_\kappa > S_\omega. \tag{3.12}$$

Then, when performing the loop (S1)-(S2)-(S3), first $\overline{\tilde{S}_\kappa}$ is replaced by $\tilde{S}_\kappa - c_\kappa$ and then (possibly) moved inwards according to (3.6) and (3.7). If we want to reach ζ, we have to reach the neighbour cell to the right or the neighbour cell below first. By (3.8) we would reach the neighbour cell to the right of ζ with the special entry being equal to $\tilde{S}_\kappa - (c_\zeta + 1)$. By (3.12) this is $\geq S_\omega - c_\zeta = \tilde{S}_\zeta$, the latter equality holding because of (3.10). Therefore, by definition of (S2), the loop either already stops here or continues with an upward move (3.7), and so does not reach ζ in any case. On the other hand, if we reach the neighbour cell below ζ, by (3.8) the special entry would equal $\tilde{S}_\kappa - (c_\zeta - 1)$. By (3.12) this is $> S_\omega - c_\zeta = \tilde{S}_\zeta$. Again, by definition of (S2), this says that the loop either already stops here or continues with a left move (3.6), so it cannot reach ζ in any case.

The second case is that we start our "later" loop in a cell κ that is located weakly to the left and strictly below of ω. By definition of (S1) we have

$$\tilde{S}_\kappa = S_\kappa \geq S_\omega. \tag{3.13}$$

Again, if we want to reach ζ, we have to reach either the neighbour cell to the right or the neighbour cell below of ζ first. We claim that it cannot happen that the cell to the right is reached. This would follow immediately from the following easy-to-check property of our modified forward jeu de taquin (S2): If the second "jeu de taquin path" is below the first "jeu de taquin path" somewhere, then it has to stay below from thereon. To make this precise, suppose that during the first loop the special entry, s_1 say, went to the left by

(3.7), see the left half of Figure 3. (The arrows mark the direction of move of the special entry.)

Figure 3

Since columns are strictly increasing in the inner part, we have $y < z$. Suppose that during the second loop we reach the cell neighbouring y and z with a special entry s_2, see the right half of Figure 3. Then by definition of the algorithm we have to stop here or, if not, we are forced to move left in the next step (S2). It is our assumption that the second "jeu de taquin path" starts below the first, therefore it has to stay below always.

Now we know that the only possibility to reach cell ζ is by reaching the neighbour cell below first. However, when we reach this neighbour cell below of ζ, by (3.8) the special entry equals $\tilde{S}_\kappa - (c_\zeta - 1)$. By (3.13) this is $> S_\omega - c_\zeta = \tilde{S}_\zeta$. Once more, by definition of (S2), this says that the loop either already stops here or continues with a left move (3.6), so it cannot reach ζ in any case.

Next we show that $\tilde{S}_\rho + c_\rho$, evaluated at cells ρ with $\tilde{U}_\rho \neq 0$, attains its maximal value at ζ. This is evident if (S, U) is the initial pair $(S_L, \mathbf{0})$, i.e. if $(\tilde{S}, \tilde{U})$ is the pair obtained by applying one loop (S1)-(S2)-(S3) to $(S_L, \mathbf{0})$, because then ζ is the only cell with $\tilde{U}_\zeta \neq 0$. So, let us assume that (S, U) is obtained after some loop (S1)-(S2)-(S3). Let γ be the cell where the next-to-last loop, which gave rise to (S, U), terminated at (S3).

Let ρ be any cell with $\tilde{U}_\rho \neq 0$. Recall that this means $\rho = \zeta$ or $U_\rho \neq 0$. We want to show

$$\tilde{S}_\rho + c_\rho \leq \tilde{S}_\zeta + c_\zeta. \tag{3.14}$$

If $\rho = \zeta$ then there is nothing to show. So let $\rho \neq \zeta$. Then we have $U_\rho \neq 0$. By induction hypothesis for γ, we have $S_\gamma + c_\gamma \geq S_\rho + c_\rho$ for all cells with $U_\rho \neq 0$. Also by induction hypothesis, we know that ρ cannot be reached during a "later" loop (S1)-(S2)-(S3), therefore we have $\tilde{S}_\rho = S_\rho$. And, again by induction hypothesis, (3.9) holds for S and γ. Hence we have

$$\overline{S_\gamma + c_\gamma} \leq \min\{\text{entries in } S^O\} = \overline{S_\omega} = \overline{\tilde{S}_\zeta + c_\zeta},$$

the equalities holding because of (3.11). Combining everything, we conclude

$$\tilde{S}_\rho + c_\rho = S_\rho + c_\rho \leq S_\gamma + c_\gamma \leq \tilde{S}_\zeta + c_\zeta, \tag{3.15}$$

which verifies (3.14).

Finally we show that ζ is the left-most and bottom-most among all cells ρ with $\tilde{U}_\rho \neq 0$ where $\tilde{S}_\rho + c_\rho$ attains its maximal value. Again, this is trivially satisfied if (S, U) is the initial pair $(S_L, \mathbf{0})$ since then ζ is the only cell with $\tilde{U}_\zeta \neq 0$. So, let us assume that (S, U) is obtained after some loop (S1)-(S2)-(S3). Also here, let γ be the cell where the next-to-last loop, which gave rise to (S,U), terminated at (S3). Let ρ be a cell with $\tilde{U}_\rho \neq 0$ where $\tilde{S}_\rho + c_\rho$ attains its maximal value. In particular, we have

$$\tilde{S}_\rho + c_\rho = \tilde{S}_\zeta + c_\zeta. \tag{3.16}$$

Of course, nothing is to show for $\rho = \zeta$, so we may assume $\rho \neq \zeta$. This implies $U_\rho \neq 0$. We already observed that then the relations (3.15) hold. Combining (3.15) and (3.16) we are forced to conclude

$$S_\rho + c_\rho = S_\gamma + c_\gamma = \tilde{S}_\zeta + c_\zeta. \tag{3.17}$$

We shall show that ζ has to lie in the region (weakly) to the left and (weakly) below of γ. Since γ was the left-most and bottom-most of all the cells ρ with $U_\rho \neq 0$ where the maximal value of $S_\rho + c_\rho$ is attained, and since $S_\rho + c_\rho = \tilde{S}_\rho + c_\rho$ for all cells with $U_\rho \neq 0$, this would establish that ζ is the left-most and bottom-most of all the cells ρ with $\tilde{U}_\rho \neq 0$ where the maximal value of $\tilde{S}_\rho + c_\rho$ is attained, as desired.

We prove the claim of the previous paragraph by excluding the other three quarter regions that are determined by the horizontal line and the vertical line running through γ.

