A LITTLEWOOD-RICHARDSON RULE FOR DUAL STABLE GROTHENDIECK POLYNOMIALS

PAVEL GALASHIN

Abstract. For a given skew shape, we build a crystal graph on the set of all reverse plane partitions that have this shape. As a consequence, we get a simple analogue of the Littlewood-Richardson rule for the expansion of the corresponding dual stable Grothendieck polynomial in terms of Schur polynomials.

1. Introduction

Dual stable Grothendieck polynomials $g_{\lambda/\mu}(x)$ were introduced in [5]. They are Hopf-dual to the stable Grothendieck polynomials, which represent some classes of the structure sheaves of Schubert varieties. The connection between them and the $K$-theory of the Grassmannian has been discussed in various papers including [6, 2, 1, 5]. The paper [5] gives an explicit combinatorial rule for the coefficients of polynomials $g_\lambda(x)$ in the basis of Schur polynomials $s_\mu(x)$. We extend this result to the case of $g_{\lambda/\mu}(x)$ for a skew shape $\lambda/\mu$, and give a different rule that provides the same coefficients for straight shapes and extends the classical Littlewood-Richardson rule. We do this by constructing a crystal graph (see [4]) on the set $\mathcal{R}(\lambda/\mu)$ of all reverse plane partitions of shape $\lambda/\mu$ with entries not exceeding a fixed number $m > 0$.

1.1. Main results. To a reverse plane partition $T \in \mathcal{R}(\lambda/\mu)$ we assign a reading word $r(T)$ in the following way: ignore each entry of $T$ that is equal to the entry directly below it; then read all the remaining entries in the left-to-right bottom-to-top order (the usual reading order for the Young tableaux). After that we define a family of operators $e_1, e_2, \ldots, e_{m-1}$ on the set $\mathcal{R}(\lambda/\mu)$ which are essentially the usual parentheses operators applied to the reading word.

Theorem 1. The operators $e_1, e_2, \ldots, e_{m-1}$ satisfy the crystal axioms (which can be found in [4] and will also be discussed in the sequel).

Date: January 5, 2015.
Therefore we get a crystal graph structure on $\mathcal{R}(\lambda/\mu)$. As a direct application of that, we get a Littlewood-Richardson rule for the reverse plane partitions:

**Corollary 2.** The dual stable Grothendieck polynomial $g_{\lambda/\mu}(x)$ is expanded in terms of Schur polynomials $s_{\nu}(x)$ as follows:

$$g_{\lambda/\mu}(x) = \sum_{\nu} h_{\lambda/\mu}^{\nu} s_{\nu}(x),$$

where the sum is over all Young diagrams $\nu$, and the coefficient $h_{\lambda/\mu}^{\nu}$ is equal to the number of reverse plane partitions $T$ of shape $\lambda/\mu$ and weight $\nu$ such that the reading word $r(T)$ is a lattice word.

We also give a self-contained proof of this Corollary without using the theory of crystal graphs. Note that the highest degree homogeneous component of $g_{\lambda/\mu}(x)$ is the skew-Schur polynomial $s_{\lambda/\mu}(x)$, so Corollary 2 is an extension of the Littlewood-Richardson rule for skew-Schur polynomials.

**1.2. Previous research.** There already is a combinatorial rule for the coefficients $h_{\lambda/\mu}^{\nu}$ in [5] for the case when $\mu = \emptyset$ and $\lambda/\mu = \lambda$ is a straight shape. Namely, $h_{\lambda}^{\nu}$ equals to the number $f_{\lambda}^{\nu}$ of *elegant fillings* of $\lambda/\nu$, that is, the number of semi-standard Young tableaux $T$ of shape $\lambda/\nu$ such that all entries in the $i$-th row of $T$ are strictly less than $i$. To prove this rule, Lam and Pylyavskyy in [5] construct a weight preserving bijection between reverse plane partitions of shape $\lambda$ and pairs $(S, U)$, where $S$ is a semi-standard Young tableau of some shape $\mu$ and $U$ is an elegant filling of $\lambda/\mu$. Following this bijection one can deduce that $T$ is a reverse plane partition of shape $\lambda$ and weight $\nu$ whose reading word is a lattice word if and only if it corresponds to a pair $(S, U)$ such that $S$ is the filling of the shape $\nu$ with all entries in the $i$-th row equal to $i$, and $U$ is an elegant tableau of shape $\lambda/\nu$. Therefore the bijection from [5] restricted to the reverse plane partitions whose reading word is a lattice word proves the equality of the numbers $h_{\lambda}^{\nu}$ and $f_{\lambda}^{\nu}$.

