Available online at www.sciencedirect.com

SCIENCE DIRECT

Journal of Combinatorial Theory, Series A 102 (2003) 110–135

Journal of Combinatorial Theory Series A

http://www.elsevier.com/locate/jcta

Generalization of Schensted insertion algorithm to the cases of hooks and semi-shuffles ☆

Mikhail Kogan

Department of Mathematics, Northeastern University, Boston, MA 02115, USA

Received 6 June 2002

Abstract

Given an rc-graph R of permutation w and an rc-graph Y of a permutation v, we provide an insertion algorithm which defines an rc-graph $R \leftarrow Y$ in the case when v is a shuffle with the descent at r and w has no descents greater than r or in the case when v is a shuffle whose shape is a hook. This algorithm gives a combinatorial rule for computing the generalized Littlewood–Richardson coefficients c^u_{wv} in the two cases mentioned above.

1. Introduction

Rc-graphs were originally introduced by Fomin and Kirillov [4] as graphical representations of reduced compatible sequences of Billey et al. [3]. They are explicit combinatorial objects which encode monomials of Schubert polynomials. Rc-graphs proved to be very useful for providing combinatorial rules for computing certain generalized Littlewood–Richardson (or just LR) coefficients (see [1,8,9]). In this paper we extend these results to more general cases.

Denote by $\mathfrak{S}_w$ the Schubert polynomial of the permutation $w \in S_\infty$. Then the generalized LR coefficients c^u_{wv} for $u, v, w \in S_\infty$ are defined by

$$\mathfrak{S}_w \cdot \mathfrak{S}_v = \sum_u c^u_{wv} \mathfrak{S}_u.$$

(If u, v, w are shuffles, also called grassmannian permutations, with the descents at r, the coefficients c^u_{wv} are just the LR coefficients.) The generalized LR coefficients are

☆ The author is supported by NSF Postdoctoral Fellowship.
E-mail address: misha@neu.edu.

0097-3165/03/$ - see front matter
doi:10.1016/S0097-3165(03)00022-0

also the structure constants of the cohomology ring of the flag manifold for the basis given by Schubert varieties. In particular, they count the number of points in certain intersections of algebraic varieties. Therefore, all c^u_{wv} are non-negative integers. There are many totally positive rules for computing the LR coefficients (see [6] for further references), but there is no known totally positive rule for the generalized LR coefficients. (By a totally positive rule we understand a construction of an explicit combinatorial set for each triple (u, v, w), such that c^u_{wv} is equal to the number of elements in this set.)

In certain cases (see [2,7]) a totally positive rule can by given by equating the generalized LR coefficients to the LR coefficients. In other cases, such as the Pieri formula (see [11,16]), a totally positive rule is given in terms of certain paths in the Bruhat order.

Yet another approach to produce a totally positive rule, adopted in [1,8,9] and in this paper, is to generalize the Schensted insertion algorithm [15] to rc-graphs. The outline of the rule, which we believe will be eventually generalized to the most general case, is the following. An algorithm is constructed. This algorithm inserts an rc-graph Y of v into an rc-graph R of w to produce an rc-graph $R \leftarrow Y$. Then, for a fixed rc-graph U of u, c^u_{wv} is the number of tuples (R, Y) with $U = R \leftarrow Y$. We present such an algorithm in the cases when v is an r-shuffle and w is an r-semi-shuffle or when v is a shuffle whose shape is a hook. (An r-shuffle is a shuffle with the descent at r, an r-semi-shuffle is a permutation with no descents greater than r.)

The first such algorithm was constructed by Bergeron and Billey [1] to prove Monk's formula (the case when v is a simple transposition). The author [8] showed that their algorithm works in the case when w is an r-semi-shuffle and v is an r-shuffle. A modified algorithm was constructed by Kumar and the author [9] to give another proof of the row case of the Pieri formula (the case when v is an r-shuffle whose shape is a row). Kumar [10] constructed an analogous algorithm for the column case of the Pieri formula. The analogy between the two algorithms in [9,10] is similar to the analogy between the row and column Schensted insertion algorithms for Young tableaux.

This paper presents an algorithm which works in all mentioned above cases as well as in the case when v is a shuffle whose shape is a hook. (A rule in this case written in terms of r-Bruhat paths was originally constructed by Sottile [16].) In many cases (see [1,8,9]) the algorithm can be simplified. But the algorithm is new for the case of hooks when no simplification of the algorithm is known.

Using the new insertion algorithm we also provide a rule for computing generalized LR coefficients in the cases mentioned above using r-Bruhat paths. This rule can be thought of as a generalized RSK correspondence. In the case when the shape of v is a hook, it is just a restatement of the formula from [16]. In the case when v is an r-shuffle and w is an r-semi-shuffle it is a new result.

All the generalized LR coefficients computed in this paper can be found using previously known methods. The novelty of the results in this paper is in the methods used for computations, namely, the new algorithm for the hook case and the new generalization of the RSK correspondence in the semi-shuffle case. Both of these approaches can potentially lead to a complete solution of the problem.

The paper is organized as follows. Section 2 introduces most of the notations and definitions and contains the statements of the main results in Theorems 2.2–2.4. Theorem 2.2 states that the algorithm defined in Section 3 works, it is proved in Section 4. Theorem 2.3 states that the inverse algorithm defined in Section 5 works, it is given without a proof, since the proof is very similar to the proof of Theorem 2.2. Theorem 2.4 gives a rule for computing certain generalized LR coefficients in terms of r-Bruhat paths. Section 6 contains examples of the algorithm, this section should be read together with Section 3 to understand how the algorithm works.

2. Notation, definitions and main results

2.1. Permutations. Let S_n be the group of permutations $w = (w(1), \ldots, w(n))$ and let $S_\infty = \bigcup_n S_n$ be the group of permutations on $\mathbb{N}$ which fix all but finitely many integers. For $1 \leqslant i < j$, denote by t_{ij} the transposition which exchanges i and j. The simple transpositions $s_i = t_{i,i+1}$ for $1 \leqslant i \leqslant n-1$ generate S_n.

A word $i_1 \ldots i_l$ in the alphabet $[1, 2, \ldots]$ is a reduced word of $w \in S_\infty$, if $w = s_{i_1} \ldots s_{i_l}$ and l is minimal. The length $l(w)$ of w is set to be l. The longest permutation w_0^n of the group S_n is given by $w_0^n(i) = n + 1 - i$ for $i \leqslant n$.

A permutation $w \in S_\infty$ is an *r-shuffle* if $w(i) < w(i+1)$ for $i \neq r$. It is an *r-semi-shuffle* if $w(i) < w(i+1)$ for $i > r$. To each shuffle we associate a partition $\lambda = (\lambda_1 \geqslant \cdots \geqslant \lambda_{r'} > 0)$ given by $\lambda_j = w(r+1-j) - r - 1 + j$ for $j \leqslant r$ where $\lambda_{r'+1} = 0$. Then w is uniquely determined by λ and r and we write $w = v(\lambda, r)$. A partition λ is *a row* if $\lambda = (\lambda_1)$, it is *a column* if $\lambda = (1, \ldots, 1)$, and it is *a hook* if $\lambda = (\lambda_1, 1, \ldots, 1)$.

2.2. Rc-graphs. Let W_0^n be the reduced word $(n-1 \ldots 1 \ldots n-1 \;\; n-2 \;\; n-1)$ of w_0^n. A subword R of W_0^n is called *a graph*. Each graph $R = i_1 \ldots i_m$ defines a permutation $w(R) = s_{i_1} \ldots s_{i_m}$. If R is a reduced word of $w(R)$, it is called *an rc-graph* of $w(R)$.[1] Note that two different subwords of W_0^n which produce the same words are two different graphs. For example, if $n = 3$, $w = s_2$, then $W_0^3 = 212$ has two different subwords whose permutation is w, namely the subword 2 placed at the first or third slot. The set of all rc-graphs of w is denoted by $\mathscr{RC}(w)$.

We think of graphs using the following pictorial presentation. Think of W_0^n as a triangular set of crossings shown in the first picture of Fig. 1 for $n = 5$. Each crossing is labelled by a letter from the alphabet $[1, \ldots, n-1]$. To get back W_0^n we read those labels from top to bottom row, from right to left in each row. Then each subword R of W_0^n is presented as a subset of the crossings for W_0^n. Two illustrations are provided in Fig. 1 where the second picture corresponds to the subword 2323, while the third picture corresponds to the subword 4132.

Connect the crossings of R by strands which intersect at the places where there is a crossing and do not intersect otherwise. For illustration see Fig. 2 where the graphs correspond to the graphs from Fig. 1. Notice that when we draw pictures of graphs

[1] Throughout the paper we make the distinction between graphs and rc-graphs. Namely, whenever R is called a *graph* it means it might or might not be a reduced word of $w(R)$. If R is called an *rc-graph* it means it is a reduced word of $w(R)$.

	1	2	3	4	5
1	$+_1$	$+_2$	$+_3$	$+_4$	
2	$+_2$	$+_3$	$+_4$		
3	$+_3$	$+_4$			
4	$+_4$				
5					

	1	2	3	4	5
1	•	$+_2$	•	•	
2	$+_2$	$+_3$	•		
3	$+_3$	•			
4	•				
5					

	1	2	3	4	5
1	$+_1$	•	•	$+_4$	
2	$+_2$	$+_3$	•		
3	•	•			
4	•				
5					

Fig. 1. Examples of graphs.

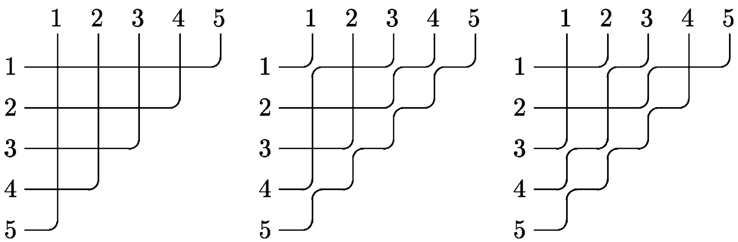


Fig. 2. Examples of graphs.

we omit those parts of graphs which have no crossings. So, graphs from Figs. 1 and 2 can be extended down and to the right by non-intersecting strands.

It is easy to see that for each graph R and $i \in \mathbb{N}$, $w(R)(i)$ is given by the column where the strand, which starts at row i, ends. (Instead of referring to a strand as "a strand, which starts at row i", we will say "strand i". So, the above statement transforms to: strand i ends in column $w(R)(i)$.) This immediately leads to

(2.1) If a graph R is constructed out of another graph R' by adding or removing a crossing of strands c and d then $w(R) = w(R')t_{cd}$.

Denote by $|R|$ the length of the corresponding subword, or, in other words, the number of crossings in R. Clearly, $|R| \geqslant l(w(R))$. The following two statements are easy to check. The first one originally appeared in [4], while the second one is a restatement of the definition.

(2.2) A graph R is an rc-graph if and only if no two strands intersect twice.
(2.3) A graph R is an rc-graph if and only if $|R| = l(w(R))$.

For example, the first and third graphs of Fig. 2 are rc-graphs, since both graphs have no double crossings and for both $|R| = l(w(R))$. But the second graph is not an rc-graph, since strands 3 and 4 intersect twice, or $|R| = 4 > l(w(R)) = 2$.

Given a graph R, let $R_{\leqslant \ell}$ be the graph which coincides with R below or at row ℓ and has no crossings above row ℓ. Let R_ℓ be the graph which coincides with R at row ℓ and has no crossings outside row ℓ.