First, suppose that ζ lies in the region strictly to the right and weakly below of γ. For these cells there holds the following basic computation. For convenience, let $\gamma = (i_1, j_1)$ and $\zeta = (i_2, j_2)$, where ζ is located in this region strictly to the right and weakly below of γ, i.e. $i_1 \le i_2$ and $j_1 < j_2$. Then, since the inner part $\tilde{S}^I$ of $\tilde{S}$ is a column-strict reverse plane partition, we have

$$\begin{aligned}\tilde{S}_\zeta + c_\zeta &= \tilde{S}_\zeta + (j_2 - i_2)\\ &\ge (\tilde{S}_\gamma + i_2 - i_1) + (j_2 - i_2) = \tilde{S}_\gamma + j_2 - i_1\\ &> \tilde{S}_\gamma + j_1 - i_1 = \tilde{S}_\gamma + c_\gamma.\end{aligned} \tag{3.18}$$

This contradicts (3.17) because of $S_\gamma = \tilde{S}_\gamma$, which again follows from the induction hypothesis that γ cannot be reached during "later" loops (S1)-(S2)-(S3). Thus, this region is excluded.

Next we show that ζ cannot lie in the region (weakly) to the right and (weakly) above γ, γ excluded. This would follow immediately from the claim that if two successive loops (S1)-(S2)-(S3) start with the same size of entry in the inward corner cells chosen by (S1) (which applies in our case since the loop that lead to S started with an entry $\overline{S_\gamma + c_\gamma}$ in some inward corner cell, and the loop that lead from S to $\tilde{S}$ started with $\overline{\tilde{S}_\zeta + c_\zeta}$, both quantities being the same by (3.17)) then the second path of moves has to stay below the first path of moves.

To check this claim, once again note that both loops started with the same size of entries in the cells chosen by (S1). By the rules in (S1), and because the outer part S^O of S is a row-strict reverse plane partition, this means that the second loop started strictly to the left and strictly below of the first. We already proved that our modified forward jeu de taquin (S2) has the property that if the second "jeu de taquin path" is below of the first "jeu de taquin path" somewhere, then it has to stay below from thereon. So the region (weakly) to the right and (weakly) above γ can never be reached.

Finally, we examine if ζ could be located in the region strictly to the left and weakly above of γ. Since $S_\gamma = \tilde{S}_\gamma$, once more the computation (3.18), with γ and ζ interchanged applies, implying $\tilde{S}_\gamma + c_\gamma > \tilde{S}_\zeta + c_\zeta$, which contradicts (3.17) because of $S_\gamma = \tilde{S}_\gamma$. Therefore we cannot reach the region under consideration.

This completes the proof of the Lemma. $\square$

¿From Lemma S it is pretty obvious what the inverse algorithm of Algorithm S could be.

Algorithm S*. The input for the algorithm is a pair (P_R, U_R), where P_R is a column-strict reverse plane partition of shape λ with entries from $1 < 2 < 3 < \cdots$, and where U_R is a (0–1)-tabloid of shape λ.

(S*0) Set $(S, U) := (P_R, U_R)$.

(S*1) If $U = \mathbf{0}$ then stop. The output of the algorithm is S.

Otherwise, consider all cells ρ with $U_\rho \neq 0$. Among these choose the cells for which $S_\rho + c_\rho$ is maximal, and among all these pick the left-most and bottom-most, cell ζ say. (Observe that among two different cells attaining the same value of $S_\rho + c_\rho$ one is always weakly to the left and strictly below of the other, again because of the computation (3.18), with $\tilde{S}$ replaced by S. So the left-most and bottom-most of these does exist.) Replace the entry S_ζ in cell ζ by $\overline{S_\zeta + c_\zeta}$. (Note that we "barred" the entry.) Call this entry *special*. Continue with (S*2).

(S*2) If the special entry, s say, does not have a right or bottom neighbour entry that is unbarred, then continue with (S*3).

If the special entry does have a right or bottom neighbour entry that is unbarred, then we have the following situation,

$$\begin{array}{|c|c|}\hline s & x \\ \hline y & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}\ , \tag{3.19}$$

where at least one of x or y is an unbarred entry. (One of x or y is also allowed to be actually not there.) If $x < y$ (in particular, this is the case if y is a barred entry) then do the move

$$\begin{array}{|c|c|}\hline x & s \\ \hline y & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}\ . \tag{3.20}$$

If $x \geq y$ (in particular, this is the case if x is a barred entry) then do the move

$$\begin{array}{|c|c|}\hline y & x \\ \hline s & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}\ . \tag{3.21}$$

The new special entry in either case is s. Repeat (S*2).

(S*3) Let S be the super semistandard tableau just obtained. (The fact that indeed a super semistandard tableau is obtained will be proved in the subsequent Lemma S*.) Replace the entry 1 in cell ζ of U by 0. The (0–1)-tabloid thus obtained is the new U. Continue with (S*1). $\square$

EXAMPLE S*. A complete example for Algorithm S* can be found in the appendix. There we map the pair on the right of Figure 2, consisting of a column-strict reverse plane partition of shape $(4,3,3,1)$ with entries from $1<2<3<\cdots$ and a (0–1)-tabloid of shape $(4,3,3,1)$, to the super semistandard tableau of shape $(4,3,3,1)$ on the left of Figure 2, such that the weight properties (3.3) and (3.4) hold. It is simply the inverse of the example for Algorithm S given in Example S. Therefore, here the appendix has to be read in the reverse direction, and in the following way. First of all, ignore all single circles, and all odd rows in the right columns. What the left columns show is the pair (S,U) that is obtained after each loop (S*1)-(S*2)-(S*3) together with a filling of the shape $(4,3,3,1)$ that shows all values $S_\rho+c_\rho$ for all cells ρ with $U_\rho\neq 0$. At each stage, the entry that is chosen by (S*1) is doubly circled. Then each intermediate step during the loop (S*1)-(S*2)-(S*3) is displayed in the even rows of the right columns. The special entry is always doubly underlined. When a super semistandard tableau is reached, the special entry is doubly boxed. The entry in the corresponding cell of the (0–1)-tabloid is subsequently decreased by 1 in step (S*3). □

Again, it should be noticed that (3.20) and (3.21) are exactly *jeu de taquin backward moves* (cf. [18, sec. 2; 16, pp. 120/169]), which reverse the forward moves (3.6) and (3.7), respectively, except for the subtraction/addition of 1 in (3.6) and (3.7).

In order to show that Algorithm S* is always well-defined, we have to confirm that when arriving at (S*3) we always obtained a super semistandard tableau. This is established in the following lemma. Besides, this lemma contains the facts about Algorithm S* that are needed to prove that the Algorithms S and S* are inverses of each other.

Lemma S*. *Let (S,U) be obtained after some loop* (S*1)-(S*2)-(S*3) *during Algorithm S*. Then for all cells ρ with $U_\rho\neq 0$ there holds*

$$\overline{S_\rho+c_\rho}\le \min\{\text{entries in } S^O\}. \tag{3.22}$$

Also, S is a super semistandard tableau. Besides, if ω is the inward corner cell of S^O that contained the special entry at the end of the loop (S*1)-(S*2)-(S*3) *that lead to S, then ω is the top-most inward corner cell of S^O that contains the minimal entry of S^O.*

PROOF. We prove the assertions by induction on the number of loops (S*1)-(S*2)-(S*3). To be precise, we prove the following (slightly stronger) statement inductively: All the assertions of the Lemma hold. Besides, during the loop (S*1)-(S*2)-(S*3) that leads to (S,U) the special entry never occupies a cell ρ with $U_\rho\neq 0$.

