Constructing Maximal Pipedreams of Double Grothendieck Polynomials

Chen-An Chou Tianyi Yu

Submitted: Dec 22, 2023; Accepted: Jun 21, 2024; Published: Aug 9, 2024

Abstract

Pechenik, Speyer and Weigandt defined a statistic $\mathsf{rajcode}(\cdot)$ on permutations which characterizes the leading monomial in top degree components of double Grothendieck polynomials. Their proof is combinatorial: They showed there exists a unique pipedream of a permutation w with row weight $\mathsf{rajcode}(w)$ and column weight $\mathsf{rajcode}(w^{-1})$. They proposed the problem of finding a "direct recipe" for this pipedream. We solve this problem by providing an algorithm that constructs this pipedream via ladder moves.

Mathematics Subject Classifications: 05E05

1 Introduction

Let S_n be the set of permutations of $\{1, \dots, n\}$. For $w \in S_n$, the matrix Schubert variety X_w is a determinantal variety that has been studied extensively (see for instance [FUL92, KM05, KMY09, WY18]). Castelnuovo-Mumford regularity measures the algebraic complexity of varieties. Since matrix Schubert varieties are Cohen-Macaulay [FUL92, KM05, Ram85], the Castelnuovo-Mumford regularity of X_w is the difference between the top and bottom degree of its K-polynomial. By the work of Knutson and Miller [KM05], the K-polynomial of X_w is the Grothendieck polynomial $\mathfrak{G}_w(\mathbf{x})$. This family of polynomials, introduced by Lascoux and Schützenberger [LS82], represents K-classes of structure sheaves of Schubert varieties in flag varieties. Their lowest degree components are the Schubert polynomials whose degrees are known.

Consequently, determining the Castelnuovo-Mumford regularity of X_w reduces to computing the degree of $\mathfrak{G}_w(\mathbf{x})$. With this motivation, there has been a recent surge in the study of top degree components of $\mathfrak{G}_w(\mathbf{x})$ [DMSD22, Haf22, PSW21, PY23, RRR+21, RRW23]. Pechenik, Speyer, and Weigandt [PSW21] defined a statistic $\mathsf{rajcode}(\cdot)$ on S_n using increasing subsequences of permutations. They showed $x^{\mathsf{rajcode}(w)}$ is the leading

Department of Mathematics, UC San Diego, La Jolla, CA, U.S.A. (c1chou@ucsd.edu, tiy059@ucsd.edu).

monomial in the top degree components of $\mathfrak{G}_w(\mathbf{x})$ with respect to the lexicographical order where $x_n > \cdots > x_1$. Pan and Yu [PY23] found a diagrammatic formula to compute $\mathsf{rajcode}(w)$ (see Definition 12).

For $w \in S_n$, the double Grothendieck polynomial $\mathfrak{G}_w(\mathbf{x}, \mathbf{y})$ involves two sets of variables: $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$. It represents Schubert classes in the torus-equivariant K-theory of flag varieties. After setting $y_1 = y_2 = \cdots = 0$, the double Grothendieck polynomial $\mathfrak{G}_w(\mathbf{x}, \mathbf{y})$ specializes to the usual Grothendieck polynomial $\mathfrak{G}_w(\mathbf{x})$. The $\mathsf{rajcode}(\cdot)$ statistic also captures the leading monomial in top degree components of $\mathfrak{G}_w(\mathbf{x}, \mathbf{y})$.

Theorem 1 ([PSW21, Theorem 1.4])**.** *The leading monomial of top degree components of $\mathfrak{G}_w(\boldsymbol{x}, \boldsymbol{y})$ is $x^{\mathsf{rajcode}(w)} y^{\mathsf{rajcode}(w^{-1})}$ with coefficient 1 for any term order with $x_n > \cdots > x_1$ and $y_n > \cdots > y_1$.*

A combinatorial formula of $\mathfrak{G}_w(\mathbf{x}, \mathbf{y})$ is given by pipedreams [BB93, BJS93, FK94, KM05]: certain tilings of a staircase grid using $\boxplus$,  and  (see Definition 5). The row (resp. column) weight of a pipedream is a weak composition where the i^{th} entry is the number of $\boxplus$ in row (resp. column) i of the pipedream. Let $\mathsf{PD}(w)$ be the set of pipedreams associated with the permutation w. Pechenik, Speyer, and Weigandt established Theorem 1 by showing there exists a unique pipedream in $\mathsf{PD}(w)$ with row weight $\mathsf{rajcode}(w)$ and column weight $\mathsf{rajcode}(w^{-1})$, which they call the maximal pipedream of w. In Remark 7.2, they said:

> "We find it frustrating that we do not have a direct recipe for the maximal pipe dream in terms of w."