Given two graphs R, S the union $R \cup S$ is defined to be the graph, which contains crossings of both R and S. If R lies above row ℓ, while S lies at or below row ℓ, then it is easy to see $w(R \cup S) = w(R)w(S)$.

"*Place* (i,j)" of a graph R will refer to either crossing or non-crossing of strands in row i and column j. For example, in the second graph from Fig. 2 strands intersect at place (2,1), but do not intersect at place (3,2). We refer to those strands which intersect or do not intersect at place (i,j) as strands, which pass place (i,j). For instance, strands 2, 4 pass places (2,2) and (1,3) in the third graph of Fig. 2. For two strands a and b let $a \boxplus b$ be the set of rows where strand a intersects strand b horizontally. Then $\ell \in a \boxplus b$ means that strand a intersects strand b in row ℓ and strand a is the horizontal strand of the crossing. For example, $3 \in 3 \boxplus 4$ and $1 \in 4 \boxplus 3$ for the second graph from Fig. 2.

2.3. Schubert polynomials. For detailed discussions of Schubert polynomials $\mathfrak{S}_w$ we refer the reader to [13] or [14]. The only property of Schubert polynomials used in this paper is stated in Theorem 2.1, proved in [3,5]. So, for purposes of this paper, we treat Theorem 2.1 as a definition.

For an rc-graph R define $x^R = x_1^{|R_1|} x_2^{|R_2|} \ldots$ (recall that in our notations $|R_i|$ is the number of crossings of R in row i).

Theorem 2.1. *For* $w \in S_\infty$,

$$\mathfrak{S}_w = \sum_{R \in \mathcal{RC}(w)} x^R.$$

If w is a shuffle $v(\lambda, r)$, then the Schubert polynomial $\mathfrak{S}_w$ is known to be equal to the Schur polynomial $S_\lambda(x_1, \ldots, x_r)$ (for a definition of Schur polynomials see [12]).

Schubert polynomials $\mathfrak{S}_w$ for all n-semi-shuffles form a basis for $\mathbb{C}[x_1, \ldots, x_n]$. Hence, for $u, v, w \in S_\infty$, we can uniquely define the generalized LR coefficients c_{wv}^u by

$$\mathfrak{S}_w \cdot \mathfrak{S}_v = \sum_u c_{wv}^u \mathfrak{S}_u. \tag{2.4}$$

2.4. Tableaux. To a partition $\lambda = (\lambda_1 \geqslant \cdots \geqslant \lambda_r > 0)$ associate a Young diagram, which is given by λ_i boxes in row i. If $m = \sum_{i=1}^r \lambda_i$, label the boxes of the Young diagram by integers 1 to m starting with the bottom row going up and going from left to right in each row as shown in the first picture of Fig. 3 for $\lambda = (3, 3, 1)$.

Given a Young diagram D, *a tableau* of shape D is a filling of boxes of the diagram D by elements of certain alphabets (we will talk later about the types of tableaux we consider). For any tableau U, produce the word of U by reading the content of the boxes 1 through m, denote it by $word(U)$. For example, the word of the second tableau of Fig. 3 is 4356145. Set $|U|$ to be the number of boxes of U.

Sometimes we consider only *partially filled tableaux*: we say that a tableau is *filled up to* i when boxes 1 to i are filled and the other boxes are empty.

A tableau filled with positive integers is *row* (*column*) *strict* if the numbers increase (do not decrease) from left to right and do not decrease (increase)

5	6	7
2	3	4
1		

1	4	5
3	5	6
4		

14	26	27
25	36	37
35		

Fig. 3. A Young diagram and two tableaux.

from top to bottom. For example, the second tableau of Fig. 3 is row and column strict.

A tableau T filled by tuples (ab) is called *a tableau of transpositions*. Given a shuffle $v(\lambda, r)$, T is *a tableau of transpositions of* $v(\lambda, r)$ if its shape is λ and it is filled by tuples (ab) with $a \leqslant r < b$. For example, see the third picture of Fig. 3 where $\lambda = (3, 3, 1)$ and $r = 3$. For a tableau of transpositions T, let its word $word(T)$ be $(a_1 b_1) \dots (a_m b_m)$. Define the permutations

$$w_i(T) = t_{a_1 b_1} \dots t_{a_i b_i} \quad \text{for } 1 \leqslant i \leqslant m, \text{ and } w(T) = w_m(T).$$

For $w \in S_\infty$, we say that a tableau of transpositions T of $v(\lambda, r)$ is *an r-Bruhat path of* w, if for $1 \leqslant i \leqslant m$

$$l(ww_i(T)) = l(w) + i. \tag{2.5}$$

If T is filled up to j, we say that T is an r-Bruhat path of w if (2.5) holds for $1 \leqslant i \leqslant j$. For a discussion of r-Bruhat paths see [2], where any sequence $(a_1 b_1) \dots (a_m b_m)$, satisfying $l(wt_{a_1 b_1} \dots t_{a_i b_i}) = l(w) + i$ and $a_i \leqslant r < b_i$ for all i, is an r-Bruhat path. For us it is convenient to think about $(a_i b_i)$ as entries of tableaux.

We say that a triple (w, R, T) consisting of a permutation w, an rc-graph R and a tableau of transpositions T is an *r-Bruhat package* if $ww(T) = w(R)$ and T is an r-Bruhat path of w.

Associate to each permutation w and each tableau of transpositions T of $v(\lambda, r)$ another tableau $E(w, T)$ of the same shape. Fill box i of $E(w, T)$ with $ww_i(T)(b_i)$. (If T is filled up to j, then $E(w, T)$ will be filled up to j.) For example, if $w = (1, 3, 4, 2, 5, 6, \dots)$ and T is the third tableaux from Fig. 3, then the second tableau from Fig. 3 is $E(w, T)$.

2.5. Main results. We are now ready to state our main results. Given $R \in \mathcal{RC}(w)$ and $Y \in \mathcal{RC}(v(\lambda, r))$ satisfying certain conditions, in Section 3 we present an insertion algorithm which defines a graph $R \leftarrow Y$ together with a tableau of transpositions $T(R, Y)$ of $v(\lambda, r)$.

Theorem 2.2. *Let $w, v(\lambda, r) \in S_\infty$ satisfy one the following*:

(2.6) *w is an r-semi-shuffle*,
(2.7) *λ is a hook*.

Let $R \in \mathcal{RC}(w)$ and $Y \in \mathcal{RC}(v(\lambda, r))$. Then $U = R \leftarrow Y$ is an rc-graph and

(2.8) *$E(w, T(R, Y))$ is a row and column strict tableau,*
(2.9) *$(w, U, T(R, Y))$ is an r-Bruhat package,*
(2.10) $x^U = x^R x^Y$.

In Section 5 we describe an inverse insertion algorithm that, given a certain rc-graph U and a tableau of transpositions T, constructs rc-graphs $R = U \rightarrow T$ and $Y(U, T)$. The following theorem is given without a proof. Proving it amounts to showing that the statement about the inverse insertion algorithm analogous to Theorem 2.2 holds. It can be done similarly to the proof of Theorem 2.2.

Theorem 2.3. *Let $w, v(\lambda, r) \in S_\infty$ satisfy* (2.6) *or* (2.7). *Define the following sets*

- $\mathcal{L}_{w,v(\lambda,r)}$: *Pairs (R, Y) of rc-graphs of w and $v(\lambda, r)$;*
- $\mathcal{R}_{w,v(\lambda,r)}$: *Pairs (U, T) of an rc-graph U of a permutation u and a tableau of transpositions T of $v(\lambda, r)$ such that $E(w, T)$ is a row and column strict tableau and (w, U, T) is an r-Bruhat package.*

Then the maps given by the insertion algorithm and the inverse insertion algorithms $\mathcal{L}_{w,v(\lambda,r)} \rightarrow \mathcal{R}_{w,v(\lambda,r)}$ and $\mathcal{R}_{w,v(\lambda,r)} \rightarrow \mathcal{L}_{w,v(\lambda,r)}$ are inverses of each other which define a bijection between these two sets. Moreover, this bijections preserves the monomials associated to the pairs ($x^R x^Y$ for (R, Y) and x^U for (U, T)).

Since the set $\mathcal{L}_{w,v(\lambda,r)}$ indexes the monomials of the left-hand side of Eq. (2.4). The next theorem is an immediate corollary of Theorems 2.1 and 2.3.

Theorem 2.4. *Assume $w, u, v(\lambda, r) \in S_\infty$ satisfy* (2.6) *or* (2.7). *Then $c^u_{wv(\lambda,r)}$ is equal to the number of tableaux of transpositions T of $v(\lambda, r)$ such that T is an r-Bruhat path of w, $E(w, T)$ is row and column strict and $ww(T) = u$.*

Let us restate Theorem 2.4 in the case when the shape of v is a hook in the form it appeared in [16]. Given w and a tableau of transpositions T of v, define $w^{(i)} = ww_i(T)$. If the shape of v is a hook $(p, 1^{q-1})$ then $E(w, T)$ is row and column strict if and only if

$$w^{(1)}(b_1) > \cdots > w^{(p)}(b_p) \quad \text{and} \quad w^{(p)}(b_p) < \cdots < w^{(m)}(b_m), \tag{2.11}$$

where $m = l(v) = p + q - 1$. Using the fact that t_{ab} and $t_{a'b'}$ commute as long as a, b, a', b' are distinct, it can be shown that there is a one to one correspondence between r-Bruhat paths of w which satisfy (2.11) and r-Bruhat paths of w which satisfy

$$w^{(1)}(a_1) > \cdots > w^{(p)}(a_p) \quad \text{and} \quad w^{(p)}(a_p) < \cdots < w^{(m)}(a_m). \tag{2.12}$$

(This correspondence can be constructed by starting with a path which satisfies (2.11) and commuting transpositions of this path to make sure (2.12) holds.)

So Theorem 2.4 in case (2.7) can be restated as it originally appeared in [16].

Theorem 2.5. *Assume* $v = ((p, 1^{q-1}), r)$. *Then* $\mathfrak{S}_w\mathfrak{S}_v = \sum \mathfrak{S}_{ww(T)}$, *the sum over all r-Bruhat paths* T *of* w *of shape* $(p, 1^{q-1})$ *which satisfy* (2.12).

3. Insertion algorithm

3.1. Preliminaries. We need some preliminaries before defining the algorithm.

First, let Y be an rc-graph with $w(Y) = v(\lambda, r)$. We think of the Young diagram of λ as the shape of Y, denoted by $sh(Y)$. It is easy to see that any strand s with $s \leqslant r$ intersects exactly λ_{r+1-s} other strands. Let these intersections be in the rows $i_1 \geqslant \cdots \geqslant i_{\lambda_{r+1-s}}$ (one number for each crossing, so that repetitions are allowed), then define $word(Y, s) = i_1 \dots i_{\lambda_{r+1-s}}$. Define $word(Y)$ to be the concatenation $word(Y, 1) \dots word(Y, r)$. Notice that if two strands a, b intersect in Y and $a < b$, then $a \leqslant r < b$. Hence, every crossing of Y corresponds to a single letter in $word(Y)$. Also notice that for any ℓ, the permutation of $Y_{\leqslant \ell}$ is again a shuffle. Moreover, the shape of $Y_{\leqslant \ell}$ is a subdiagram of the shape of Y.

Secondly, we need the following lemma and the construction after the lemma.