To begin with, we know that when we start with Algorithm S* we have a pair (S,U), where S is a super semistandard tableau (since $S=P_R$ is a column-strict reverse plane partition in $1<2<3<\cdots$). Also, (3.22) is vacuously satisfied (there is no barred entry in $S=P_R$). This will suffice for the start of the induction.

As induction hypothesis let us assume that the assertions of the Lemma and the assertion that the special entry never occupies a cell ρ with $U_\rho\neq 0$ are true for (S,U) and all preceding pairs occuring in step (S*3) during the process of the algorithm, except of course that the assertion about the inward corner cell ω and the assertion about the special entry do not hold for the initial pair (because they do not make sense for the initial pair).

Let ζ be the cell where the loop (S*1)-(S*2)-(S*3) starts from (S,U), i.e. the cell that is chosen by applying (S*1) to (S,U), and let κ be the cell where the loop stops at (S*3). Furthermore, let $(\tilde{S},\tilde{U})$ be the outcome after this loop. Then, by definition of the algorithm we have

$$\overline{\tilde{S}_\kappa} = \overline{S_\zeta + c_\zeta}. \tag{3.23}$$

Note that, also by definition of the algorithm, the cells ρ with $\tilde{U}_\rho \neq 0$ are those with $U_\rho \neq 0$, excluding ζ. In particular, $\tilde{U}_\rho \neq 0$ implies $U_\rho \neq 0$.

First we prove that during the loop (S*1)-(S*2)-(S*3) leading from (S,U) to $(\tilde{S},\tilde{U})$ the special entry never occupies a cell ρ with $\tilde{U}_\rho \neq 0$. Recall that, by definition of (S*1), the cell ζ, the cell where the loop starts, is the left-most and bottom-most cell among the cells with $U_\zeta \neq 0$ where the value $S_\zeta + c_\zeta$ is maximal. Let the special entry be located in some cell ρ with $\tilde{U}_\rho \neq 0$, at some stage during the performance of the loop. Clearly, ρ is located (weakly) to the right and (weakly) below of ζ. Since, by definition of (S*3), we have $\tilde{U}_\zeta = 0$, cell ρ is different from ζ. If ρ were located *strictly* to the right of ζ, then the computation (3.18), with $\tilde{S}$ replaced by S, ζ replaced by ρ, and γ replaced by ζ, would imply that $S_\rho + c_\rho > S_\zeta + c_\zeta$. But this contradicts the definition of ζ. So the only other possibility is that ρ is located in the same column as ζ and strictly below ζ. Then a computation similar to (3.18) would show that $S_\rho + c_\rho = S_\zeta + c_\zeta$, which again contradicts the definition of ζ. Therefore the special entry can only meet cells ρ with $\tilde{U}_\rho = 0$. In particular, this implies that the entries in cells ρ with $\tilde{U}_\rho \neq 0$ do not change during the loop. Therefore we have

$$\tilde{S}_\rho = S_\rho \tag{3.24}$$

for all cells ρ with $\tilde{U}_\rho \neq 0$.

Now we prove (3.22) for $\tilde{S}$. Let ρ be any cell with $\tilde{U}_\rho \neq 0$. This implies $U_\rho \neq 0$. Therefore, by (3.24) and by construction of ζ in (S*1), we have

$$\tilde{S}_\rho + c_\rho = S_\rho + c_\rho \leq S_\zeta + c_\zeta. \tag{3.25}$$

Also by construction of ζ, we have $U_\zeta \neq 0$, and hence by induction hypothesis (3.22) that $\overline{S_\zeta + c_\zeta} \leq \min\{\text{entries in } S^O\}$. This implies that $\overline{S_\zeta + c_\zeta} = \min\{\text{entries in } \tilde{S}^O\}$, since the multiset of entries of $\tilde{S}^O$ equals the multiset of entries in S^O, with the special entry $\overline{S_\zeta + c_\zeta}$, created in (S*1) and finally located in cell κ in $\tilde{S}$, added. Hence, (3.25) proves (3.22) with S replaced by $\tilde{S}$, as desired.

Now we prove that $\tilde{S}$ is a super semistandard tableau. By construction of Algorithm S*, the inner part $\tilde{S}^I$ of $\tilde{S}$ automatically is a column-strict reverse plane partition. Therefore, it is only to prove that the outer part $\tilde{S}^O$ of $\tilde{S}$ is a row-strict reverse plane partition. If $\overline{S_\zeta + c_\zeta} < \min\{\text{entries in } S^O\}$ then this assertion certainly holds, since $\overline{\tilde{S}_\kappa} = \overline{S_\zeta + c_\zeta}$ is the only new entry in $\tilde{S}$, and it is located in an inward corner cell of $\tilde{S}^O$. Note in particular, that we are in this case at the very beginning.

By induction hypothesis, (3.22) holds for ζ, so the only other case is

$$\overline{S_\zeta + c_\zeta} = \min\{\text{entries in } S^O\}. \tag{3.26}$$

Observe that the only difficulty arises when we reach the cell κ at the end of a loop (S*1)-(S*2)-(S*3), and the neighbour cell to the right of κ contains the entry $\overline{S_\zeta + c_\zeta}$. In this case row-strictness of the outer part $\tilde{S}^O$ of $\tilde{S}$ would be violated. We have to show that this case cannot occur.

Let (S', U') be the pair preceding (S, U), i.e. (S, U) is obtained by applying one loop (S*1)-(S*2)-(S*3) to (S', U'). As we just noted, (S', U') exists, since if (S, U) were the initial pair we would not be in this case. Furthermore, let γ be the cell where this loop starts, and let ω be the corner cell where it stops. By definition of the algorithm we have

$$\overline{S_\omega} = \overline{S'_\gamma + c_\gamma}. \tag{3.27}$$

Now, by induction hypothesis and the definition of the algorithm, there holds

$$\overline{S'_\gamma + c_\gamma} = \min\{\text{entries in } S^O\}. \tag{3.28}$$

Furthermore, there holds $S'_\zeta = S_\zeta$, since by induction hypothesis the cell ζ (which is a cell with $U_\zeta \neq 0$) was not met by the special entry during the loop (S*1)-(S*2)-(S*3) that lead from (S', U') to (S, U). Hence, by definition of (S*1), $S'_\gamma + c_\gamma \geq S'_\zeta + c_\zeta = S_\zeta + c_\zeta$. Combining this with (3.26) and (3.28), we are forced to conclude

$$S'_\gamma + c_\gamma = S'_\zeta + c_\zeta \; (= S_\zeta + c_\zeta). \tag{3.29}$$

Therefore, again by definition of (S*1), ζ lies weakly to the right and strictly above γ.

It is an easy-to-check property of backward jeu de taquin (S*2) that if the second "jeu de taquin path" is above the first "jeu de taquin path" somewhere, then it has to stay above from thereon. To be precise, suppose that during the first loop (S*1)-(S*2)-(S*3) the special entry, s_1 say, went to the right by (3.20), see the left half of Figure 4. (Again, the arrows mark the direction of move of the special entry.)

Figure 4

Since columns are strictly increasing in the inner part, we have $y < z$. Suppose that during the second loop we reach the cell neighbouring y and z with a special entry s_2, see the right half of Figure 4. Then by definition of the algorithm we have to stop here or, if not, we are forced to move right in the next step (S*2).