**1.3. The structure of the paper.** The rest of this section contains some background information about dual stable Grothendieck polynomials, crystal graphs and introduces the operators $e_i$ that occur in the statement of Theorem 1.

The second section is dedicated to the proof of Theorem 1 and Corollary 2 by exploring further properties and connections between the reading words of reverse plane partitions and the action of operators $e_i$. 
1.4. Preliminaries.

1.4.1. Reverse plane partitions. We follow the notations of [5]. Let $\lambda/\mu$ be a skew shape. A reverse plane partition $T$ of shape $\lambda/\mu$ is a tableau of this shape such that its entries do not exceed $m$ and weakly increase both in rows and in columns. For $i \in [m] := \{1, \ldots, m\}$, by $T(i)$ we denote the number of columns of $T$ that contain $i$. To each reverse plane partition $T$ we attach a monomial $x_T = \prod_{i \in \mathbb{N}} x_{T(i)}$.

For a skew shape $\lambda/\mu$, define the dual stable Grothendieck polynomial $g_{\lambda/\mu}(x_1, \ldots, x_m)$ as the sum of weights of all reverse plane partitions $T$ of shape $\lambda/\mu$:

$$g_{\lambda/\mu}(x) = \sum_T x_T.$$  

As it was shown in [5], these polynomials are symmetric.

1.4.2. Crystal graphs. Crystal graphs are important for representation theory of certain quantized universal enveloping algebras, and have been a fruitful topic of research for the past two decades. We give a brief adaptation of the crystal graph theory based on [4, 8, 7] with a very low yet sufficient for the rest of this paper level of detail.

A crystal graph $G$ can be viewed as a set $V$ of vertices together with a set $e_1, \ldots, e_{m-1} : V \to V \cup \{0\}$ of operators that act on the vertices of $G$ and return either a vertex of $G$ or zero. In addition, these operators are required to satisfy a set of simple crystal axioms. If they do, then they are called crystal operators, and $G$ is called a crystal graph.

Instead of providing the list of these axioms, we give an important example of a crystal graph, which is the only crystal graph that we will be interested in. Fix $n > 0$. Let $S := [m]^n$ be the set of all strings of length $n$ in the alphabet $[m]$. For $s = (s_1, s_2, \ldots, s_n) \in S$, the weight $w(s) = (w_1(s), \ldots, w_m(s))$ is defined as

$$w_i(s) := \#\{j \in [n] : s_j = i\}.$$  

For $i \in [m-1]$ we define the operator $E_i : S \to S \cup \{0\}$. For $s := (s_1, s_2, \ldots, s_n) \in S$ the value $E_i(s)$ is evaluated using the following algorithm:

1. Ignore all entries of $s$ other than the ones equal to $i$ or to $i+1$;
2. Ignore all occurrences of $i+1$ immediately followed by $i$;
3. After doing the previous step as many times as possible we obtain a string that consists of several $i$'s followed by several $i+1$'s. If there is at least one $i+1$, then $E_i$ replaces the leftmost $i+1$ by a $i$, and otherwise we set $E_i(s) := 0$. 

3
In other words, $E_i$ labels each $i$ by a closing parenthesis, each $i + 1$ by an opening parenthesis, and then it replaces the leftmost unmatched opening parenthesis by a closing one if there are any unmatched opening parentheses present.

Similarly, define the operators $F_i$ to be the operators that replace the rightmost unmatched closing parenthesis by an opening one. The operators $E_i$ and $F_i$ are “inverse to each other” in the sense that for any two strings $u, v \in S$, $E_i(u) = v$ if and only if $F_i(v) = u$.