Lemma 3.1. (1) *If* R *is an rc-graph and* $l(w(R)t_{cd}) = l(w(R)) - 1$, *then strands* c *and* d *intersect in* R, *and removing their crossing produces another rc-graph.*

(2) *Let* R *be an rc-graph with strands* c *and* d *passing place* (ℓ, j) *but never intersecting in* R. *Then inserting a crossing into place* (ℓ, j) *produces an rc-graph.*

Proof. For $w \in S_\infty$, its length is $l(w) = \#\{(i,j)\colon\ i<j, w(i) > w(j)\}$. Hence,

$$\text{for } c<d, l(wt_{cd}) = l(w)+1 \text{ if and only if } w(c)<w(d) \text{ and}$$
$$\text{there is no } i \text{ with } c<i<d,\ w(c)<w(i)<w(d). \tag{3.1}$$

To prove the first part of the lemma, notice that (3.1) applied to $w = w(R)t_{cd}$ immediately implies that $w(R)(c) > w(R)(d)$. In particular, strands c and d must intersect in R. Remove their crossing to produce the graph R'. Using (2.3), (2.1) and $l(w(R')) = l(w(R)t_{cd}) = l(w(R)) - 1 = |R'|$ we conclude R' is an rc-graph.

To prove the second part, add the crossing of strands c and d in place (ℓ, j) of R to produce a graph R'. By (2.3) it is enough to check $l(w(R')) = l(w(R)) + 1$. Since strands c and d do not intersect in R, $w(R)(c) < w(R)(d)$, hence by (3.1), it is enough to check that there is no i with

$$c<i<d,\ w(R)(c)<w(R)(i)<w(R)(d).$$

It is very easy to see that if such i existed, strand i would have to intersect either strand c or strand d twice in R, which is impossible. $\square$

As a consequence to Lemma 3.1 let us present the following construction. Given an r-Bruhat package $\mathscr{P} = (w, R, T)$, let $m = |T|$. Set $S_m(\mathscr{P}) = R$. Then, by Lemma 3.1, rc-graphs $S_j(\mathscr{P})$ for $0 \leqslant j \leqslant m$ are uniquely defined once we require that $w(S_j(\mathscr{P})) = ww_j(T)$, and $S_j(\mathscr{P})$ is constructed out of $S_{j+1}(\mathscr{P})$ by removing exactly one crossing. An example is provided in Section 6.1.

3.2. Outline. Here is an informal outline of the algorithm. The formal definitions start in Section 3.3.

Assume we are given an rc-graph R with $w = w(R)$ and an rc-graph Y with $v(\lambda, r) = w(Y)$ satisfying (2.6) or (2.7). Our goal is to define a graph $R \leftarrow Y$ and a tableau of transpositions $T(R, Y)$ of $v(\lambda, r)$. By analogy with the Schensted algorithm we would like to insert one by one the letters of $word(Y)$ into R. It does not work, but something similar does.

The algorithm starts with row r and goes up. After it finishes with row ℓ, it produces the result of the insertion of $Y_{\leqslant \ell}$ into $R_{\leqslant \ell}$: $R(\ell) = R_{\leqslant \ell} \leftarrow Y_{\leqslant \ell}$ and $T(\ell) = T(R_{\leqslant \ell}, Y_{\leqslant \ell})$. Think of $T(\ell)$ as "the history" of this insertion. Namely, $T(\ell)$ says how to go from $w(R_{\leqslant \ell})$ to $w(R(\ell))$ along a path in the Bruhat order labelled by the letters of the word of $Y_{\leqslant \ell}$.

Assume the algorithm has been performed up to row $\ell + 1$ so that $R(\ell + 1)$ and $T(\ell + 1)$ are defined. The next row where it operates is row ℓ. To define $R(\ell)$ and $T(\ell)$ the algorithm goes through steps which mimic inserting the letters of $word(Y_{\leqslant \ell})$ into R. But it essentially operates only in row ℓ so that the rc-graphs $R(\ell)$ and $R(\ell + 1)$ are identical outside row ℓ. Moreover, the r-Bruhat path from $w(R_{\leqslant \ell})$ to $w(R(\ell))$ given by $T(\ell)$ is constructed out of the path from $w(R_{\leqslant \ell+1})$ to $w(R(\ell + 1))$ such that properties (2.8) and (2.9) always hold. The shape of $T(\ell)$ contains all the boxes of the shape of $T(\ell + 1)$. In the simplest case, which rarely happens, $T(\ell)$ is constructed out of $T(\ell + 1)$ by filling the boxes of $T(\ell)$ which are not in $T(\ell + 1)$.

3.3. Steps of the algorithm. Throughout the rest of this section the statements which require proofs are underlined and then proved in Section 4. The footnotes supply the reader with the references to the relevant examples in Section 6.

For each (ℓ, i) with $r \geqslant \ell \geqslant 1$ and $0 \leqslant i \leqslant m_\ell$ (where $m_\ell = |Y_{\leqslant \ell}|$) the algorithm performs a step which we call *step* (ℓ, i). The steps go in the following order. Step $(\ell, i + 1)$ goes after step (ℓ, i) if $m_\ell - 1 \geqslant i \geqslant 0$. Step $(\ell, 0)$ follows step $(\ell + 1, m_{\ell+1})$.

Before giving the detailed description of each step let us present the data produced by each step and the conditions this data satisfies. Step (ℓ, i) constructs

(3.2) rc-graphs $R(\ell, i)$, with no crossings above row ℓ,
(3.3) tableau of transpositions $T(\ell, i)$ for the shuffle $w(Y_{\leqslant \ell})$ filled up to i.

Here, $R(\ell, i)$ and $T(\ell, i)$ play the role of the intermediate result of the algorithm. After each step, $R(\ell, i)$ and $T(\ell, i)$ must satisfy the following conditions

(3.4) $E(w(R_{\leqslant \ell}), T(\ell, i))$ is a row and column strict tableau,
(3.5) $\mathscr{P}(\ell, i) = (w(R_{\leqslant \ell}),\ R(\ell, i), T(\ell, i))$ is an r-Bruhat package.

Remark 3.2. Conditions (3.4) and (3.5) are analogues of (2.8) and (2.9) from Theorem 2.2. That is why they have to be satisfied by the intermediate results of the algorithm. Condition (3.4) is the condition, which we do not know how to generalize to cases other than (2.6) and (2.7).

For each row ℓ we start with *a row-to-row* step $(\ell, 0)$ which sets up the data needed for performing the algorithm in this row. Then we perform a step for each letter of $word(Y_{\leqslant \ell})$. If this letter is equal to ℓ it is *an insertion* step and we will insert a crossing in row ℓ to the current rc-graph. If the letter is not ℓ, then we perform *a rectification* and rectify, if necessary, both the rc-graph $R(\ell, i)$ and the path given by $T(\ell, i)$ to guarantee both (3.4) and (3.5) are satisfied.

The rest of Section 3.3 introduces the additional notations and states the two additional conditions which clarify certain parts of the algorithm and simplify certain proofs.

After we are finished with all the steps for row ℓ, we are given $R(\ell, m_\ell)$ and $T(\ell, m_\ell)$, which, to shorten the notations, we denote by $R(\ell)$ and $T(\ell)$. Denote by $\mathscr{P}(\ell)$ the r-Bruhat package $(w(R_{\leqslant \ell}), R(\ell), T(\ell))$.

Fix ℓ, let $word(Y_{\leqslant \ell}) = k_1 \dots k_{m_\ell}$. Each letter k_i of $word(Y_{\leqslant \ell})$ corresponds to a crossing of $Y_{\leqslant \ell}$ in row k_i. If $k_i > \ell$, then the letter k_i is also a part of $word(Y_{\leqslant \ell+1}) = k'_1 \dots k'_{m_{\ell+1}}$, let the index of k_i inside $word(Y_{\leqslant \ell+1})$ be i_+. Set $i_+ = 0$, if $i = 0$. (So, if we think of $sh(Y_{\leqslant \ell+1})$ as a subdiagram of $sh(Y_{\leqslant \ell})$, then box i of $Y_{\leqslant \ell}$ coincides with box i_+ of $sh(Y_{\leqslant \ell+1})$.) The two additional conditions are

(3.6) $x^{R(\ell)} = x^{R_{\leqslant \ell}} x^{Y_{\leqslant \ell}}$,
(3.7) If $k_i < \ell$, then $R(\ell, i)_{\leqslant \ell+1} = S_{i_+}(\mathscr{P}(\ell + 1))$.

It will be obvious from the description of the algorithm that these conditions are always satisfied. Condition (3.6) implies that (2.10) holds for the final result, while (3.7) indicates that step (ℓ, i) only operates in row ℓ, as the part of $R(\ell, i)$ which lies below row ℓ is uniquely determined by $\mathscr{P}(\ell + 1)$.

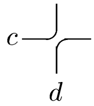
3.4. Start of the algorithm. [2] Set $R(r, 0) = R_{\leqslant r}$ and let $T(r, 0)$ be the empty tableau of shape $sh(Y_{\leqslant r})$, then $\underline{R(r, 0) \text{ and } T(r, 0) \text{ satisfy (3.4) and (3.5)}}$.

3.5. Row-to-row steps. [3]Each step $(\ell, 0)$ is called *a row-to-row step*. This step sets $T(\ell, 0)$ to be the empty tableau of shape $sh(Y_{\leqslant \ell})$ and

$$R(\ell, 0) = S_0(\mathscr{P}(\ell + 1)) \cup R_\ell. \tag{3.8}$$

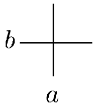
Then $\underline{R(\ell, 0) \text{ is an rc} - \text{graphs}}$ and $\underline{R(\ell, 0) \text{ and } T(\ell, 0)\text{satisfy (3.4) and (3.5)}}$.

[2] See steps (3,0) from Example 6.3 and (2,0) from Example 6.4.
[3] See steps (1,0) from Example 6.2 and (1,0) from Example 6.4.

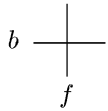


with $c \leq r < d$,

Fig. 4. A place where an insertion is allowed.



where (ab) is the entry of box i of T'.



where (ab) and (af) are the entries of boxes i and $i-1$ of T'.

Fig. 5. One of these crossing in row ℓ needs to be removed.

As mentioned before, this step sets up the data for performing the algorithm in row ℓ. $T(\ell+1)$ defines a path from $w(R_{\leqslant \ell+1})$ to $w(R(\ell+1))$. On the level of rc-graph this path is given by the rc-graphs $S_0(\mathscr{P}(\ell+1)), \dots, S_{m_{\ell+1}}(\mathscr{P}(\ell+1))$. So, we can think of the construction of $R(\ell,0)$ as of the backtracking the algorithm from $R(\ell+1)$ to $S_0(\mathscr{P}(\ell+1))$ and then adding the crossings of R that lie in row ℓ.

3.6. Insertions. [4]Assume $word(Y_{\leqslant \ell}) = k_1 \dots k_{m_\ell}$. If k_i is the letter ℓ, then step (ℓ,i) is called *an insertion step*.

During an insertion step, we say that an insertion into a place (ℓ,j) is allowed, if strands c,d pass this place in $R(\ell,i-1)$ as shown in Fig. 4 and $c \leqslant r < d$.

There exists a place in row ℓ of $R(\ell,i-1)$ where an insertion is allowed. Find the rightmost such place (ℓ,j_0) and let strands c and d pass through it. Define $R(\ell,i)$ by adding a crossing to $R(\ell,i-1)$ into place (ℓ,j_0). Define $T(\ell,i)$ by adding (cd) to box i of $T(\ell,i-1)$. Then $R(\ell,i)$ is an rc-graph and (3.4) and (3.5) are satisfied.