We know that the second "jeu de taquin path" starts at ζ, which is weakly to the right and strictly above of γ, the starting cell of the first "jeu de taquin path". Therefore the second path has to stay above the first path always. Hence κ, the cell where the second path terminates, cannot be the left neighbour cell of ω. Therefore, κ is in a row above ω. By induction hypothesis, ω is the top-most inward corner cell of S^O that contains the minimal entry of S^O. This implies that the cell to the right of κ contains and entry that is $> \overline{S_\omega} = \overline{\tilde{S}_\kappa}$, the equality following from a combination of (3.27), (3.29), and (3.23). As noted above, this guarantees that $\tilde{S}$ is a super semistandard tableau.

Finally, we prove that κ is the top-most inward corner cell in $\tilde{S}^O$ that contains the minimal entry in $\tilde{S}^O$. This is trivially true if $\overline{S_\zeta + c_\zeta} < \min\{\text{entries in } S^O\}$, again by remembering (3.23). Note that this inequality is in particular true at the very beginning of Algorithm S*. Because of the induction hypothesis (3.22), the only other case is $\overline{S_\zeta + c_\zeta} = \min\{\text{entries in } S^O\}$. Since we are *not* at the very beginning, we are allowed to assume that this last assertion of Lemma S* holds for S and ω. However, we already considered the case $\overline{S_\zeta + c_\zeta} = \min\{\text{entries in } S^O\}$ before (see (3.26)) and showed that the "jeu de taquin path" leading from ζ to κ has to stay above of the "jeu de taquin path" leading from γ to ω. Hence κ is strictly above ω. By induction hypothesis, ω was the top-most corner cell containing the minimal entry of S^O. So κ, which by (3.23) contains $\overline{S_\zeta + c_\zeta} = \min\{\text{entries in } S^O\}$ in $\tilde{S}$, is the top-most corner cell of $\tilde{S}$ containing $\min\{\text{entries in } \tilde{S}^O\}$ ($= \min\{\text{entries in } S^O\}$), as desired.

This completes the proof of the Lemma. $\square$

¿From Lemmas 1 and 2 it is abundantly clear that the Algorithms S and S* are inverses of each other. This finishes the bijective proof of (3.1).

4. More on modified jeu de taquin. In this section we explore the relations between our modified jeu de taquin in Algorithm S and a modified jeu de taquin in a paper of Goulden and Greene [7, sec. 3]. The main result of this section is an invariance property (Theorem SG) of modified jeu de taquin, which somehow is the analogue of the invariance of standard jeu de taquin (see [18; 16, Theorem 3.9.7]).

Super Schur functions are also defined for skew shapes λ/μ. For the purposes of this section, we assume that $\mathbf{x}$ and $\mathbf{y}$ are *doubly* infinite sequences of variables, $\mathbf{x} = (\dots, x_{-1}, x_0, x_1, \dots)$ and $\mathbf{y} = (\dots, y_{-1}, y_0, y_1, \dots)$. Then, the skew super Schur function $\mathcal{S}_{\lambda/\mu}(\mathbf{x}, \mathbf{y})$ is defined by

$$\mathcal{S}_{\lambda/\mu}(\mathbf{x}, \mathbf{y}) = \det_{1 \le i,j \le r} (\mathcal{S}_{\lambda_i - i - \mu_j + j}(\mathbf{x}, \mathbf{y})), \tag{4.1}$$

where the series $\mathcal{S}_m(\mathbf{x}, \mathbf{y})$ are determined by their generating function

$$\sum_{m \ge 0} \mathcal{S}_m(\mathbf{x}, \mathbf{y}) z^m = \prod_{i=-\infty}^{\infty} \frac{1 + y_i z}{1 - x_i z},$$

(note the difference to (1.2) in the range of the product). The combinatorial description (2.1) extends to skew super Schur functions. Namely, if we extend the definition of super semistandard tableau by saying that it is a filling of some skew shape with entries from the ordered alphabet $\dots < -1 < 0 < 1 < \dots < \overline{-1} < \overline{0} < \overline{1} < \dots$ satisfying (SST1) (but the shape ν replaced by ν/μ) and (SST2), then (again SST is short for 'super semistandard tableau')

$$\mathcal{S}_{\lambda/\mu}(\mathbf{x}, \mathbf{y}) = \sum_{\substack{S \text{ a SST} \\ \text{of shape } \lambda/\mu}} \prod_{i=-\infty}^{\infty} x_i^{\#i\text{'s in } S} y_i^{\#\bar{i}\text{'s in } S}. \tag{4.2}$$

As in the nonskew case (4.2) can be proven combinatorially by the Gessel–Viennot method of nonintersecting lattice paths, see [15, sec. 5; 4, Theorem 3.4].

Algorithm S was defined for super semistandard tableaux of a nonskew shape λ. However, everything can be defined and works fine also for a skew shape λ/μ. To be precise, it is readily checked that Algorithm S for a skew shape λ/μ proves the following alternative combinatorial desription for (skew) super Schur functions due to Goulden and Greene [7, Theorem 1.1] (here CRPP is short for column-strict reverse plane partition with integer entries)

$$\mathcal{S}_{\lambda/\mu}(\mathbf{x},\mathbf{y}) = \sum_{\substack{P \text{ a CRPP}\\ \text{of shape } \lambda/\mu}} \prod_{\rho\in\lambda/\mu} (x_{P_\rho} + y_{P_\rho+c_\rho}). \tag{4.3}$$

(In fact, we needed the restriction of a nonskew shape only at one place, namely, in the second paragraph after Example S, when showing that all the entries are at least 1 when arriving at (S3). But now we are considering alphabets that are unbounded above *and* below, so this step is superfluous in this context.) Goulden and Greene also construct a bijection for proving that the representations (4.2) and (4.3) for $\mathcal{S}_{\lambda/\mu}(\mathbf{x},\mathbf{y})$ agree. This bijection is based on a generalization of Bender and Knuth's involution [1, pp. 46/47; 16, pp. 152/153] that proves the symmetry of Schur functions combinatorially. In passing, they provide (without proof) an alternative description of this bijection, in terms of a jeu de taquin-like procedure. Though similar, this procedure is different from ours. To understand the relationship between the two procedures, we introduce a *nondeterministic* jeu de taquin procedure which contains both Algorithm S and Goulden and Greene's algorithm as special cases.

The following (nondeterministic) algorithm maps a super semistandard tableau S of shape λ/μ to an array A of shape λ/μ containing unbarred and barred positive integers such that

(A1) the multiset of entries in S and A agree,
(A2) if in A each barred entry, $\bar y$ say, contained in cell ρ, is replaced by $y - c_\rho$, then A becomes a column-strict reverse plane partition.

It should be observed that such an array A is exactly equivalent to a pair (P, U) of a column-strict reverse plane partition P and a (0–1)-tabloid U, the latter pairs occuring as output of Algorithm S. Given such an array A the corresponding pair (P, U) is obtained by defining P to be the column-strict reverse plane partition constructed in (A2) above, and by defining U to be the (0–1)-tabloid with 1 in each cell that contains a barred entry in A and with 0 in each cell that contains an unbarred entry in A. Clearly, this correspondence is a bijection.