These operators satisfy the crystal axioms and therefore have a lot of nice properties, which we summarize in the following Lemma:

**Lemma 3.**

1. Each connected component of the corresponding edge-colored graph has exactly one vertex $v \in S$ such that for every $i \in [m - 1]$, $E_i(v) = 0$.
2. This component is completely determined (up to an isomorphism of edge-colored graphs) by the weight $w(v)$, which is clearly a weakly decreasing sequence of integers.
3. The sum of $x^{w(u)}$ over all vertices $u$ in this connected component is equal to the Schur polynomial $s_{w(v)}$.

Even though all of these properties follow from the fact that $E_i$ and $F_i$ satisfy crystal axioms, we prove them just to make the proof of Corollary 2 self-contained.

**Proof.** Note that if the words $u, u' \in S$ are Knuth-equivalent, then the words $E_i(u)$ and $E_i(u')$ are Knuth-equivalent (or both zero), and also the words $F_i(u)$ and $F_i(u')$ are Knuth-equivalent (or both zero). And for each word $u \in S$ there is exactly one word $u' \in S$ which is Knuth-equivalent to $u$ and such that it is a reading word of some semi-standard Young tableau $T$. But the operators $E_i$ and $F_i$ applied to the reading word of $T$ produce a reading word of some other tableau that has the same shape as $T$.

Now all three properties follow from the fact that any two semi-standard Young tableaux of the same straight shape can be obtained from one another by applying a sequence of operators $E_i$ and $F_i$. To show this, consider a tableau $T_0$ of shape $\lambda$ such that for every $j$, all of its entries in the $j$-th row are equal to $j$. Consider an integer $k \geq 1$, and let $T$ be a tableau of shape $\lambda$ such that for $j \geq k$, all entries of $T$ in the $j$-th row are equal to $j$ and such that for $j < k$, the entries of $T$ in the $j$-th row are less than or equal to $k$. Then we claim that such $T$ can be obtained from $T_0$ by applying a sequence of operators $F_i$ for different $i$’s. This statement is true for $k = 1$ and can be easily proven by induction for all $k \geq 1$. \[\Box\]
1.5. The crystal operators for reverse plane partitions.

1.5.1. The conflict-resolving algorithm. We describe the conflict-resolving algorithm for reverse plane partitions from [3], where it was used in order to describe the analogue of the Bender-Knuth involution for reverse plane partitions. Let $\lambda/\mu$ be a skew shape, and fix $i \in [m-1]$. For a tableau $T'$ of shape $\lambda/\mu$ such that the entries of $T'$ are equal to either $i$ or $i+1$ and weakly increase in columns but not necessarily in rows, we say that a column of $T'$ is $i$-uniform, if it contains a $i$ but does not contain a $i+1$. Similarly, we call a column $i+1$-uniform if it contains a $i+1$ but does not contain a $i$. If a column contains both $i$ and $i+1$, then we call this column mixed.

**Definition 4.** A tableau $T'$ is a pseudo-reverse plane partition if the entries of $T'$ weakly increase in columns and for every two mixed columns $A$ and $B$ ($A$ is to the left of $B$), the lowest $i$ in $A$ is not higher than the lowest $i$ in $B$. In other words, the vertical coordinates of the borders between $i$’s and $i+1$’s in mixed columns weakly increase from left to right.

The conflict-resolving algorithm takes a pseudo-reverse plane partition $T'$ and converts it into a reverse plane partition of the same shape and weight.

A pseudo-reverse plane partition $T'$ may easily fail to be a reverse plane partition. More specifically, it may contain a $i+1$ with a $i$ directly to the right of it – we call such a situation a conflict. Let $A$ be the column containing a $i+1$ and $A+1$ be the column containing a $i$. Then there are three possible types of conflicts depending on the types of the columns $A$ and $A+1$:

1. $A$ is $i+1$-uniform and $A+1$ is mixed
2. $A$ is mixed and $A+1$ is $i$-uniform
3. $A$ is $i+1$-uniform and $A+1$ is $i$-uniform

There is a fourth type of conflicts in which both columns are mixed, but the pseudo-reverse plane partition property implies that such conflicts are impossible. For a conflict of each of these three types, [3] provides a conflict-resolving step, which changes only the entries of $A$ and $A+1$ and resolves this conflict.