3.7. Rectifications. If $k_i > \ell$, then step (ℓ,i) is called *a rectification*. The first part of rectification is to define a graph R' and a tableau of transpositions T'. The rc-graph $S_{i_+}(\mathscr{P}(\ell+1))$ has one more crossing than $S_{i_+-1}(\mathscr{P}(\ell+1))$, add this crossing to $R(\ell,i-1)$ to produce R'. (Then, since (3.7) holds for $R(\ell,i-1)$, R' coincides with $S_{i_+}(\mathscr{P}(\ell+1))$ below row ℓ and row ℓ of R' is the same as row ℓ of $R(\ell,i-1)$.) To produce T', add to box i of $T(\ell,i-1)$ the entry (ab) of box i_+ of $T(\ell+1)$.

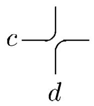
If $E(w(R_{\leqslant \ell}),T')$ is row and column strict and $(w(R_{\leqslant \ell}),R',T')$ is an r-Bruhat package, set $R(\ell,i) = R'$ and $T(\ell,i) = T'$ and move on to the next step.[5] Otherwise, there is a crossing in R' in row ℓ which fits the description in Fig. 5.

If R' has a crossing which looks like the first crossing of Fig. 5, remove this crossing to produce R'' and remove (ab) from box i of T' to produce T''. Otherwise remove the crossing of strands b and f shown in Fig. 5 to produce R'', remove (ab) from box i in T' and replace the entry of box $i-1$ of T' by (ab) to produce T''.

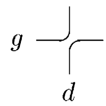
We say that insertions into places in R'' shown in Fig. 6 are allowed.

[4] See steps (2,3) from Example 6.2, (3,1) from Example 6.3 and (2,2) from Example 6.4.

[5] See steps (1,1) from Example 6.2 and (2,1) from Example 6.3.



with $c \le r < d$,



where (eg) is the entry of box $i-1$ of T'', and $r < d$

Fig. 6. Places, where insertions are allowed during rectification.

Remember that R'' is constructed out of R' by removing a crossing from some place (ℓ, j_0). There exists a place in row ℓ of $R(\ell, i\text{-}1)$ to the left of (ℓ, j_0) where an insertion is allowed. Find the rightmost such place to the left of (ℓ, j_0). Insert a crossing there to produce $R(\ell, i)$. If this is the place of the first type from Fig. 6, then insert (cd) into box i of T'' to define $T(\ell, i)$. In the second case, replace the entry of box $i-1$ of T'' by (ed) and place (eg) into box i of T'' to define $T(\ell, i-1)$. Then $R(\ell, i)$ is an rc-graph and (3.4) and (3.5) are satisfied.[6]

3.8. End of the algorithm. Set $R \leftarrow Y = R(1, m_1)$ and $T(R, Y) = T(1, m_1)$.

3.9. Concluding remarks. As mentioned before, the algorithm can be simplified in all the cases except for the case when the shape of v is a hook. For case (2.6) (see [8]) our algorithm produces the same result as inserting letters of $word(Y)$ one by one into R using the algorithm of Bergeron and Billey [1] which, in the case when w is an r-shuffle, is the Schensted algorithm.

For the case when the shape of v is a row a simplified algorithm is given in [9]. In the case when the shape of v is a column the only simplification of our insertion algorithm, which we know of, is omitting the second picture from Fig. 6.

4. Proof of Theorem 2.2

To prove Theorem 2.2, it is enough to prove all the statements underlined in Section 3. Let us repeat these statements.

(1) $R(r, 0)$ and $T(r, 0)$ satisfy conditions (3.4) and (3.5).
(2) For $r > \ell \geqslant 1$, $R(\ell, 0)$ is an rc-graph and $R(\ell, 0)$, $T(\ell, 0)$ satisfy (3.4) and (3.5).
(3) During every insertion step, there exists a place in row ℓ of $R(\ell, i-1)$ where an insertion is allowed. After every insertion step, $R(\ell, i)$ is an rc-graph and (3.4), (3.5) are satisfied.
(4) During every rectification, if R' is not an rc-graph or $E(w(R_{\leqslant \ell}), T')$ is not row and column strict, there is a crossing in R' shown in Fig. 5.
(5) During every rectification, there exists a place in row ℓ of $R(\ell, i-1)$ to the left of place (ℓ, j_0) where an insertion is allowed. After every rectification, $R(\ell, i)$ is an rc-graph and (3.4), (3.5) are satisfied.

[6] In steps (1,3) and (1,4) from Example 6.2 first cases of Figs. 5 and 6 occur. In step (1,2) from Example 6.3 the second case of Fig. 5 and the first case of Fig. 6 occur. In step (1,2) from Example 6.4 the first case of Fig. 5 and the second case of Fig. 6 occur.

Remark 4.1. Before going into the proofs, let us make a remark about the entries of the tableaux $E(w(R_{\leqslant\ell}), T(\ell))$ and $E(w(R_{\leqslant\ell}), T(\ell, i))$. The ith entry e_i of $E(w(R_{\leqslant\ell}), T(\ell))$ and $E(w(R_{\leqslant\ell}), T(\ell, i))$ is equal to $w(R(\ell, i))(b_i)$. Graphically, it is just the column of $R(\ell, i)$ where strand b_i ends. This will be helpful in understanding a lot of statements about $E(w(R_{\leqslant\ell}), T(\ell))$.

4.1. Proof of (1). Since $T(r, 0)$ is an empty tableau, condition (3.4) is vacuous, while (3.5) follows directly from $R(r, 0) = R_{\leqslant r}$.

4.2. Proof of (2). Since $S_0(\mathscr{P}(\ell+1))$ has no crossings above row $\ell+1$ and R_ℓ has only crossings in row ℓ, we know from (3.8)

$$\begin{aligned} w(R(\ell, 0)) &= w(S_0(\mathscr{P}(\ell+1)))w(R_\ell) = w(R_{\leqslant\ell+1})w(R_\ell) \\ &= w(R_{\leqslant\ell+1} \cup R_\ell) = w(R_{\leqslant\ell}). \end{aligned}$$

On the other hand,

$$|R(\ell, 0)| = |S_0(\mathscr{P}(\ell+1))| + |R_\ell| = |R_{\leqslant\ell+1}| + |R_\ell| = |R_{\leqslant\ell}|.$$

Hence $l(w(R(\ell, 0))) = |R(\ell, 0)|$ and using (2.3) we conclude $R(\ell, i)$ is an rc-graph.

Since $T(r, 0)$ is an empty tableau, condition (3.4) is vacuous, while (3.5) follows immediately from $w(R(\ell, 0)) = w(R_{\leqslant\ell})$.

4.3. Proof of (3). For an rc-graph R define the sequence of strands c_k by $c_0 = \ell$ and

$$a_k = w(R_{\leqslant\ell})(c_k), \quad c_{k+1} = w(R_{\leqslant\ell+1})^{-1}(a_k + 1). \tag{4.1}$$

Another way of defining this sequence is to look at all strands of R which have horizontal parts in row ℓ, in other words, which do not cross row ℓ vertically. These strands do not cross each other in row ℓ and, hence, can be ordered from left to right with the first strand being ℓ. This is the sequence of strands c_k. For example, if R is the third graph of Fig. 2 and $\ell = 1$, then $c_0 = 1$, $c_1 = 4$, $c_2 = 2$ and so on.

Throughout the proofs we will use different parts of this sequence for different rc-graphs. Whenever we refer to a part of this sequence, we specify a strand $c_{\bar{k}}$ where this part starts and a strand $c_{\underline{k}}$ where it ends (if the later is omitted, it means the part we consider runs to infinity).

We start the proof of (3) with showing that there are places in $R(\ell, i-1)$, where insertions are allowed. For $R(\ell, i-1)$, consider the whole sequence of strands c_k (namely $\bar{k} = 0$).

It is clear that for large k, $c_k > r$. Since $c_0 = \ell \leqslant r$, there exists k' with $c_{k'} \leqslant r < c_{k'+1}$. By construction, we know that strands $c_{k'}$ and $c_{k'+1}$ pass next to each other in row ℓ in some place (ℓ, j). Then an insertion into place (ℓ, j) is allowed, as $c = c_{k'} \leqslant r < c_{k'+1} = d$ as required in Fig. 4.

4.3.1. *$R(\ell, i)$ is an rc-graph and (3.5) holds.* $R(\ell, i)$ is an rc-graph by the second part of Lemma 3.1. Moreover, $l(w(R(\ell, i)) = l(w(R(\ell, i-1))) + 1$ and

$$w(R(\ell, i)) = w(R(\ell, i-1))t_{cd}, \quad w(T(\ell, i)) = w(T(\ell, i-1))t_{cd}, \tag{4.2}$$

which immediately implies that (3.5) holds.

4.3.2. *After every insertion step, (3.4) holds.* If $i = 1$, so that the insertion step corresponds to the first letter $k_1 = \ell$, condition (3.4) is vacuous. Otherwise, we will show that there exists j such that an insertion into (ℓ, j) is allowed and

$$\ell + j - 1 = w(R(\ell, i-1))t_{c'd'}(d') > w(R(\ell, i-1))(f), \tag{4.3}$$

where c', d' are the strands passing place (ℓ, j) and (ef) is the entry of box $i-1$ of $T(\ell, i-1)$. This will be enough to prove (3.4). Indeed if $R(\ell, i)$ is defined by adding a crossing of strand c, d in place (ℓ, j_0), then $j_0 \geqslant j$. Therefore, by Remark 4.1, (4.3) holds for c', d' substituted by c, d. Hence $E(w(R_{\leqslant \ell}), T(\ell, i))$ is row and column strict.

To show that j satisfying (4.3) exists, consider the part of the sequence c_k for $R(\ell, i-1)$ which starts at $c_{\bar{k}}$ defined as follows. If strand e in $R(\ell, i-1)$ intersects vertically another strand e' in row ℓ then set $c_{\bar{k}} = e'$, otherwise set $c_{\bar{k}} = e$ (notice that $e' < e \leqslant r$). Since $c_{\bar{k}} \leqslant r$ and $c_k > r$ for large k, there exists $k' \geqslant \bar{k}$ with $c_{k'} \leqslant r < c_{k'+1}$. Let $c' = c_{k'}$ and $d' = c_{k'+1}$. Then strands c' and d' pass next to each other in row ℓ at a place (ℓ, j) and an insertion into (ℓ, j) is allowed. Moreover, since strand $c_{\bar{k}}$ is either strand e or it intersects strand e horizontally in row ℓ, the following calculation proves (4.3)

$$\begin{aligned} w(R(\ell-1, i)t_{c'd'})(d') &= w(R(\ell, i-1))(c') \geqslant w(R(\ell, i-1))(e) \\ &> w(R(\ell, i-1))(f). \end{aligned}$$

4.4. Proof of (4). Recall that R' is constructed out of $R(\ell, i-1)$ by adding a crossing of strands a, b to guarantee R' coincides with $S_{i_+}(\mathscr{P}(\ell+1))$ below row ℓ.

Let us show that R' is not an rc-graph if and only if strands a and b intersect in row ℓ as shown in Fig. 5. Indeed, if a, b intersect in row ℓ, they intersect twice in R', so R' is not an rc-graph. Conversely, if they do not intersect in row ℓ of R', then they do not intersect in $R(\ell, i-1)$. (If they intersect below row ℓ in $R(\ell, i-1)$ then $S_{i_+}(\mathscr{P}(\ell+1))$ is not an rc-graph.) Hence by the second part of Lemma 3.1, R' is an rc-graph.