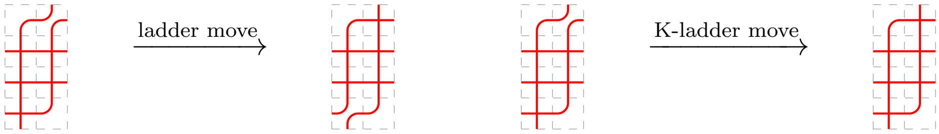
The main goal of this paper is to relieve their frustration: We give an explicit algorithm to construct the maximal pipedream $\widehat{P}(w) \in \mathsf{PD}(w)$.

Theorem 2. *For $w \in S_n$, the pipedream $\widehat{P}(w)$ we construct has row weight $\mathsf{rajcode}(w)$ and column weight $\mathsf{rajcode}(w^{-1})$.*

Our algorithm involves a local move. When row r column c of a pipedream P is $\boxplus$, we write $(r, c) \in P$. We may apply the move on a $\boxplus$ in row r column c of a pipedream P if all the following are satisfied:

- $(r, c+1) \notin P$.
- There exists $r' < r$ such that $(r', c) \notin P$ and $(r', c+1) \notin P$. In addition, $(i, c), (i, c+1) \in P$ for any $r' < i < r$.

Now we perform the move at the $\boxplus$ in row r column c of P. First turn the  at row r' column $c+1$ into a $\boxplus$. Then we may or may not turn the $\boxplus$ at row r column c into . If we do that, the move is called a *ladder move*. Otherwise, the move is called a *K-ladder move*. Locally, the moves look like the following:

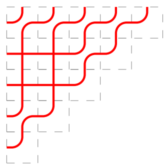


We use "(K-)ladder move" to denote either ladder move or K-ladder move. Clearly, a (K-)ladder move does not change the permutation associated to a pipedream.

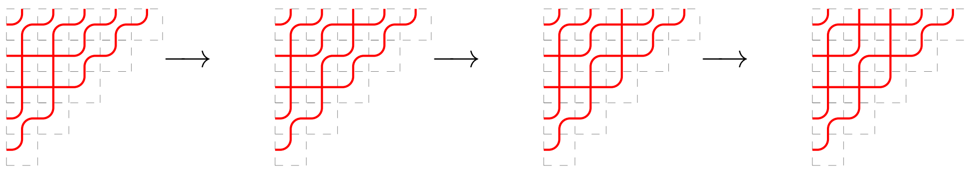
For $w \in S_n$, the statistic $\mathsf{invcode}(w)$ is a sequence of n numbers where the i^{th} number is the number of $j > i$ such that $w(j) < w(i)$. It is well-known that $\mathsf{PD}(w)$ contains the pipedream with row weight $\mathsf{invcode}(w)$ where all ⊞ are left-justified. By [BB93], elements of $\mathsf{PD}(w)$ with lowest number of ⊞ can be obtained from the left-justified pipedream via ladder moves. It is also known that all elements in $\mathsf{PD}(w)$ can be obtained by performing (K-)ladder moves from the left-justified pipedream. Due to the lack of references, we prove this statement for completeness in §2 (see Lemma 8).

We now describe our algorithm that constructs the maximal pipedream of w. We start from the left-justified $\mathsf{Rothe}(w)$ and perform an iterative algorithm. Each iteration places a bar right above row i for $i = n-2, n-3, \ldots, 1$. During each iteration, we only look under the bar and imagine row i is the topmost row. Scan through the columns from right to left. Within each column, scan through the ⊞ from top to bottom. Whenever we see a ⊞ at which we can perform a ladder move, we perform a ladder move. After going through a column, if we have performed ladder moves on this column, we turn the last ladder move into a K-ladder move. We denote the final pipedream by $\widehat{P}(w)$.