For conflicts of the first two types, the conflict-resolving step switches the roles of the columns but preserves the vertical coordinate of the lowest $i$ in the mixed column; this determines the operation uniquely. For a conflict of the third type, it simply replaces all $i$’s by $i+1$’s and vice versa in both columns. It is clear that the resulting tableau will
also be a pseudo-reverse plane partition. The conflict-resolving steps for \( i = 1 \) are visualized in Figure 1.

![Figure 1. The conflict-resolving steps.](image)

The conflict-resolving algorithm performs these conflict-resolving steps until there are no conflicts left, which means that we get a reverse plane partition. This algorithm terminates, because \( i \)-uniform columns always move to the right while \( i + 1 \)-uniform columns always move to the left. Also, it is shown in [3] that the result of the algorithm does not depend on the order in which the conflicts are resolved.

1.5.2. The definition of \( e_i \)'s and \( f_i \)'s. Let \( \lambda/\mu \) be a skew shape, and fix \( i \in [m - 1] \). For a reverse plane partition \( T \) of shape \( \lambda/\mu \), define \( e_i(T) \) as follows. First, consider only the subtableau of \( T \) that consists of entries equal to either \( i \) or \( i + 1 \). Then, label each \( i \)-uniform column by a closing parenthesis and each \( i + 1 \)-uniform column by an opening parenthesis (and ignore the mixed columns).

Choose the \((i + 1\text{-uniform})\) column \( A \) that corresponds to the leftmost unmatched opening parenthesis (if all opening parentheses are matched, set \( e_i(T) := 0 \)). Replace all the \( i + 1 \)'s in \( A \) by \( i \)'s, and then apply the conflict-resolving algorithm to the resulting pseudo-reverse plane partition.

Similarly, \( f_i \) chooses the \((i\text{-uniform})\) column \( B \) that corresponds to the rightmost unmatched closing parenthesis and replaces all the \( i \)'s in it by \( i + 1 \)'s and then applies the conflict-resolving algorithm.

We discuss the properties of \( e_i \)'s and \( f_i \)'s and their connection to the defined above reading word in the next section.

2. Properties of the reading words of reverse plane partitions

Recall that the reading word \( r(T) \) of a reverse plane partition \( T \) of shape \( \lambda/\mu \) is defined as the usual left-to-right bottom-to-top Young
tableaux reading word that ignores each entry that has the same entry below it. An example is shown in Figure 2.

![Tableaux Example](image)

**Figure 2.** The reading word of a skew-shaped reverse plane partition.

We assume the coordinates of the boxes are in the matrix notation. For a reverse plane partition $T$, define its **height vector** $h(T)$ to be the sequence of vertical coordinates of the entries of $T$ that contribute to $r(T)$ arranged in the exact same order as they appear in the reading word. For example, for $T$ as in Figure 2 we put $h(T) := (7, 7, 6, 6, 5, 4, 4, 4, 3, 3, 1)$. It is always a weakly decreasing sequence of positive integers. Similarly, we define the height vector of a pseudo-reverse plane partition. Note that each conflict-resolving step preserves the height vector, and, therefore, so do the operators $e_i$ and $f_i$.

**Lemma 5.** Fix a skew shape $\lambda/\mu$ and a sequence $h$ of positive integers. Then for each reading word $r$ there is at most one reverse plane partition $T$ of shape $\lambda/\mu$ with $r(T) = r$ and $h(T) = h$.