Since $w(R') = w(R(\ell, i-1))t_{ab}$, we can immediately conclude that if R' is an rc-graph, then $(w(R_{\leqslant \ell}), R', T')$ is an r-Bruhat package.

To finish the proof of (4), it remains to prove that if $E(w(R_{\leqslant \ell}), T')$ is not column or row strict then the second crossing from Fig. 5 must occur. We will do it separately for cases (2.6) and (2.7).

4.4.1. *Case* (2.6). We start with some preliminary lemmas, which we also use in the proof of (5). These lemmas (especially Lemmas 4.4) also explains why the algorithm can be simplified in this case.

Lemma 4.2. *If $u \in S_\infty$ is an r-semi-shuffle, $u' = ut_{ab}$ with $a \leqslant r < b$ and $l(u') = l(u) + 1$, then u' is an r-semi-shuffle.*

Proof. We must show that if $r < b' < b''$ then $ut_{ab}(b') < ut_{ab}(b'')$. If $b' \neq b$ and $b'' \neq b$, then $ut_{ab}(b') = u(b') < u(b'') = ut_{ab}(b'')$, since u is an r-semi-shuffle.

If $b' = b$, then $ut_{ab}(b') = u(a) < u(b) < u(b'') = ut_{ab}(b'')$, since $l(ut_{ab}) = l(u) + 1$ and u is an r-semi-shuffle.

If $b'' = b$, then $ut_{ab}(b') = u(b') < u(a) = ut_{ab}(b'')$, where $u(b') < u(a)$, since otherwise $a < b' < b$ and $u(a) < u(b') < u(b)$ which contradicts (3.1). $\square$

Assume T is a tableau of transpositions (possibly partially filled). If $(a_k b_k)$ are the entries of T, let $B(T)$ be the tableau of the same shape with the entries b_k.

Lemma 4.3. *Let w be an r-semi-shuffle and T be an r-Bruhat path of w. Then $E(w, T)$ is row and column strict if and only if $B(T)$ is row strict.*

Proof. Assume $u \in S_\infty$ is an r-semi-shuffle and $l(ut_{ab}) = l(u) + 1$ for $a \leqslant r < b$. Let $b' > r$ then it is easy to see by Lemma 4.2 that

$$u(b') < ut_{ab}(b) \text{ if and only if } b' < b. \tag{4.4}$$

Clearly, (4.4) implies that rows of $E(w, T)$ strictly increase from left to right if and only if the same holds for $B(T)$.

Let us show that if $B(T)$ is row strict, then $E(w, T)$ is column strict. Denote by e_k the entry of box k of $E(w, T)$. Let box i' be directly above box i in $sh(T)$. Consider boxes i through i' in the diagram $sh(T)$ as shown in Fig. 7.

To show $E(w, T)$ is column strict it is enough to show $e_{i'} < e_i$ for any i not in the top row. If $B(T)$ is row-strict, then $b_i \neq b_{\tilde{i}}$ for $i < \tilde{i} < i'$. If $b_{i'} < b_i$, then

$$e_i = ww_i(T)(b_i) = ww_{i'}(T)(b_i) > ww_{i'}(T)(b_{i'}) = e_{i'},$$

since by Lemma 4.2 $ww_{i'}(T)$ is an r-semi-shuffle. If $b_{i'} = b_i$, then

$$e_i = ww_i(T)(b_i) = ww_{i'-1}(T)(b_i) = ww_{i'}(T)(a_{i'}) > ww_{i'}(T)(b_{i'}) = e_{i'},$$

since $l(ww_{i'-1}(T)) + 1 = l(ww_{i'}(T))$.

Conversely, assume $E(w, T)$ is row and column strict. To show $B(T)$ is row strict, it is enough to show $b_{i'} \leqslant b_i$ for any i such that box i is not in the top row. Assume for

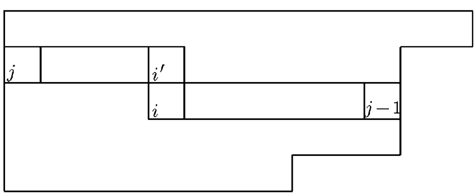


Fig. 7. Boxes i through i' of T.

a moment $b_i \neq b_{\tilde{i}}$ for any $j \leqslant \tilde{i} \leqslant i'$. Then

$$ww_{i'}(T)(b_i) = ww_i(T)(b_i) = e_i > e_{i'} = ww_{i'}(T)(b_{i'}),$$

since $E(w, T)$ is row and column strict. Thus, since $ww_{i'}(T)$ is an r-semi-shuffle, we conclude $b_{i'} < b_i$.

Otherwise, if $b_i = b_{\tilde{i}}$ for some $j \leqslant \tilde{i} \leqslant i'$, then we use induction on i to show that $\tilde{i} = i'$. If i is the first box in its row, then $b_{i'} = b_i$, as $i' = j$. Otherwise, assume the box underneath box $\tilde{i}$ contain $\bar{b}$. By induction $\bar{b} \geqslant b_{\tilde{i}}$. On the other hand, we know that $\bar{b} < b_i$ if $\tilde{i}_- \neq i$. Hence, if $b_i = b_{\tilde{i}}$, then box i must be underneath box $\tilde{i}$. □

Lemma 4.4. *In case* (2.6) *the second pictures of Figs.* 5 *and* 6 *never happen. Moreover, if p is the entry of box i of $B(T(\ell))$ and q is the entry of box i_+ of $B(T(\ell))$ then $p \leqslant q$.*

Proof. Adopt the notations of the rectification step, namely, let R' be the graph that is produced out of $R(\ell, i-1)$ by adding the crossing of strands a and b. If strands a and b do not intersect in $R(\ell, i-1)$, then it was shown already that R' is an rc-graph. By Lemma 4.2, $w(R(\ell, i-1))$ as well as $w(R')$ are r-semi-shuffles. Hence, strand $b > r$ cannot intersect another strand $f > r$ in R'. Therefore, the second picture of Fig. 5 is impossible.

If the rectification step does not stop at R', the crossing of strands a and b is removed from place (ℓ, j_0) of R' to produce R''. If strands g and d pass a place in row ℓ of R'' to the left of (ℓ, j_0) which looks like the second picture of Fig. 6, then $r < g < d$, since $w(R'')$ is an r-semi-shuffle, and $g, d \neq b$. Moreover,

$$w(R(\ell, i-1))(e) > w(R(\ell, i-1))(d), \tag{4.5}$$

since $w(R(\ell, i-1))(e) > w(R(\ell, i-1))(g)$ and strands d and e cannot intersect twice in $R(\ell, i-1)$.

Then by $w(R(\ell, i-2))t_{eg} = w(R(\ell, i-1))$ and (4.5)

$$w(R(\ell, i-2))(g) = w(R(\ell, i-1))(e) > w(R(\ell, i-1))(d) = w(R(\ell, i-2))(d)$$

which is impossible, since $w(R(\ell, i-2))$ is an r-semi-shuffle and $r < g < d$.

Hence, in case (2.6) during rectification only the first picture of Fig. 6 happens. If the algorithm inserts a crossing into a place where strands c and d pass to the left of the place (ℓ, j_0) then $d < b$ (otherwise strands d and b intersect in R'', which is impossible, since $w(R'')$ is an r-semi-shuffle). This proves the last part of the lemma, as $p = d < b = q$. (Notice that q is equal to p if R' is an rc-graph.) □

In light of Lemma 4.4, to give the proof of (4) in case (2.6) we will prove that if R' is an rc-graph, then $E(w(R_{\leqslant \ell}), T')$ is row and column strict, or, by Lemma 4.3, it is enough to show $B(T')$ is row strict.

Let b_k be the entries of $B(T')$, so that $b_i = b$. Since boxes 1 through $i-1$ of $B(T')$ and $B(T(\ell, i-1))$ coincide, $B(T')$ can fail to be row strict if box $i-1$ is in the same row as box i and $b_{i-1} \geqslant b_i = b$, or, if there is box i_* underneath box i such that $b = b_i > b_{i*}$. Let us show both cases are impossible. This will finish the proof of (4) in case (2.6).

If $(\bar{e}\bar{f})$ is the entry of box $i_+ - 1$ of $T(\ell+1)$, then by Lemma 4.4 $b_{i-1} \leqslant \bar{f}$. At the same time if box $i-1$ is in the same row as box i, then $\bar{f} < b_i = b$, since $B(T(\ell+1))$ is row strict. So, $b_{i-1} < b$ whenever box $i-1$ is in the same row as box i.

It remains to show that if box i_* is the box underneath box i in $B(T')$, then $b \leqslant b_{i*}$. We will prove it by induction on i. We first prove the induction step when step $(\ell+1, i_*)$ is an insertion.

Denote temporarily $\tilde{R} = R(\ell, i_* - 1)$. We will prove there exists a place in $\tilde{R}$ in row $\ell + 1$ shown in Fig. 4 with $d \geqslant b$. Then we will be guaranteed $b_{i*} \geqslant b$.

Look at how strand b passes row $\ell+1$ in $\tilde{R}$. If it passes it vertically, then it intersects certain strand a' with $a' \leqslant r$ (since $w(\tilde{R})$ is an r-semi-shuffle). Consider the sequence c_k for $\tilde{R}$, set $c_{\tilde{k}} = a'$. By the same argument as in the proof of (3) we can find a place (ℓ, j), shown in Fig. 4, to the right of the place where strand b passes row $\ell+1$. Hence, for such a place $d > b$.

If strand b does not pass row ℓ vertically, look at the whole sequence c_k for $\tilde{R}$. Strand b is an element of this sequence. Let $b = c_{\tilde{k}}$. Let us show that

$$c_{\tilde{k}-1} \leqslant r \quad \text{or} \quad c_{\tilde{k}-1} = b - 1. \tag{4.6}$$

Indeed, if $c_{\tilde{k}-1} > r$, then if there exist b' with $c_{\tilde{k}-1} < b' < b$, then strand b' must intersect either strand $c_{\tilde{k}}$ or strand b, which is impossible, since $w(\tilde{R})$ is an r-semi-shuffle. Hence (4.6) holds.

If $c_{\tilde{k}-1} \leqslant r$, then strands $c = c_{\tilde{k}}$ and $d = b$ pass next to each other in row $\ell+1$ at a place (ℓ, j), so an insertion into (ℓ, j) is allowed. It implies $b_{i*} \geqslant b$.

It remains to consider the case when $c_{\tilde{k}-1} = b - 1$. Let $\bar{b}_k$ denote the entries of $B(T(\ell+1))$. We will prove that

$$i_* \neq 1 \quad \text{and} \quad c_{\tilde{k}-1} = \bar{b}_{i-1} = b - 1. \tag{4.7}$$

If (4.7) holds, then, since step (ℓ, i_*) is an insertion step, i_* is not the first box in row ℓ and box $i_* - 1$ is in the same row as box i_*. Hence by the induction assumption

$$b - 1 = \bar{b}_{i-1} \leqslant b_{i*-1} < b_{i*}.$$

Therefore, since $b > \bar{b}_{i-1} = b - 1$, we conclude $b = b_i \leqslant b_{i*}$.