Example 3. Take $w \in S_5$ with one-line notation 14523. We start from the following pipedream:



When $i = 3$ and 2, we do not make any moves. When $i = 1$, we perform:



Dreyer, Mészáros, and St. Dizier [DMSD22] found the leading monomial in each homogeneous component of $\mathfrak{G}_w$. Let $\mathsf{reg}(w)$ be the difference between the sum of entries in $\mathsf{rajcode}(w)$ and the sum of entries in $\mathsf{invcode}(w)$. Define the map $\mathsf{IR}(\cdot)$ that sends w to a sequence of monomials $m_0, m_1, \ldots, m_{\mathsf{reg}(w)}$. First, $m_0 := x^{\mathsf{invcode}(w)}$. For $i > 0$, $m_i := m_{i-1}x_p$ where p is the largest such that $m_{i-1}x_p$ divides $x^{\mathsf{rajcode}(w)}$. For each m_i, Dreyer, Mészáros, and St. Dizier [DMSD22] explicitly constructed a climbing chain, another combinatorial model of $\mathfrak{G}_w$ introduced in [LRS06], showing m_i is the leading monomial in its degree of $\mathfrak{G}_w$. In our algorithm, we start from a pipedream with row weight $\mathsf{invcode}(w)$. During the algorithm, we obtain the pipedreams corresponding to $m_1, \cdots, m_{\mathsf{reg}(w)}$.

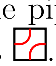
Theorem 4. *Let $w \in S_n$. Perform our algorithm to compute $\widehat{P}(w)$. The algorithm makes $\mathsf{reg}(w)$ K-ladder moves. Right after the i^{th} K-ladder move, we record the row weight of the pipedream as $a_i(w)$. Then $x^{a_i(w)} = m_i$ where $\mathsf{IR}(w) = (m_0, m_1, \ldots, m_{\mathsf{reg}(w)})$.*

The rest of the paper is structured as follows. In §2, we cover necessary background regarding pipedreams and $\mathsf{rajcode}(w)$. In §3, we introduce recursive formulas to compute $\mathsf{rajcode}(w)$, $\mathsf{rajcode}(w^{-1})$ and $\mathsf{IR}(w)$. In §4, we prove our main results using Proposition 23 and Corollary 25, whose proofs are in §5.

2 Background

For integer n, we let $[n] := \{1, \ldots, n\}$.

2.1 Pipedreams and Grothendieck polynomials

Definition 5. *Pipedreams* of size n are tilings with $n+1-i$ left justified tiles in row i. The rightmost tile in each row is  and all other tiles can be  or . For a pipedream of size n, it is associated with a permutation $w \in S_n$. We label the pipes $1, 2, \ldots, n$ along the top edge and follow the pipes. Whenever two pipes cross more than once, we treat all but the first crossing as . Let $\mathsf{PD}(w)$ be the set of the pipedreams associated with $w \in S_n$.

Example 6. Pipedreams in Example 3 are all in $\mathsf{PD}(w)$ where w has one-line notation 14523.

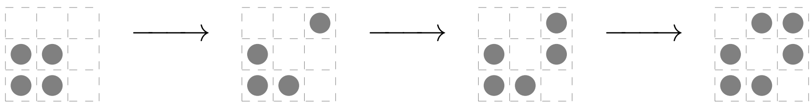
Let P be a pipedream. We write $(i, j) \in P$ if row i column j of P is . Following [KM05] and [FK94], *double Grothendieck polynomial* $\mathfrak{G}_w(\mathbf{x}, \mathbf{y})$ and *Grothendieck polynomial* $\mathfrak{G}_w(\mathbf{x})$ can be defined as

$$\mathfrak{G}_w(\mathbf{x}, \mathbf{y}) := \sum_{P \in \mathsf{PD}(w)} \prod_{(i,j) \in P} (x_i + y_j - x_i y_j), \qquad \mathfrak{G}_w(\mathbf{x}) := \sum_{P \in \mathsf{PD}(w)} \prod_{(i,j) \in P} x_i.$$

In the rest of the paper, we identify a pipedream with a *diagram*, which is a finite subset of $\mathbb{Z}_{>0} \times \mathbb{Z}_{>0}$. We represent a diagram D by drawing a cell in row i column j for each $(i, j) \in D$. We use the matrix coordinates: Row 1 is the topmost row and column 1 is the leftmost column. A *weak composition* is an infinite sequence of $\mathbb{Z}_{\geqslant 0}$ with finitely many positive entries. If α is a weak composition, we use α_i to denote its i^{th} entry. We write α as $(\alpha_1, \ldots, \alpha_n)$ where α_n is the last positive entry in α. The row weight (resp. column weight) of a diagram D is a weak composition where the i^{th} entry is the number of cells in row i (resp. column i) of D. We denote the row weight of a diagram D by $\mathsf{wt}(D)$.

Pipedreams of size n are in bijection with diagrams contained in $\{(i,j) : 1 \leqslant i \leqslant n-1, 1 \leqslant j \leqslant n-i\}$. Under this identification, $\mathsf{PD}(w)$ is a set of diagrams. The ladder move is a move on diagrams and our algorithm is applying ladder moves to diagrams.