**Proof.** Suppose that there exists a reverse plane partition $T$ of shape $\lambda/\mu$ with $r(T) = r$ and $h(T) = h$. Then $T$ can be uniquely reconstructed from $r$ and $h$ by filling the boxes of $\lambda/\mu$ in the reading order:

1. Set $j = 1$;
2. Let $B$ be the first (in the reading order) box of $\lambda/\mu$ which is not filled with a number. Let $a$ be the value in the box directly below it, and let $c$ be the value in the box directly to the left of it (if there is no such box then we put $a := +\infty$ or $c := 0$);
3. If the height of $B$ is not equal to $h_j$, then set the entry in the box $B$ equal to $a$ and proceed to the next box (go to step 2);
4. If the number $r_j$ does not satisfy $c \leq r_j < a$, then, again, set the entry in the box $B$ equal to $a$ and proceed to the next box;
5. Otherwise, we set the entry in the box $B$ equal to $r_j$, increase $j$ by 1 and proceed to the next box.
Note that if $r$ and $h$ are the reading word and the height vector of some reverse plane partition, then the entries of $h$ weakly decrease, and the entries of $r$ that have the same height weakly increase. We prove by induction that the first $k$ entries of $T$ (in the reading order) are the same as the first $k$ entries of the reverse plane partition that the algorithm produces. For $k = 0$ it is true. Now, we want to put $r_j$ somewhere into the row $h_j$ so that the entry below it is strictly bigger than $r_j$ and so that the entries in the row weakly increase. Thus if $r_j$ cannot be put into the current box (because either $r_j \geq a$ or $r_j < b$), then this box should be ignored by the reading word, so its value should be the same as the value directly below it. If $b \leq r_j < a$, then $r_j$ has to be put into the current box, because if we put $r_j$ somewhere to the right, then we have to fill this box with the value directly below it (with $a$), which is strictly bigger than $r_j$, so the entries in the row will not be weakly increasing. □

Recall that the operators $F_i$ are defined on the set $S = [m]^n$ of all strings of length $n$, and replace the rightmost unmatched closing parenthesis (corresponding to an entry equal to $i$) by an opening parenthesis (by a $i + 1$). Meanwhile, the operators $f_i$ act on $R(\lambda/\mu)$, which is the set of all reverse plane partitions of shape $\lambda/\mu$ with entries less than or equal to $m$. It turns out that these two actions commute with the operation of taking the reading word:

**Lemma 6.** Let $T$ be a reverse plane partition. Then

$$F_i(r(T)) = r(f_i(T)).$$

In particular, if $f_i(T)$ is zero then $F_i(r(T))$ is zero and the converse is also true.

And, because $e_i$ and $f_i$ are “inverse to each other” (in the same sense as above), and the same is true for $E_i$ and $F_i$, we get

**Corollary 7.** Let $T$ be a reverse plane partition. Then

$$E_i(r(T)) = r(e_i(T)).$$

**Proof of Lemma 6.** The operator $f_i$ labels $i$-uniform columns by closing parentheses and $i + 1$-uniform columns by opening parentheses. Then it finds the rightmost unmatched closing parenthesis and replaces the corresponding $i$-uniform column by a $i + 1$-uniform column. After that we get a pseudo-reverse plane partition $T'$, and then we apply the conflict-resolving algorithm to $T'$ which produces a reverse plane partition $T'' =: f_i(T)$. Our proof consists of two parts:

1. $r(T') = r(T'')$;
\( F_i(r(T)) = r(T') \).

**Remark 8.** Note that both of these parts are false for \( e_i \) and \( E_i \). To make them true, one needs to introduce the reading word that ignores each entry equal to the entry directly above it, rather than directly below it.

We start with the first part. Note that even though \( T' \) and \( T'' \) differ by a sequence of conflict-resolving steps, it is not true in general that the conflict-resolving steps preserve the reading word. Fortunately, as we will see later, all the appearing conflicts are of the first type. And the corresponding conflict-resolving step (see Figure 3) clearly does not change the reading word.

![Figure 3. The first conflict-resolving step.](image)

The reason we only need this conflict-resolving step is the definition of \( f_i \). Namely, \( f_i \) changes only one \( i \)-uniform column \( A \) into a \( i + 1 \)-uniform column. And this column is required to be labeled by the rightmost unmatched closing parenthesis. Let \( B \) be the leftmost \( i + 1 \)-uniform column to the right of \( A \), and let \( C \) be the rightmost \( i \)-uniform column to the left of \( A \). If there was a \( i \)-uniform column between \( A \) and \( B \), then it would also be unmatched, so \( A \) would not be labeled by the rightmost unmatched closing parenthesis. Also, if there was a \( i + 1 \)-uniform column \( D \) between \( C \) and \( A \), then it would have to be matched to some \( i \)-uniform column between \( D \) and \( A \), so \( C \) would not be the rightmost \( i \)-uniform column to the left of \( A \). All in all we can see that all the columns between \( C \) and \( A \) and between \( A \) and \( B \) are mixed. If either \( C \) or \( B \) is undefined, then all the columns to the left (resp., to the right) of \( A \) are mixed.