Algorithm SG. The input for the algorithm is a super semistandard tableau S of shape λ/μ.

(SG0) Set $A := S$.

(SG1) Consider all pairs $(x_1, \bar y_1)$, where x_1 is an unbarred entry and $\bar y_1$ is a barred entry located in cell ρ_1, x_1 being the left neighbour of $\bar y_1$, and all pairs $(x_2, \bar y_2)$, where x_2 is an unbarred entry and $\bar y_2$ is a barred entry located in cell ρ_2, x_2 being the top neighbour of

$\bar{y}_2$. If for all such pairs there holds

$$x_1 \le y_1 - c_{\rho_1} \tag{4.4}$$

and

$$x_2 < y_2 - c_{\rho_2}, \tag{4.5}$$

then stop. The output of the algorithm is A. (Observe that (4.4) and (4.5) are required for A to satisfy property (A2) above.)

Otherwise, do one of the following two steps B or U.

B. Consider all violations of (4.4) or (4.5). Let $\bar{y}$ be the *top-most* among the *minimal barred* entries involved. (From Lemma 1, (2), it will follow that $\bar{y}$ is unique.) Let $\bar{y}$ be located in cell ρ. Then we have the following situation,

$$\begin{array}{cc} \cline{2-2} & \multicolumn{1}{|c|}{x_t} \\ \hline \multicolumn{1}{|c|}{x_l} & \multicolumn{1}{c|}{\bar{y}} \\ \hline \end{array}\ , \tag{4.6}$$

where at least one of $x_l > y - c_\rho$ and $x_t \ge y - c_\rho$ holds. (One of x_l or x_t is also allowed to be actually not there, or, in abuse of notation, to be a barred entry. In the latter case, by convention the corresponding inequality does not hold.) If $x_l > x_t$ (or if x_t is a barred entry) then do the move

$$\begin{array}{cc} \cline{2-2} & \multicolumn{1}{|c|}{x_t} \\ \hline \multicolumn{1}{|c|}{\bar{y}} & \multicolumn{1}{c|}{x_l} \\ \hline \end{array}\ . \tag{4.7}$$

If $x_l \le x_t$ (or if x_l is a barred entry) then do the move

$$\begin{array}{cc} \cline{2-2} & \multicolumn{1}{|c|}{\bar{y}} \\ \hline \multicolumn{1}{|c|}{x_l} & \multicolumn{1}{c|}{x_t} \\ \hline \end{array}\ . \tag{4.8}$$

U. Consider all violations of (4.4) or (4.5). Let x be the *right-most* among the *maximal unbarred* entries involved. (From Lemma 1, (1), it will follow that x is unique.) Let the cell to the right of x be ρ_r and the cell below of x be ρ_b. Then we have the following situation,

$$\begin{array}{cc} \hline \multicolumn{1}{|c|}{x} & \multicolumn{1}{c|}{\bar{y}_r} \\ \hline \multicolumn{1}{|c|}{\bar{y}_b} & \\ \cline{1-1} \end{array}\ , \tag{4.9}$$

where at least one of $x > y_r - c_{\rho_r}$ and $x \ge y_b - c_{\rho_b}$ holds. (One of $\bar{y}_r$ or $\bar{y}_b$ is also allowed to be actually not there, or, in abuse of notation, to be an unbarred entry. In the latter case, by convention the corresponding inequality does not hold.) If $\bar{y}_r \le \bar{y}_b$ (or if $\bar{y}_b$ is an unbarred entry) then do the move

$$\begin{array}{cc} \hline \multicolumn{1}{|c|}{\bar{y}_r} & \multicolumn{1}{c|}{x} \\ \hline \multicolumn{1}{|c|}{\bar{y}_b} & \\ \cline{1-1} \end{array}\ . \tag{4.10}$$

If $\bar{y}_r > \bar{y}_b$ (or if $\bar{y}_r$ is an unbarred entry) then do the move

$$\begin{array}{|c|c|}\hline \bar{y}_b & \bar{y}_r \\ \hline x & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}\ . \tag{4.11}$$

Let the new A be the array of shape λ/μ just obtained. Repeat (SG1). □

The output of the algorithm trivially satisfies condition (A1) above. It will follow from Lemma 1 that the output also satisfies (A2), see the Remark after Lemma 1.

It is an easy observation that when always choosing step B in Algorithm SG we basically do the same as in Algorithm S. The only difference is that in Algorithm SG we do not subtract the content from the barred entries and keep the barred entries as such, unlike in Algorithm S. The translation of an output of Algorithm SG into the corresponding output of Algorithm S was described in the paragraph preceding the description of Algorithm SG. Similarly, when always choosing step U in Algorithm SG we (basically) do the same as in Goulden and Greene's algorithm. So, roughly speaking, what we do in Algorithm S is to move the barred entries, one after the other, "into" the unbarred entries, while Goulden and Greene move the unbarred entries, one after the other, "into" the barred entries. Nevertheless, both procedures turn out to be equivalent, i.e. the final result is the same in both cases. In fact, the main result of this section (Theorem SG below) is that, regardless which choices we take in step (SG1), the final result will be the same. Therefore, though Goulden and Greene write that, unlike the standard jeu de taquin, their algorithm is deterministic and different sequences of moves may give different results, there is *some* freedom in choosing the order of moves. But there is not as much freedom as in standard jeu de taquin where there is complete freedom.

Theorem SG. *The output of Algorithm SG applied to some given super semistandard tableau S is independent of the choices between B and U that are taken in step (SG1).*

The proof of Theorem SG is based on the following two Lemmas.

Lemma 1. *After each application of* (SG1) *during Algorithm SG, the obtained array A has the following properties.*

(1) *Let x be an unbarred entry located (weakly) to the left and (weakly) above the unbarred entry y. Then $x \le y$. Also, each column is strictly increasing in the unbarred entries.*

(2) *Let $\bar{x}$ be a barred entry located (weakly) to the left and (weakly) above the barred entry $\bar{y}$. Then $\bar{x} \le \bar{y}$. Also, each row is strictly increasing in the barred entries.*

(3) *For all pairs $(\bar{x}_1, y_1)$, where $\bar{x}_1$ is a barred entry located in cell ρ_1 and y_1 is an unbarred entry, $\bar{x}_1$ being the left neighbour of y_1, there holds*

$$x_1 - c_{\rho_1} \le y_1. \tag{4.12}$$

For all pairs $(\bar{x}_2, y_2)$, where $\bar{x}_2$ is a barred entry located in cell ρ_2 and y_2 is an unbarred entry, $\bar{x}_2$ being the top neighbour of y_2, there holds

$$x_2 - c_{\rho_2} < y_2. \tag{4.13}$$

REMARK. Clearly, this implies that the output of Algorithm SG satisfies (A2). For, in addition to the properties (1)–(3) of the Lemma, the output satisfies also (4.4) and (4.5) since otherwise the step (SG1) was not finished.

PROOF. We prove the Lemma by induction on the number of applications of (SG1). The claim of the Lemma is certainly true at the very beginning since we start with a super semistandard tableau.