Example 7. We repeat Example 3 under our new convention:



The last diagram is $\widehat{P}(w)$ when w has one-line notation 14523. Its row weight and column weight are both $(2,2,2)$.

We prove the following well-known result for completeness. Our proof is adapted from an argument of Weigandt in a private communication.

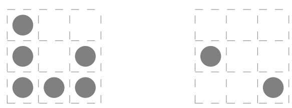
Lemma 8. *Any pipedream in* $\mathsf{PD}(w)$ *can be obtained from the left-justified pipedream via a series of (K-)ladder moves.*

Proof. We give a partial order on $\mathsf{PD}(w)$ by (K-)ladder move. In other words, P is weakly less than P' if P can be obtained from P' via a series of (K-)ladder move. This is clearly a partial order on $\mathsf{PD}(w)$. Our lemma reduces to showing that the maximal element in $\mathsf{PD}(w)$ is left-justified. Take $P \in \mathsf{PD}(w)$ not left-justified. It is enough to find $P' \in \mathsf{PD}(w)$ that is weakly larger than P. We may find $(r,c) \in P$ such that $(r, c-1) \notin P$ and r is chosen to be maximal. Let $\hat{r} > r$ be the smallest such that $(\hat{r}, c) \notin P$. Since r is chosen to be maximal, for any $r < r' < \hat{r}$, we have $(r', c-1), (r', c) \in P$. If $(\hat{r}, c-1) \in P$, we let $P' = P \setminus \{(r,c)\}$. Then P is obtained from P' by a K-ladder move. Otherwise, we let $P' = P \setminus \{(r,c)\} \sqcup \{(\hat{r}, c-1)\}$. Then P is obtained from P' by a ladder move. □

2.2 Snow diagrams and rajcode

For any diagrams D, Pan and Yu defined $\mathsf{dark}(D) \subseteq D$ which can be computed as follows: Scan through D from bottom to top. For each row r, if there exists $(r,c) \in D$ such that currently there is no cells in column c of $\mathsf{dark}(D)$, we find the largest such c and put (r,c) in $\mathsf{dark}(D)$. Cells in $\mathsf{dark}(D)$ of D are called *dark clouds* of D.

Example 9. The following is a diagram D and $\mathsf{dark}(D)$



There is an alternative characterization of $\mathsf{dark}(D)$.

Proposition 10. *The diagram* $\mathsf{dark}(D)$ *is the unique subset of* D *such that*

- *There is at most one cell in each row or column of* D.

- *For any* $(i,j) \in D$*, there is* $(i',j) \in \mathsf{dark}(D)$ *with* $i' > i$ *or there is* $(i,j') \in \mathsf{dark}(D)$ *with* $j' > j$*.*

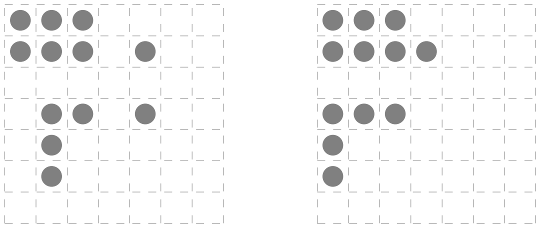
Proof. By Remark 3.4 of [PY23], $\mathsf{dark}(D)$ satisfies the two conditions. The uniqueness is trivial. □

The *Rothe diagram* of w, denoted as $\mathsf{Rothe}(w)$, is the following diagram:

$$\{(i, w(j)) : i < j, w(i) > w(j)\}.$$

For $w \in S_n$, the first n numbers in $\mathsf{wt}(\mathsf{Rothe}(w))$ form $\mathsf{invcode}(w)$. Let $\overleftarrow{\mathsf{Rothe}(w)}$ be the diagram obtained by left-justifying all cells in $\mathsf{Rothe}(w)$. This is the diagram in $\mathsf{PD}(w)$ that our algorithm starts with.

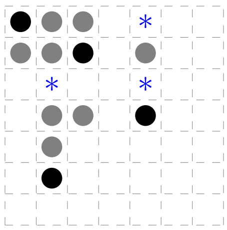
Example 11. Take $w \in S_7$ with one-line notation 4617352. The following are $\mathsf{Rothe}(w)$ and $\overleftarrow{\mathsf{Rothe}(w)}$.



For each $w \in S_n$, Pechenik, Speyer and Weigandt defined the weak composition $\mathsf{rajcode}(w)$ using increasing subsequences of w. In this paper, we use a diagrammatic definition of Pan and Yu [PY23]

Definition 12 ([PY23])**.** Take $w \in S_n$. For each cell in $\mathsf{dark}(\mathsf{Rothe}(w))$, we fill all the empty cells above it in $\mathsf{Rothe}(w)$. The resulting diagram is the *snow diagram* of w. Define $\mathsf{rajcode}(w)$ as the row weight of the snow diagram of w.