Now it is clear why only the conflicts of the first type appear while the conflict-resolving steps are performed. The column \( A \) becomes \( i + 1 \)-uniform, so the only possible conflict can occur between \( A \) and \( A + 1 \), and as we resolve it, the \( i + 1 \)-uniform column moves to the right. But
because it is surrounded by mixed columns, the only appearing conflicts are between this \( i + 1 \)-uniform column and the mixed column to the right of it. And if this \( i + 1 \)-uniform column moves to the position \( B - 1 \), then there are no conflicts left, because \( B \) is also \( i + 1 \)-uniform. This finishes the proof of the first part.

The second part asks for a certain correspondence between two different matchings. The first one appears when we label \( i \)-uniform columns by closing parentheses, \( i + 1 \)-uniform columns by opening parentheses, and then say that two uniform columns match each other if their labels (two parentheses) match each other in the parenthesis sequence. In this situation we say that these two columns match in the reverse plane partition. The second matching appears when we label the entries of the reading word by parentheses and say that two entries of the reading word match each other if their labels match each other. In this situation we say that these two entries match in the reading word.

The second part of the Lemma states that a \( i \)-uniform column is labeled by the rightmost unmatched closing parenthesis in the reverse plane partition if and only if the corresponding entry in the reading word is also labeled by the rightmost unmatched closing parenthesis in the reading word. Here we can restrict our attention to reverse plane partitions that are filled only with \( i \)'s and \( i + 1 \)'s. For a column \( A \), let \( j(A) \) be the position of the corresponding entry of the reading word if \( A \) is either \( i \)- or \( i + 1 \)-uniform. If \( A \) is mixed, then set \( j^{-}(A) \) to be the position of the entry of the reading word corresponding to \( i \) and set \( j^{+}(A) \) to be the position of the entry of the reading word corresponding to \( i + 1 \).

We need to check three implications:

1. If a column \( A \) is \( i \)-uniform and unmatched in the reverse plane partition, then the entry \( j(A) \) is unmatched in the reading word.
2. If a column \( A \) is mixed, then the entry \( j^{-}(A) \) is matched to something (not necessarily to \( j^{+}(A) \)) in the reading word.
3. If a column \( A \) is \( i \)-uniform and matched to some \( i + 1 \)-uniform column \( B \) in the reverse plane partition, then the entry \( j(A) \) is also matched to something (not necessarily to \( j(B) \)) in the reading word.

It is clear that these three properties together imply that the \( i \)-uniform columns unmatched in the reverse plane partition correspond exactly to the unmatched \( i \)'s in the reading word. And because the reading word preserves the order of uniform columns, the second part of the lemma reduces to proving these three implications.
Note that if a column $A$ is $i$-uniform, then for every other column $B$ that is to the right (resp., left) of $A$, the entry $j(B)$ or $j^-(B)$ or $j^+(B)$ if defined is also to the right (resp., left) of $j(A)$. Another simple useful observation is that if we have any injective map that attaches to each $i + 1$ in the reading word a $i$ to the right of it, then all the $i + 1$’s in this reading word are matched. Now we are ready to check the implications (1)-(3).

(1) If a column $A$ is $i$-uniform and unmatched, then we can just throw everything to the right of $A$ and $j(A)$ out. Now, every $i + 1$-uniform column to the left of $A$ is matched to something in the reverse plane partition, so for every $i + 1$ to the left of $j(A)$ in the reading word we have a $i$ that is between it and $j(A)$, and for different $i + 1$’s these $i$’s are also different. Therefore every $i + 1$ to the left of $j(A)$ is matched in the reading word as well, so $j(A)$ is unmatched in the reading word.