It remains to show that if $c_{\tilde{k}-1} = b - 1$ then (4.7) holds. Since $w(\tilde{R})$ is an r-semi-shuffle and strands b and $b-1$ pass next to each other in row ℓ of $\tilde{R}$,

$$w(\tilde{R}_{\leqslant \ell+1})(b) - 1 = w(\tilde{R}_{\leqslant \ell+1})(b-1).$$

$R(\ell, i-1)_{\leqslant \ell+1}$ is constructed out of $\tilde{R}_{\leqslant \ell+1}$ by adding some crossings. It is not difficult to see that if (4.7) fails, none of these crossings involve strands b or $b-1$. So

$$w(R(\ell, i-1)_{\leqslant \ell+1})(b) - 1 = w(R(\ell, i-1)_{\leqslant \ell+1})(b-1).$$

But it is impossible by (3.1), since $a < b - 1 < b$ and

$$w(R(\ell, i-1)_{\leqslant \ell+1})(a) < w(R(\ell, i-1)_{\leqslant \ell+1})(b-1) < w(R(\ell, i-1)_{\leqslant \ell+1})(b)$$

while $l(w(R(\ell, i-1)_{\leqslant \ell+1})t_{ab}) = l(w(R(\ell, i-1)_{\leqslant \ell+1})) + 1$.

This finishes the proof in the case step $(\ell+1, i_*)$ is an insertion. If this step is a rectification, let $(\bar{c}, \bar{b})$ be the entry of box $(i_*)_+$ of $T(\ell+1)$. Then $b \leqslant \bar{b}$, since $B(T(\ell+1))$ is row strict. If during this step no crossing is removed from R', then $b_{i_*} = \bar{b}$ and there is nothing to prove. Otherwise, consider the intermediate rc-graph R' of the rectification step (ℓ, i_*). We have to show that there exists a place in R' in row ℓ of the type shown in the first picture of Fig. 6 with $b \leqslant d \leqslant \bar{b}$. This can be done by an argument almost identical to the case when step (ℓ, i_*) is an insertion. The only difference is that all the considered parts of the sequence c_k must be finite, ending at $c_{\underline{k}} = \bar{c}$.

4.4.2. *Case* (2.7). As before, let (ab) and (ef) be the entries of boxes i and $i-1$ of T'. Assume $(\ell, i-1)$ is a rectification (the argument below can be easily modified to provide a proof in the case when step $(\ell, i-1)$ is an insertion step). Assume $(\bar{e}, \bar{f})$ is the entry of box $i_+ - 1$ of $T(\ell+1)$.

Start with the case when box i is not in the first column of $sh(T')$. Let us show that if R' is an rc-graph, then $E(w(R_{\leqslant \ell}), T')$ is row and column strict. It is obvious if $(ef) \neq (\bar{e}\bar{f})$, since in this case the entry of box $i-1$ of $E(w(R_{\leqslant \ell}), T')$ is smaller then the value of box $i_+ - 1$ of $E(w(R_{\leqslant \ell}), T(\ell+1))$.

If $(ef) = (\bar{e}\bar{f})$ and $E(w(R_{\leqslant \ell}), T')$ is not row and column strict, then strands b and f intersect in row ℓ in R' such that $\ell \in f \boxplus b$. But then f intersects a horizontally in $R(\ell, i-1)$, which is impossible since $a < f$.

In the case when box i is in the first column and R' is an rc-graph we will show that one of the following holds:

$$w(R(\ell, i-1))(f) > w(R(\ell, i-1))(a), \tag{4.8}$$

$$a = e \quad \text{and} \quad \ell \in a \boxplus f \ \text{ in } R(\ell, i-1). \tag{4.9}$$

If (4.8) holds, then $E(w(R_{\leqslant \ell}), T')$ is row and column strict. If (4.9) holds, then $\ell \in b \boxplus f$ in R' as shown in the second picture of Fig. 5. So, it remain to prove (4.8) or (4.9) hold in the case when box i is in the first column and R' is an rc-graph.

Since $E(w(R_{\leqslant \ell+1}), T(\ell+1))$ is a row and column strict tableau, we know

$$w(S_{i_+-1}(\mathscr{P}(\ell+1)))(\bar{f}) > w(S_{i_+}(\mathscr{P}(\ell+1)))(b) = w(S_{i_+-1}(\mathscr{P}(\ell+1)))(a). \tag{4.10}$$

Moreover, removing crossings from row ℓ of $R(\ell, i-1)$ produces $S_{i_+-1}(\mathscr{P}(\ell+1))$.

In the case $(ef) = (\bar{e}\bar{f})$ inequality (4.10) implies (4.8) unless $\ell \in a \boxplus f$ in $R(\ell, i-1)$. But it is not difficult to see that if $(ef) = (\bar{e}\bar{f})$, strands a and f cannot intersect.

Otherwise, if $(ef) \neq (\bar{e}\bar{f})$, we will show during the proof of (5) that a crossing of strands $\bar{e}, \bar{f}$ has been removed during step $(\ell, i-1)$ from place $(\ell, \tilde{j})$ and then another crossing has been inserted to the left of $(\ell, \tilde{j})$. Assume $\bar{R}'$ is the intermediate rc-graph in step $(\ell, i-1)$ constructed by removing a crossing from $R(\ell, i-2)$. Consider the sequence c_k for $\bar{R}'$. Let $c_{\bar{k}} = a$ if a does not intersect row ℓ vertically, otherwise there exists a unique a' with $\ell \in a' \boxplus a$, set $c_{\bar{k}} = a'$. Then set $\bar{f} = c_{\underline{k}}$ ($\bar{f}$ passes row ℓ horizontally). There exists a strand $c_{k'}$ in the considered part of the sequence (that is $\bar{k} \leqslant k' < \underline{k}$) and a place (ℓ, j) where an insertion is allowed with strands $c_{k'}$ and

$c_{k'+1}$ passing through (ℓ, j). Choose such a place with the largest possible j, let it be (ℓ, j_1). Then $R(\ell, i-1)$ and $T(\ell, i-1)$ are defined in such a way that $e = c_{k'}$ and $f = c_{k'+1}$. If $\bar{k} = k'$ and $c_{\bar{k}} = a$, (4.9) holds, otherwise (4.8) must be satisfied.

4.5. Proof of (5). Assume that a crossing at place (ℓ, j_0) in R' has been removed to produce R''. It is not difficult to see that R'' is an rc-graph, $(w(R_{\leqslant \ell}), R'', T'')$ is an r-Bruhat package and $E(w(R_{\leqslant \ell}), T')$ is row and column strict. We need to show that there exists a place where an insertion is allowed to the left of (ℓ, j_0) and, after $R(\ell, i)$ and $T(\ell, i)$ are defined, (3.4) and (3.5) are satisfied.

4.5.1. *Case* (2.6). As in the proof of (3), we can use sequence c_k for R'' to show that a place where an insertion is allowed to the left of the place (ℓ, j_0) exists. Moreover, by Lemma 4.4, the rightmost place (ℓ, j_1) where an insertion is allowed looks like the first picture in Fig. 6.

After we insert a crossing into place (ℓ, j_1) of R'' to define $R(\ell, i)$ and (cd) into box i of T'' to define $T(\ell, i-1)$, it is easy to see $R(\ell, i)$ is an rc-graph and (3.5) is satisfied. By Lemma 4.3, to show that (3.4) holds it is enough to show $B(T(\ell, i))$ is row strict. Notice that $B(T(\ell, i))$ differs from $B(T(\ell, i-1))$ only in box i. So we just have to check that the entry of box i is still greater than the entry of the box to the left of box i and not greater than the entry of the box below box i. This can be done by an argument which is almost identical to the argument used in Section 4.4.1.

4.5.2. *Case* (2.7). Recall that (ab) is the entry of box i of T', (eg) is the entry of box $i-1$ of T''. Consider the part of the sequence c_k for R'' which starts at $c_{\bar{k}} = \ell$ and ends at $c_{\underline{k}} = b$. Then there exists k' between $\bar{k}$ and $\underline{k} - 1$ such that strands $c_{k'}$ and $c_{k'+1}$ pass next to each other in a place where an insertion is allowed. Let the rightmost place to the left of (ℓ, j_0) where an insertion is allowed be (ℓ, j_1).

Consider the case when box i is in the first column of T''. Then, using the sequence c_k, it is easy to see that place (ℓ, j_1) looks like the first pictures from Fig. 6. (We used this in Section 4.4.2.) Therefore, as in the insertion step, $R(\ell, i)$ is an rc-graph and (3.5) holds. Moreover (3.4) holds, since strand g passes row ℓ to the right of place (ℓ, j_1).

Otherwise, if box i is in the first row, but not the first element of this row, then strand g is either an element of the sequence c_k or intersects one of the strands c_k in row ℓ. So place (ℓ, j_1) could look like the first picture of Fig. 6 and strand g passes to the left of this place. Or, it could look like the second picture of Fig. 6.

If it is the first picture, then, as before, R'' is an rc-graph, (3.5) holds, while (3.4) holds, since strand g passes row ℓ to the left of place (ℓ, j_1).

If it is the second picture, it is easy to see that R'' is an rc-graph and that (3.4) holds while to prove (3.5), we must show

$$l(w(R(\ell, i))) = l(w(R(\ell, i))t_{eg}t_{ed}) + 2 = l(w(R(\ell, i))t_{eg}) + 1. \tag{4.11}$$

To prove the first equality of (4.11), notice $t_{eg}t_{ed} = t_{gd}t_{eg}$, hence

$$\begin{aligned} l(w(R(\ell,i))t_{eg}t_{ed}) &= l(w(R(\ell,i))t_{gd}t_{eg}) = l(w(R'')t_{eg}) \\ &= l(w(R'')) - 1 = l(w(R(\ell,i))) - 2. \end{aligned}$$

For the second equality, notice that $e<g$ and $w(R(\ell,i))(e)>w(R(\ell,i))(g)$, thus

$$l(w(R(\ell,i))t_{eg})<l(w(R(\ell,i))).$$

At the same time, $e<d$ and $w(R(\ell,i))t_{eg}(e)>w(R(\ell,i))t_{eg}(d)$, hence

$$l(w(R(\ell,i)t_{ef}t_{ed})<l(w(R(\ell,i))t_{eg}).$$

This proves the second part of (4.11).

5. Inverse insertion algorithm

Let U be an rc-graph and T be a tableau of transposition of $v(\lambda,r)$ such that T is an r-Bruhat path of $w = w(U)w(T)^{-1}$, $E(w,T)$ is row and column strict, and $w, v(\lambda,r)$ satisfy (2.6) or (2.7). The inverse insertion algorithm presented in this section defines rc-graphs $U \to T$ and $Y(R,T)$.

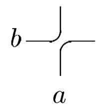
5.1. Sequence of inverse steps. The inverse insertion algorithm performs the same steps as the insertion algorithm but in the opposite order. Each step will be either an inverse row-to-row step, an inverse insertion step or an inverse rectification.

Each step (ℓ,i) with $1\leqslant i\leqslant m_\ell$ constructs an rc-graph $R(\ell,i-1)$ with no crossings above row ℓ and a tableau of transposition $T(\ell,i-1)$ filled up to $i-1$. Each step $(\ell,0)$ defines an integer $m_{\ell+1}$, an rc-graph $R(\ell+1,m_{\ell+1})$ with no crossings above row $\ell+1$ and a tableau of transpositions $T(\ell+1,m_{\ell+1})$. Conditions (3.4), (3.5) always hold.