For the induction step a careful case-by-case analysis has to be performed. We consider just one case out of four (two cases each for steps B and U, depending on whether moving horizontally or vertically), the others are similar.

Suppose that by performing step U, (4.11), an unbarred entry, x say, is moved down one row. Figure 5 displays the generic situation before and after the move. Here, m, n, s, t can be any nonnegative integers.

$$\begin{array}{r|c|l}
 & \text{cell } \rho & \\
\hline
\dots \bar{b}\, x_{-m} \dots x_{-2}\, x_{-1} & x & x_1\, x_2 \dots x_s\, \bar{z} \dots \\
\hline
\dots a\, \bar{y}_{-n} \dots \bar{y}_{-2}\, \bar{y}_{-1} & \bar{y} & \bar{y}_1\, \bar{y}_2 \dots \bar{y}_t\, u \dots \\
\hline
 & \text{cell } \omega &
\end{array}
\qquad
\begin{array}{r|c|l}
 & \text{cell } \rho & \\
\hline
\dots \bar{b}\, x_{-m} \dots x_{-2}\, x_{-1} & \bar{y} & x_1\, x_2 \dots x_s\, \bar{z} \dots \\
\hline
\dots a\, \bar{y}_{-n} \dots \bar{y}_{-2}\, \bar{y}_{-1} & x & \bar{y}_1\, \bar{y}_2 \dots \bar{y}_t\, u \dots \\
\hline
 & \text{cell } \omega &
\end{array}$$

a. before the move b. after the move

Figure 5

The pictures have to be taken symbolically in the sense that all combinations of $n > m$ or $n \le m$, and $s > t$ or $s \le t$ are possible.

By definition of step U, x is the right-most among the maximal unbarred entries that violate (4.4) or (4.5) in Figure 5.a. Moreover, since we moved down by (4.11), we have (for the definition of ω see Figure 5)

$$x \ge y - c_\omega. \tag{4.14}$$

Of course, by induction hypothesis, nothing has to be shown for entries that did not move. In fact, it suffices to prove

$$\text{ad (1):} \quad a \le x \le u \tag{4.15}$$

$$\text{ad (2):} \quad \bar{b} < \bar{y} < \bar{z} \tag{4.16}$$

$$\text{ad (3):} \quad y - c_\rho < x,\ y - c_\rho \le x_1 \text{ and } y_{-1} - c_{\omega_{-1}} \le x, \tag{4.17}$$

where we assume that $\bar{y}_i$ is located in cell ω_i, $i = -n, \dots, -1, 1, \dots, t$.

In order to prove $a \le x$, we distinguish between two cases. First suppose $a > y_{-n} - c_{\omega_{-n}}$. This is a violation of (4.4). Since x is the right-most among the maximal unbarred entries that violate (4.4) or (4.5) in Figure 5.a, we must have $x \ge a$. On the other hand, suppose $a \le y_{-n} - c_{\omega_{-n}}$. Then, by induction hypothesis (2) (applied to Figure 5.a of course), and by (4.14), we have

$$a \le y_{-n} - c_{\omega_{-n}} \le y_{-n+1} - c_{\omega_{-n+1}} \le \cdots \le y_{-1} - c_{\omega_{-1}} \le y - c_\omega \le x.$$

The inequality $x \le u$ follows from the induction hypothesis (1).

For proving $\bar{b} < \bar{y}$, we first note that by induction hypothesis (2) we have $\bar{b} \le \bar{y}$. So, we only have to show that $\bar{b} \ne \bar{y}$. By way of contradiction, let us assume $\bar{b} = \bar{y}$. Then, by induction hypothesis (2), it is impossible that there are any barred entries to the left of $\bar{y}$ in Figure 5.a. This means that $n = 0$, and that a is the left neighbour of $\bar{y}$ in Figure 5.a. At the very beginning the entry $\bar{b}$ was strictly above and weakly to the right of the entry $\bar{y}$, because otherwise $\bar{b}$ and $\bar{y}$ would have been in the same row at some stage, contradicting induction hypothesis (2). For similar reasons it is impossible that a rectangle determined by $\bar{b}$ as its top-left corner and $\bar{y}$ as its bottom-right corner contained another barred entry at any time. Finally, because of the argument visualized in Figure 3, the path of entry $\bar{y}$ always stays below the path of entry $\bar{b}$. Therefore $\bar{b}$ must have passed all the way through from cell ρ to the cell that it is now occupying in Figure 5. Moreover, $x_{-m}, \dots, x_{-1}, x$ and a did not move after $\bar{b}$ passed through. But then, at some point, we must have considered the situation

x	$\bar{b}$
a	

before. By induction hypothesis (1), this implies $x < a$. This contradicts $x \ge a$ which was already established.

Next we show $\bar{y} < \bar{z}$. Let $\bar{z}$ be located in cell ζ. If $s = 0$, i.e. if $\bar{z}$ is the right neighbour of x in Figure 5.a, then $\bar{y} < \bar{z}$ holds by definition of step U, (4.11). If $s > 0$, then we must have $x_s \le z - c_\zeta$. For, otherwise (4.4) would be violated. Besides, by induction hypothesis (1) we have $x \le x_s$, and so x would not be the right-most among the maximal unbarred entries that violate (4.4) or (4.5) in Figure 5.a. Combining this with (4.14), we conclude $y - c_\omega \le x \le x_s \le z - c_\zeta$ and hence $y \le z - s - 2 < z$.

All three inequalities in (4.17) follow directly from (4.14), $x \le x_1$, and $\bar{y}_{-1} < \bar{y}$, the latter two being due to the induction hypothesis (1), (2). $\square$

Lemma 2. *Suppose that A is an array satisfying (1)–(3) in Lemma 1. Furthermore suppose that the right-most among the maximal unbarred entries that violate (4.4) or (4.5) is not the left or top neighbour of the top-most among the minimal barred entries that violate (4.4) or (4.5). Then the steps* U *and* B *commute,*

$$\mathrm{U}(\mathrm{B}(A)) = \mathrm{B}(\mathrm{U}(A)).$$

PROOF. This is established by a straight-forward case-by-case analysis, utilizing Lemma 1. $\square$

Finally we are in the position to prove Theorem SG.

PROOF OF THEOREM SG. Suppose that during Algorithm SG we applied the sequence $W(\mathrm{U},\mathrm{B})$ of steps U and B to the initial semistandard tableau S. One should think of $W(\mathrm{U},\mathrm{B})$ as a word in U's and B's. Let the length of $W(\mathrm{U},\mathrm{B})$ be L. We shall show that

$$W(\mathrm{U},\mathrm{B})(S) = \mathrm{B}^L(S), \tag{4.18}$$

i.e. that the result is the same when we apply step B all the time (which is basically the same as Algorithm S, as we already noted). Clearly this would suffice to prove the Theorem.