(2) Suppose $A$ is mixed. If we throw out all the columns that are to the right of $A$, then several $i + 1$’s between $j^+(A)$ and $j^-(A)$ will be thrown out of the reading word, but all the $i$’s to the left of $j^-(A)$ will remain untouched. Let $B$ be the rightmost $i$-uniform column to the left of $A$. Now we throw out all the columns to the left of $B$ and also $B$ itself, which corresponds to erasing the part the reading word from the beginning to $j(B)$ (if there was no such $B$ then we do not throw anything out of the reading word). Now we have a reverse plane partition that contains no $i$-uniform columns, so by the counting argument $j^-(A)$ is matched in the reading word. But then it was also matched in the original reading word.

(3) Suppose $A$ is $i$-uniform and is matched in the reverse plane partition to some $i + 1$-uniform column $B$ to the left of $A$. Let $C$ be the rightmost $i$-uniform column to the left of $B$. We throw out everything that is to the right of $A$ or to the left of $C$, which corresponds to keeping all the entries of the reading word between $j(C)$ and $j(A)$. We also remove $C$ and $j(C)$. All the $i$-uniform columns between $A$ and $B$ are matched in the reverse plane partition to some $i + 1$-uniform columns between $A$ and $B$, and there are no $i$-uniform columns between $B$ and $C$, so the number of $i + 1$’s between $j(C)$ and $j(A)$ is strictly bigger than the number of $i$’s between $j(C)$ and $j(A)$, so $j(A)$ has to be matched to something in the reading word. We finish the proof of the third implication, which finishes the proof of the second (last) part of the Lemma.

Let $\lambda/\mu$ be a skew shape, and let $h$ be a sequence of positive integers. Lemmas 5 and 6 give a vertex-injective map from the graph of all reverse plane partitions $T$ of shape $\lambda/\mu$ with $h(T) = h$ to the graph $S$.
of all strings of the same length as $h$, and this map takes the operators $e_i$ and $f_i$ to $E_i$ and $F_i$. Therefore each connected component of the graph of all reverse plane partitions is isomorphic to the corresponding connected component of the graph $S$. Now the proof of Theorem 1 follows from the observations about crystal graphs made in Subsection 1.4.2 in particular, the proof of Corollary 2 follows from Lemma 3.

ACKNOWLEDGMENTS

I am grateful to Prof. Alex Postnikov and to Darij Grinberg for their valuable remarks.

REFERENCES

[1] Anders Skovsted Buch. A Littlewood-Richardson rule for the $K$-theory of Grassmannians. *Acta Math.*, 189(1):37–78, 2002.
[2] Sergey Fomin and Anatol N. Kirillov. The Yang-Baxter equation, symmetric functions, and Schubert polynomials. In *Proceedings of the 5th Conference on Formal Power Series and Algebraic Combinatorics (Florence, 1993)*, volume 153, pages 123–143, 1996.
[3] Pavel Galashin, Darij Grinberg, and Gaku Liu. A Bender-Knuth involution for reverse plane partitions. *in preparation*, 2015.
[4] Masaki Kashiwara. On crystal bases. In *Representations of groups (Banff, AB, 1994)*, volume 16 of *CMS Conf. Proc.*, pages 155–197. Amer. Math. Soc., Providence, RI, 1995.
[5] Thomas Lam and Pavlo Pylyavskyy. Combinatorial Hopf algebras and $K$-homology of Grassmannians. *Int. Math. Res. Not. IMRN*, (24):Art. ID rnm125, 48, 2007.
[6] Alain Lascoux and Marcel-Paul Schützenberger. Structure de Hopf de l’anneau de cohomologie et de l’anneau de Grothendieck d’une variété de drapeaux. *C. R. Acad. Sci. Paris Sér. I Math.*, 295(11):629–633, 1982.
[7] Peter Littelmann. A Littlewood-Richardson rule for symmetrizable Kac-Moody algebras. *Invent. Math.*, 116(1-3):329–346, 1994.
[8] John R. Stembridge. A local characterization of simply-laced crystals. *Trans. Amer. Math. Soc.*, 355(12):4807–4823 (electronic), 2003.

Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, 02139
E-mail address: galashin@mit.edu