5.2. Start of the algorithm. Set $m_1 = |T|$, $R(1,m_1) = U$ and $T(1,m_1) = T$.

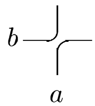
5.3. Inverse insertion step. Consider step (ℓ,i) with $i>0$. We need to construct $R(\ell,i-1)$ and $T(\ell,i-1)$. Let (cd) be the entry of box i of $T(\ell,i)$. By Lemma 3.1, strand c and d intersect in $R(\ell,i)$ at some place (ℓ_0,j_0). If $\ell=\ell_0$ define T'' by removing the entry of box i from $T(\ell,i)$. Define R'' by removing the crossing of strands c and d in $R(\ell,i)$ from place (ℓ,j_0). We say that an insertion into place (ℓ,j) is allowed if strands a,b pass this place as shown in Fig. 8. If there are no places (ℓ,j) where an insertion is allowed with $j>j_0$, this step is an inverse insertion. It sets $R(\ell,i-1) = R''$ and $T(\ell,i-1) = T''$.

5.4. Inverse rectification. All steps (ℓ,i) with $i>0$ which are not inverse insertions are inverse rectifications.

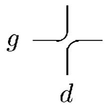


with $a \leq r < b$,

Fig. 8. A place where an insertion is allowed.



with $a \leq r < b$,



where (eg) is the entry of box $i-1$ of T''

Fig. 9. Places, where insertions are allowed during the inverse rectification.

Adopt the notation from the previous section. If $\ell_0 \neq \ell$, define $T(\ell, i-1)$ by emptying box i of $T(\ell, i)$ and define $R(\ell, i-1)$ by removing the crossing of strands c and d. Then move on to the next step, except for the case when (ef) is the entry of $i-1$ of $T(\ell, i)$, $c = e$, and $\ell \in f \boxplus d$. In this case define R'' by removing the crossing of b and f and define T'' by emptying box i of $T(\ell, i)$ and placing $(ed) = (cd)$ in box $i-1$. If $\ell_0 = \ell$, define R' and T' as it was done in the previous section.

Once R'' and T'' are constructed we say that insertion into places in row ℓ shown in Fig. 9 are allowed. Find the leftmost place where an insertion is allowed to the right of place (ℓ, j_0). Insert a crossing into this place to define R'. If this place looks like the first picture of Fig. 9, add (ab) to box i of T'' to construct T', otherwise insert (ed) and (eg) into boxes $i-1$ and i of T'' to produce T'.

Once R' and T' are constructed let (ab) be the entry of box i of T'. Then it can be shown that strands a and b intersect below row ℓ. Remove this crossing to produce $R(\ell, i-1)$ and construct $T(\ell, i-1)$ by emptying box i of T'.

5.5. Inverse row-to-row steps. Steps $(\ell, 0)$ are inverse row-to-row steps. They define m_ℓ to be the number of inverse rectifications $(\ell, i_1), \ldots, (\ell, i_{m_\ell+1})$ for row ℓ. Also each step $(\ell, 0)$ sets $R(\ell+1, m_{\ell+1}) = R(\ell, m_\ell)_{\leqslant \ell+1}$.

The shape of $T(\ell+1, m_{\ell+1})$ is the subdiagram of $sh(T(\ell, m_\ell))$ consisting of boxes $(i_1, \ldots, i_{m_{\ell+1}})$. By construction this will be a Young diagram. The entry $(a_k b_k)$ of box k of $T(\ell+1, m_{\ell+1})$ is determined by

$$w(R(\ell, i_{k-1})_{\leqslant \ell+1}) = w(R(\ell, i_k)_{\leqslant \ell+1})t_{a_k b_k}.$$

5.6. End of the inverse algorithm. Set $U \to T = R(r, 0)$. We will define $Y(U, T)$ by presenting its word. Set $word_{r+1}$ to be empty. Define $word_\ell$ of length m_ℓ by adding letters ℓ to $word_{\ell+1}$ as follows. If $(\ell, i_1), \ldots, (\ell, i_{m_\ell+1})$ are the rectification steps for row ℓ. Then set letter i_k of $word_\ell$ to be the same as letter k of $word_{\ell+1}$, set all the other letters of $word_\ell$ to be equal to ℓ. Finally, set $word(Y(U, T)) = word_1$.

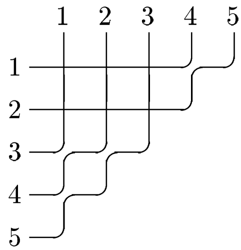
6. Examples

6.1. Example of rc-graphs $S_j(\mathscr{P})$. Assume R and T are given in Fig. 10. Define $w = w(R)w(T)^{-1} = (2, 1, 4, 3, 5, 6, \ldots)$. Then $\mathscr{P} = (w, R, T)$ is an r-Bruhat package.

Then the sequence $\{S_j(\mathscr{P})\}$ is given in Fig. 11. In each graph $S_j(\mathscr{P})$ the circled crossing needs to be removed to construct $S_{j-1}(\mathscr{P})$. Since $word(T) = (14)(23)(25)(15)$, S_3 is constructed out of S_4 by removing the crossing of strands 1 and 5, S_2 out of S_3 by removing the crossing of strands 2 and 5, and so on.

6.2. Example of the insertion algorithm in the case (2.6). From now on we draw only crossings of rc-graphs without drawing strands, as it was done in Fig. 1. It makes it easier to see how rc-graphs change during the algorithm. At the same time, as usual, we assume each rc-graph extends infinitely down and to the right and the omitted parts of rc-graphs have no crossings.

Assume R and Y are given in Fig. 12, so that $r = 3$, $w(R) = (1, 4, 3, 2, 5, 6, \ldots)$ is a 3-semi-shuffle and $w(Y) = (1, 4, 5, 2, 3, 6, \ldots) = v((2, 2), 3)$ is a 3-shuffle. We will illustrate all the steps of the algorithm for $R \leftarrow Y$.



25	15
14	23

Fig. 10. Rc-graph R and tableau of transpositions T.

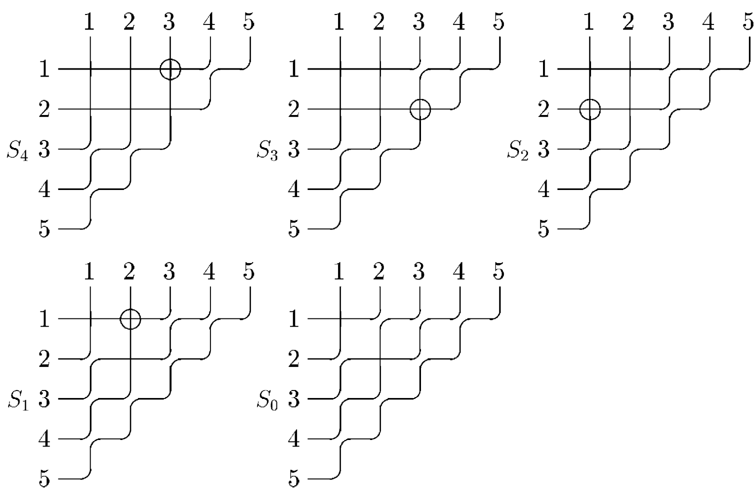


Fig. 11. Rc-graphs $S_4(\mathscr{P})$ through $S_0(\mathscr{P})$.

$$R \quad \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & + & + \\ 2 & \bullet & + & \bullet \\ 3 & \bullet & \bullet & \bullet \end{array} \qquad\qquad Y \quad \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & \bullet & + \\ 2 & + & \bullet & + \\ 3 & + & \bullet & \bullet \end{array}$$

Fig. 12. Rc-graph R and Y.

$$\begin{array}{cc} R(3,0) & T(3,0) \\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & \bullet & \bullet \\ 2 & \bullet & \bullet & \bullet \\ 3 & \bullet & \bullet & \bullet \end{array} & \begin{array}{|c|}\hline \\ \hline \end{array} \end{array} \;\Bigg|\; \begin{array}{cc} R(3,1) & T(3,1) \\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & \bullet & \bullet \\ 2 & \bullet & \bullet & \bullet \\ 3 & \oplus & \bullet & \bullet \end{array} & \begin{array}{|c|}\hline 34 \\ \hline \end{array} \end{array} \;\Bigg|\; \begin{array}{cc} R(2,0) & T(2,0) \\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & \bullet & \bullet \\ 2 & \bullet & + & \bullet \\ 3 & \bullet & \bullet & \bullet \end{array} & \begin{array}{|c|c|}\hline & \\ \hline \multicolumn{1}{|c|}{} \\ \cline{1-1} \end{array} \end{array}$$

Fig. 13. Steps (3,0), (3,1) and (2,0).

$$\begin{array}{cc} R(2,1) & T(2,1) \\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & \bullet & \bullet \\ 2 & \bullet & + & \oplus \\ 3 & \bullet & \bullet & \bullet \end{array} & \begin{array}{|c|c|}\hline & \\ \hline \multicolumn{1}{|c|}{35} \\ \cline{1-1} \end{array} \end{array} \;\Bigg|\; \begin{array}{cc} R(2,2) & T(2,2) \\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & \bullet & \bullet \\ 2 & \oplus\!\leftarrow & \odot & + \\ 3 & \oplus & \bullet & \bullet \end{array} & \begin{array}{|c|c|}\hline 24 & \\ \hline \multicolumn{1}{|c|}{35} \\ \cline{1-1} \end{array} \end{array} \;\Bigg|\; \begin{array}{cc} R(2,3) & T(2,3) \\ \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & \bullet & \bullet & \bullet & \bullet \\ 2 & + & \bullet & + & \oplus \\ 3 & + & \bullet & \bullet & \bullet \end{array} & \begin{array}{|c|c|}\hline 24 & 36 \\ \hline \multicolumn{1}{|c|}{35} \\ \cline{1-1} \end{array} \end{array}$$

Fig. 14. Steps (2,1), (2,2) and (2,3).

$$\begin{array}{cc} R(1,0) & T(1,0) \\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & + & + \\ 2 & \bullet & \bullet & \bullet \\ 3 & + & \bullet & \bullet \end{array} & \begin{array}{|c|c|}\hline & \\ \hline & \\ \hline \end{array} \end{array} \;\Bigg|\; \begin{array}{cc} R(1,1) & T(1,1) \\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & + & + \\ 2 & \bullet & \bullet & \oplus \\ 3 & + & \bullet & \bullet \end{array} & \begin{array}{|c|c|}\hline & \\ \hline 35 & \\ \hline \end{array} \end{array} \;\Bigg|\; \begin{array}{cc} R(1,2) & T(1,2) \\ \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \bullet & + & + & \bullet & \oplus \\ 2 & \bullet & \bullet & + & \bullet & \bullet \\ 3 & + & \bullet & \bullet & \bullet & \bullet \end{array} & \begin{array}{|c|c|}\hline & \\ \hline 35 & 36 \\ \hline \end{array} \end{array}$$

Fig. 15. Steps (1,0), (1,1) and (1,2).

$$\begin{array}{cc} R(1,3) & T(1,3) \\ \begin{array}{c|ccccc} & 1 & 2 & 3 & & \\ \hline 1 & \oplus\!\leftarrow & \odot & + & \bullet & + \\ 2 & \oplus & \bullet & + & \bullet & \bullet \\ 3 & + & \bullet & \bullet & \bullet & \bullet \end{array} & \begin{array}{|c|c|}\hline 14 & \\ \hline 35 & 36 \\ \hline \end{array} \end{array} \;\Bigg|\; \begin{array}{cc} R(1,4) & T(1,4) \\ \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & + & \bullet & + & \oplus\!\leftarrow & \odot \\ 2 & + & \bullet & + & \oplus & \bullet \\ 3 & + & \bullet & \bullet & \bullet & \bullet \end{array} & \begin{array}{|c|c|}\hline 14 & 26 \\ \hline 35 & 36 \\ \hline \end{array} \end{array}$$