We establish (4.18) by successively changing U's in $W(\mathrm{U},\mathrm{B})$ into B's and by applying the commutation rule in Lemma 2. First we claim that there is a word $W_1(\mathrm{U},\mathrm{B})$ such that

$$W(\mathrm{U},\mathrm{B})(S)=\mathrm{B}\,W_1(\mathrm{U},\mathrm{B})(S). \tag{4.19}$$

This is trivially true if $W(\mathrm{U},\mathrm{B})$ already has the form $\mathrm{B}\,W_1(\mathrm{U},\mathrm{B})$. So let us assume that $W(\mathrm{U},\mathrm{B})$ has the form $\mathrm{U}\,W_1(\mathrm{U},\mathrm{B})$. Consider the array $W_1(\mathrm{U},\mathrm{B})(S)$, the array obtained before the final step. Since $W(\mathrm{U},\mathrm{B})(S)=\mathrm{U}\,W_1(\mathrm{U},\mathrm{B})(S)$ satisfies (4.4) and (4.5), it is obvious that the right-most among the maximal unbarred entries that violate (4.4) or (4.5) in $W_1(\mathrm{U},\mathrm{B})(S)$ is the left or top neighbour of the top-most among the minimal barred entries that violate (4.4) or (4.5) in $W_1(\mathrm{U},\mathrm{B})(S)$. Hence, it does not make any difference whether we apply U or B to $W_1(\mathrm{U},\mathrm{B})(S)$. This settles (4.19) also in this case.

Now we use the commutation rule in Lemma 2 to move the left B in the right-hand side of (4.19) inside the word $W_1(\mathrm{U},\mathrm{B})$. The only problem arises, when we arrive at $W_2(\mathrm{U},\mathrm{B})\,\mathrm{B}\,\mathrm{U}\,W_3(\mathrm{U},\mathrm{B})(S)$, where $W_3(\mathrm{U},\mathrm{B})(S)$ is an array in which the right-most among the maximal unbarred entries that violate (4.4) or (4.5) is the left or top neighbour of the top-most among the minimal barred entries that violate (4.4) or (4.5). In this case Lemma 2 does not apply. But then we have $\mathrm{U}\,W_3(\mathrm{U},\mathrm{B})(S)=\mathrm{B}\,W_3(\mathrm{U},\mathrm{B})(S)$, and therefore

$$W(\mathrm{U},\mathrm{B})(S)=W_2(\mathrm{U},\mathrm{B})\,\mathrm{B}\,\mathrm{B}\,W_3(\mathrm{U},\mathrm{B}). \tag{4.20}$$

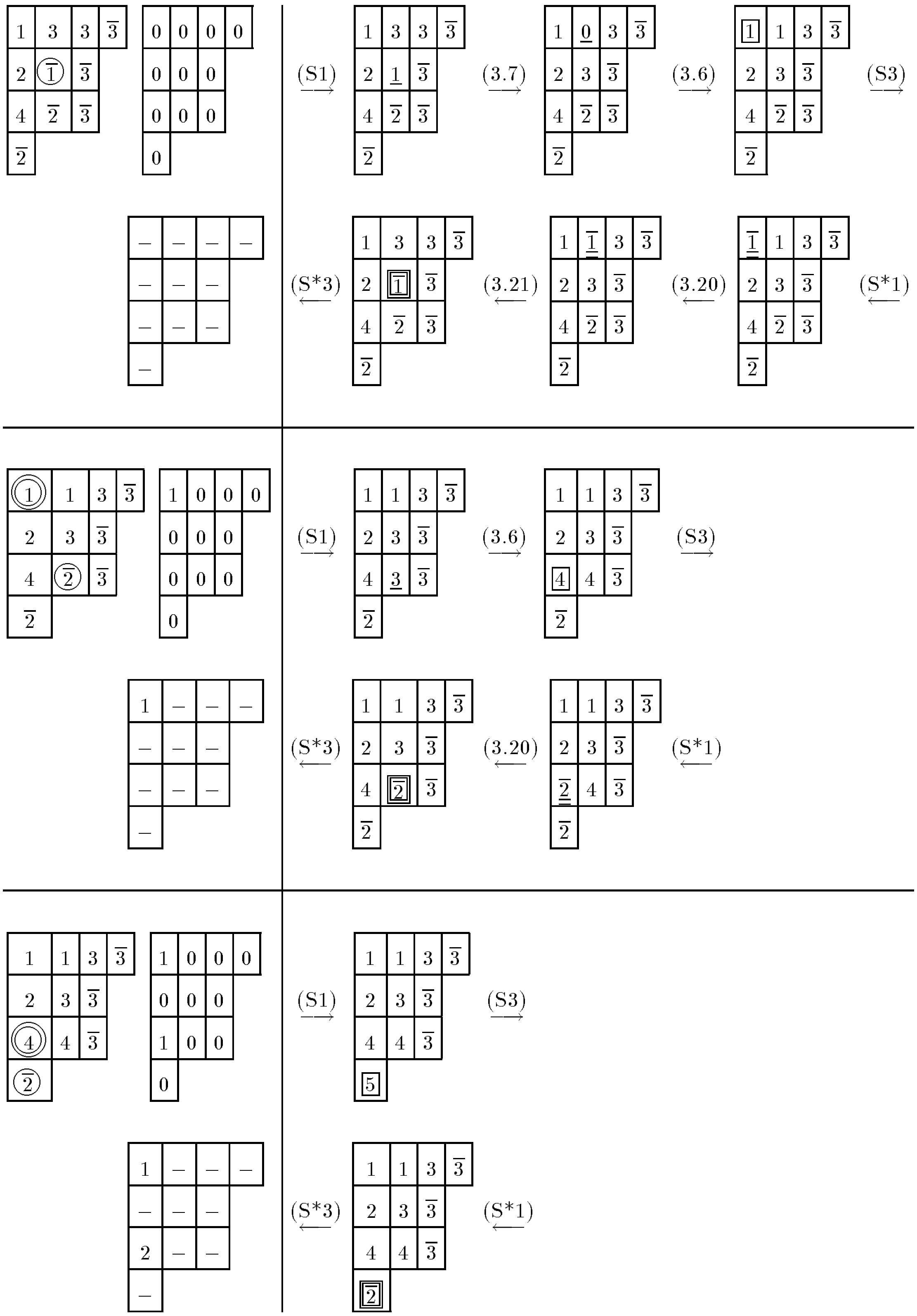
We continue by moving this "new" B inside the word $W_3(\mathrm{U},\mathrm{B})$, until we arrive at