Fig. 16. Step (1,3) and the final step (1,4).

$$R\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & \bullet & + \\ 2 & \bullet & \bullet & \bullet \\ 3 & \bullet & + & + \end{array} \qquad Y\ \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & \bullet & + & \bullet & + \\ 2 & \bullet & \bullet & \bullet & \bullet \\ 3 & + & \bullet & \bullet & \bullet \end{array}$$

Fig. 17. Rc-graph R and Y.

$$\begin{array}{c}R(3,0)\\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & \bullet & \bullet \\ 2 & \bullet & \bullet & \bullet \\ 3 & \bullet & + & + \end{array}\end{array} \quad \begin{array}{c}T(3,0)\\ \boxed{}\end{array} \quad \Bigg| \quad \begin{array}{c}R(3,1)\\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & \bullet & \bullet \\ 2 & \bullet & \bullet & \bullet \\ 3 & \oplus & + & + \end{array}\end{array} \quad \begin{array}{c}T(3,1)\\ \boxed{34}\end{array} \quad \Bigg| \quad \begin{array}{c}R(2,0)\\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & \bullet & \bullet \\ 2 & \bullet & \bullet & \bullet \\ 3 & \bullet & + & + \end{array}\end{array} \quad \begin{array}{c}T(2,0)\\ \boxed{}\end{array}$$

Fig. 18. Steps (3,0), (3,1) and (2,0).

$$\begin{array}{c}R(2,1)\\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \bullet & \bullet & \bullet \\ 2 & \bullet & \bullet & \bullet \\ 3 & \oplus & + & + \end{array}\end{array} \quad \begin{array}{c}T(2,1)\\ \boxed{34}\end{array} \quad \Bigg| \quad \begin{array}{c}R(1,0)\\ \begin{array}{c|ccc} & & & \\ \hline 1 & \bullet & \bullet & + \\ 2 & \bullet & \bullet & \bullet \\ 3 & \bullet & + & + \end{array}\end{array} \quad \begin{array}{c}T(1,0)\\ \begin{array}{|c|c|}\hline & \\ \hline \\ \cline{1-1}\end{array}\end{array} \quad \Bigg| \quad \begin{array}{c}R(1,1)\\ \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & \bullet & \bullet & + & \oplus \\ 2 & \bullet & \bullet & \bullet & \bullet \\ 3 & \bullet & + & + & \bullet \end{array}\end{array} \quad \begin{array}{c}T(1,1)\\ \begin{array}{|c|c|}\hline & \\ \hline 36 \\ \cline{1-1}\end{array}\end{array}$$

Fig. 19. Steps (2,1), (1,0) and (1,1).

$$\begin{array}{c}R(1,2)\\ \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & \bullet & \oplus & + & \odot \\ 2 & \bullet & \bullet & \bullet & \bullet \\ 3 & \oplus & + & + & \bullet \end{array}\end{array} \quad \begin{array}{c}T(1,2)\\ \begin{array}{|c|c|}\hline 24 & \\ \hline 34 \\ \cline{1-1}\end{array}\end{array} \quad \Bigg| \quad \begin{array}{c}R(1,3)\\ \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \bullet & + & + & \bullet & \bullet & \oplus \\ 2 & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ 3 & + & + & + & \bullet & \bullet & \bullet \end{array}\end{array} \quad \begin{array}{c}T(1,3)\\ \begin{array}{|c|c|}\hline 24 & 37 \\ \hline 34 \\ \cline{1-1}\end{array}\end{array}$$

Fig. 20. Step (1,2) and the final step (1,3).

Figs. 13–16 show rc-graphs $R(\ell, i)$ and tableaux of transposition $T(\ell, i)$. Steps (3,0), (2,0) and (1,0) are row-to-row steps. Steps (3,1), (2,1), (2,3) and (1,2) are insertion steps. Steps (2,2), (1,1), (1,3) and (1,4) are rectifications. We circle all crossings of $R(\ell, i)$ with $i > 0$ which are removed or added by the current step. We also show by an arrow how crossings move during rectifications.

Let us also recall that each row-to-row step $(\ell, 0)$ constructs the sequence of rc-graphs $S_j(\mathscr{P}(\ell+1))$ and then defines $R(\ell, 0) = S_0(\mathscr{P}(\ell+1))$. We omit the details of this construction and refer to Section 6.1 for an example. Also, keep in mind that

$$R\quad \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ 1 & \bullet & \bullet & + & + \\ 2 & \bullet & \bullet & + & \bullet \\ 3 & \bullet & \bullet & + & \bullet \end{array} \qquad Y\quad \begin{array}{c|ccc} & 1 & 2 & 3 \\ 1 & \bullet & \bullet & \bullet \\ 2 & + & + & \bullet \\ 3 & \bullet & \bullet & \bullet \end{array}$$

Fig. 21. Rc-graph R and Y.

$$\begin{array}{cc} R(2,0) & T(2,0) \\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ 1 & \bullet & \bullet & \bullet \\ 2 & \bullet & \bullet & + \\ 3 & \bullet & \bullet & + \end{array} & \boxed{}\boxed{} \end{array} \;\Bigg|\; \begin{array}{cc} R(2,1) & T(2,1) \\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ 1 & \bullet & \bullet & \bullet \\ 2 & \oplus & \bullet & + \\ 3 & \bullet & \bullet & + \end{array} & \boxed{23}\boxed{} \end{array} \;\Bigg|\; \begin{array}{cc} R(2,2) & T(2,2) \\ \begin{array}{c|ccc} & 1 & 2 & 3 \\ 1 & \bullet & \bullet & \bullet \\ 2 & + & \oplus & + \\ 3 & \bullet & \bullet & + \end{array} & \boxed{23}\boxed{24} \end{array}$$

Fig. 22. Steps (2,0), (2,1) and (2,2).

$$\begin{array}{cc} R(1,0) & T(1,0) \\ \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ 1 & \bullet & \bullet & + & + \\ 2 & \bullet & \bullet & + & \bullet \\ 3 & \bullet & \bullet & + & \bullet \end{array} & \boxed{}\boxed{} \end{array} \;\Bigg|\; \begin{array}{cc} R(1,1) & T(1,1) \\ \begin{array}{c|cccc} & 1 & 2 & 3 & \\ 1 & \bullet & \bullet & + & + \\ 2 & \oplus & \bullet & + & \bullet \\ 3 & \bullet & \bullet & + & \bullet \end{array} & \boxed{23}\boxed{} \end{array} \;\Bigg|\; \begin{array}{cc} R(1,2) & T(1,2) \\ \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ 1 & \bullet & \oplus & + & \odot \\ 2 & + & \oplus & + & \bullet \\ 3 & \bullet & \bullet & + & \bullet \end{array} & \boxed{24}\boxed{23} \end{array}$$

Fig. 23. Steps (1,0), (1,1) and final step (1,2).

after each row-to-row step $w(R(\ell,0)) = w(R_{\leqslant \ell})$, but the rc-graphs $R(\ell,0)$ and $R_{\leqslant \ell}$ could be different. For example, see step (1,0) in Fig. 15.

6.3. Example of the insertion algorithm in the case (2.7). Let us now present an example in the case when the shape of Y is a hook. Let R and Y be shown in Fig. 17. In particular, both $w(R) = (1,2,4,6,3,5,7,8,\ldots)$ and $w(Y) = (1,3,5,2,4,6,7,\ldots)$ are shuffles, but $w(R)$ has its descent at 4, while $w(Y)$ has its descent at 3, so case (2.6) does not apply.

Figs. 18–20 contain the results of all the steps of the algorithm. Steps (3,0), (2,0) and (1,0) are row-to-row steps, steps (3,1), (1,1) and (1,3) are insertion steps, while (2,1) and (1,2) are rectifications. Notice that step (1,2) is the only step where the second situation of Fig. 5 occurs.

6.4. Another example in case (2.7). The last example is for R and Y defined in Fig. 21. In this case $w(R) = (1,2,5,4,6,3,7,8,\ldots)$ and $w(Y) = (1,4,2,3,5,6,\ldots) = v((2,0),2)$, the shape of Y is a row, while $w(R)$ is a permutation with two descents.

The steps of the algorithm are shown in Figs. 22 and 23. Steps (2,0) and (1,0) are row-to-row steps, steps (2,1) and (2,2) are insertion steps, while steps (1,1) and (1,2) are rectifications. Notice that step (1,2) is the only step where the second case of Fig. 6 occurs.

Acknowledgments

The author thanks Frank Sotille, Sara Billey and Abhinav Kumar for useful discussions as well as Cristian Lenart and both referees for making valuable remarks about the paper.

References

[1] N. Bergeron, S. Billey, *RC*-graphs and Schubert polynomials, Exp. Math. 2 (1993) 257–269.
[2] N. Bergeron, F. Sottile, Schubert polynomials, the Bruhat order, and the geometry of flag manifolds, Duke Math. J. 95 (2) (1998) 373–423.
[3] S. Billey, W. Jockusch, R. Stanley, Some combinatorial properties of Schubert polynomials, J. Algebraic Combin. 2 (4) (1993) 345–374.
[4] S. Fomin, A.N. Kirillov, Yang-Baxter equation, symmetric functions, and Schubert polynomials, Discrete Math. 153 (1996) 123–143.
[5] S. Fomin, R. Stanley, Schubert polynomials and the nil-Coxeter algebra, Adv. in Math. 103 (2) (1994) 196–207.
[6] W. Fulton, Young Tableaux, in: London Mathematical Society Student Texts, Vol. 35, Cambridge University Press, Cambridge, 1997.
[7] A. Knutson, Descent-cycling in Schubert calculus, Exp. Math. 10 (2001) 345–353.
[8] M. Kogan, *RC*-graphs and a generalized Littlewood–Richardson rule, Internat. Math. Res. Notices, Vol. 15, (2001) 765–782.
[9] M. Kogan, A. Kumar, A proof of Pieri's formula using generalized Schensted insertion algorithm for rc-graphs, Proc. Amer. Math. Soc. 130 (9) (2002) 2525–2534.
[10] A. Kumar, A generalized insertion algorithm for *rc*-graphs to prove Pieri's column multiplication rule, unpublished note.
[11] A. Lascoux, M.P. Schützenberger, Polynômes de Schubert, C. R. Acad. Sci. Paris 294 (1982) 447–450.
[12] I.G. Macdonald, Symmetric Functions and Hall Polynomials, 2nd Edition, With contributions by A. Zelevinsky, Oxford Mathematical Monographs, Oxford Science Publications, The Clarendon Press, Oxford University Press, New York, 1995.
[13] I.G. Macdonald, Notes on Schubert Polynomials, Publications du LACIM, Vol. 6, Universite du Quebec a Montreal, 1991.
[14] L. Manivel, Fonctions symtriques, polynmes de Schubert et lieux de dgnrescence, in: Cours Spcialiss, Vol. 3, Socit Mathmatique de France, Paris, 1998.
[15] C. Schensted, Longest increasing and decreasing subsequences, Canad. J. Math. 13 (1961) 179–191.
[16] F. Sottile, Pieri's formula for flag manifolds and Schubert polynomials, Ann. Inst. Fourier (Grenoble) 46 (1) (1996) 89–110.