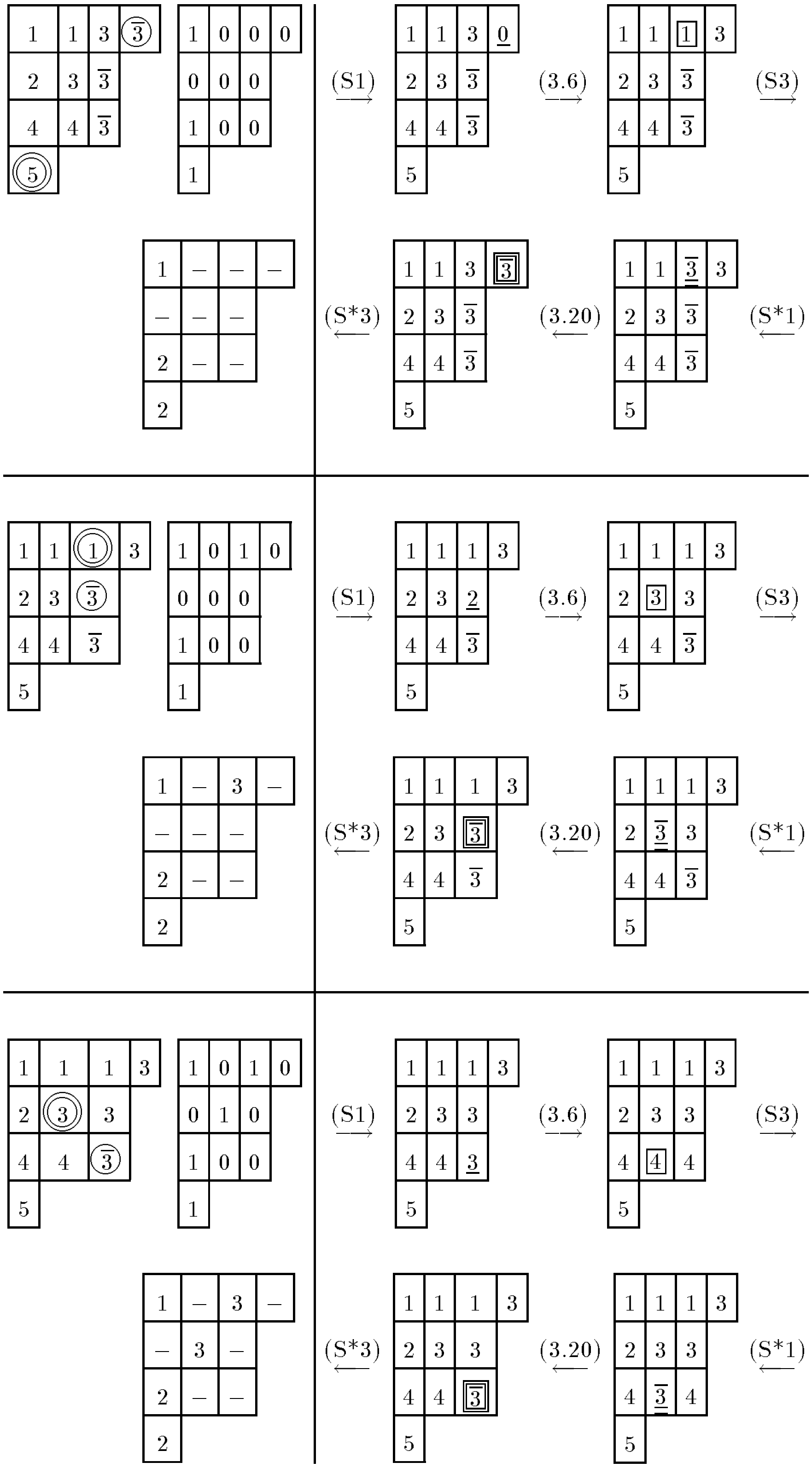
$$W(\mathrm{U},\mathrm{B})(S)=\big(W_4(\mathrm{U},\mathrm{B})\,\mathrm{B}\big)(S), \tag{4.21}$$

for some word $W_4(\mathrm{U},\mathrm{B})$. By iterating the process (4.19)-(4.20)-(4.21) we finally obtain (4.18). $\square$

Acknowledgement. This work was carried out while the author visited the University of California at San Diego. He thanks the University of California and in particular Adriano Garsia for making this visit possible. Besides, he is indebted to Adriano Garsia for drawing his attention to the problem of finding "nice" combinatorial proofs of hook formulas.

Appendix. The appendix contains a complete example for Algorithms S and S* for $\lambda=(4,3,3,1)$, setting up a mapping between the two sides of Figure 2. See the specific descriptions given in Examples 1 and 2 of how to read the following tables.


(S1)
(3.7)
(3.6)
(S3)
(S*3)
(3.21)
(3.20)
(S*1)
(S1)
(3.6)
(S3)
(S*3)
(3.20)
(S*1)
(S1)
(S3)
(S*3)
(S*1)


(S1)
(3.6)
(S3)
(S*3)
(3.20)
(S*1)

1	1	1	3
2	3	3	
4	4	4	
5			

1	0	1	0
0	1	0	
1	1	0	
1			

1	–	3	–
–	3	–	
2	3	–	
2			

References

1. E. A. Bender and D. E. Knuth, *Enumeration of plane partitions*, J. Combin. Theory A **13** (1972), 40—54.
2. A. Berele and A. Regev, *Hook Young diagrams with applications to combinatorics and representations of Lie superalgebras*, Adv. in Math. **44** (1987), 118–175.
3. A. Berele and J. B. Remmel, *Hook flag characters and their combinatorics*, J. Pure Appl. Algebra **35** (1985), 225–245.
4. F. Brenti, *Determinants of super-Schur functions, lattice paths, and dotted plane partitions*, Adv. in Math. **98** (1993), 27–64.
5. I. M. Gessel and X. Viennot, *Binomial determinants, paths, and hook length formulae*, Adv. in Math. **58** (1985), 300—321.
6. I. M. Gessel and X. Viennot, *Determinants, paths, and plane partitions*, preprint (1989).
7. I. Goulden and C. Greene, *A new tableau representation for supersymmetric Schur functions*, J. Algebra **170** (1994), 687–703.
8. A. P. Hillman and R. M. Grassl, *Reverse plane partitions and tableau hook numbers*, J. Combin. Theory Ser. A **21** (1976), 216–221.
9. V. G. Kac, *Lie superalgebras*, Adv. in Math. **26** (1977), 8–96.
10. V. G. Kac, *Characters of typical representations of classical Lie superalgebras*, Comm. Algebra **5** (1977), 889–897.
11. V. G. Kac, *Representations of classical Lie superalgebras*, Differential geometrical methods in mathematical physics II, Lecture Notes in Math., vol. 676, Springer–Verlag, Berlin, 1978, pp. 597–626.
12. C. Krattenthaler, *An involution principle-free bijective proof of Stanley's hook-content formula*, preprint.
13. I. G. Macdonald, *Symmetric Functions and Hall Polynomials*, second edition, Oxford University Press, New York/London, 1995.
14. N. Metropolis, G. Nicoletti and G.-C. Rota, *A new class of symmetric functions*, Math. Anal. Appl. Part B, Adv. in Math. Suppl. Stud. **7** (1981), 563–575.
15. J. B. Remmel, *The combinatorics of (k,l)-hook Schur functions*, Combinatorics and algebra (C. Greene, ed.), Contemporary Math., vol. 34, Amer. Math. Soc., Providence, R.I., 1984, pp. 253–287.
16. B. E. Sagan, *The symmetric group*, Wadsworth & Brooks/Cole, Pacific Grove, California, 1991.
17. M. Scheunert, *The theory of Lie superalgebras*, Lecture Notes in Math., vol. 716, Springer–Verlag, Berlin, 1979.
18. 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.

INSTITUT FÜR MATHEMATIK DER UNIVERSITÄT WIEN, STRUDLHOFGASSE 4, A-1090 WIEN, AUSTRIA.