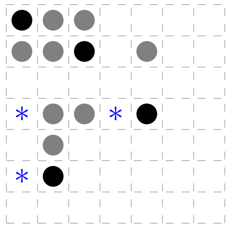
Example 13. Take $w \in S_7$ with one-line notation 4617352. The following is its snow diagram. For clarity, we represent dark clouds by a black circle and use $*$ to denote the added cells.



Thus, $\mathsf{rajcode}(w) = (4, 4, 2, 3, 1, 1)$.

It is well-known that $\mathsf{Rothe}(w)$ and $\mathsf{Rothe}(w^{-1})$ are conjugations of each other. By Proposition 10, $\mathsf{dark}(\mathsf{Rothe}(w))$ and $\mathsf{dark}(\mathsf{Rothe}(w^{-1}))$ are conjugations of each other. Thus, we define the *left snow diagram* of w as the diagram where we fill empty spots on the left of each dark cloud in $\mathsf{Rothe}(w)$. Its column weight will be the same as the row weight of the snow diagram of w^{-1}, which is $\mathsf{rajcode}(w^{-1})$.

Example 14. Keep the same w as in Example 13. Its left snow diagram is



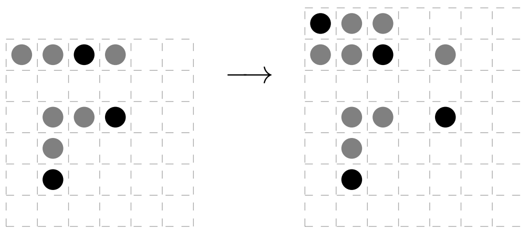
Thus, $\mathsf{rajcode}(w^{-1}) = (4, 5, 3, 1, 2)$.

3 Various recursions

We describe a recursive way to construct $\mathsf{Rothe}(w)$ and $\mathsf{dark}(\mathsf{Rothe}(w))$. Then we obtain recursive formulas for $\mathsf{rajcode}(w)$ and $\mathsf{rajcode}(w^{-1})$. Notice that $\mathsf{invcode}(\cdot)$ is a bijection from S_n to weak compositions $(\alpha_1, \alpha_2, \dots)$ where $\alpha_i \leqslant n-i$ for $i \in [n-1]$ and $\alpha_n = \alpha_{n+1} = \cdots = 0$. We identify $w \in S_n$ with $(a, u) \in \{0, 1, \dots, n-1\} \times S_{n-1}$ where $a = \mathsf{invcode}(w)_1$ and u is the unique permutation in S_{n-1} with $\mathsf{invcode}(u) = (\mathsf{invcode}(w)_2, \mathsf{invcode}(w)_3, \dots)$. We simply write $w = (a, u)$. Then we may recursively construct $\mathsf{Rothe}(w)$ as follows. Start from $\mathsf{Rothe}(u)$. Shift all cells downward by 1. Then shift all cells in columns $a+1, a+2, \dots$ to the right by 1. Finally, put cells at $(1, 1), \cdots, (1, a)$. The resulting diagram is $\mathsf{Rothe}(w)$.

Similarly, to construct $\mathsf{dark}(\mathsf{Rothe}(w))$, we can start from $\mathsf{dark}(\mathsf{Rothe}(u))$. Shift all cells downward by 1. Then shift all cells in columns $a+1, a+2, \dots$ to the right by 1. Finally, find the largest $c \in [a]$ such that $\mathsf{dark}(\mathsf{Rothe}(u))$ has no cells in column c. Put $(1, c)$ into $\mathsf{dark}(\mathsf{Rothe}(u))$.

Example 15. Keep $w \in S_7$ with one-line notation 4617352. We have $w = (a, u)$ where $a = 3$ and $u \in S_6$ has one-line notation 516342. We depict how $\mathsf{Rothe}(u)$ and $\mathsf{Rothe}(w)$ as follows. The dark cells form $\mathsf{dark}(\mathsf{Rothe}(u))$ and $\mathsf{dark}(\mathsf{Rothe}(w))$ respectively.



Consequently, we may compute $\mathsf{rajcode}(w)$ and $\mathsf{rajcode}(w^{-1})$ recursively. Let $d_c(u)$ be the number of cells in $\mathsf{dark}(\mathsf{Rothe}(u))$ that are strictly to the right of column c.

Proposition 16. *Take $w = (a, u) \in S_n$.*

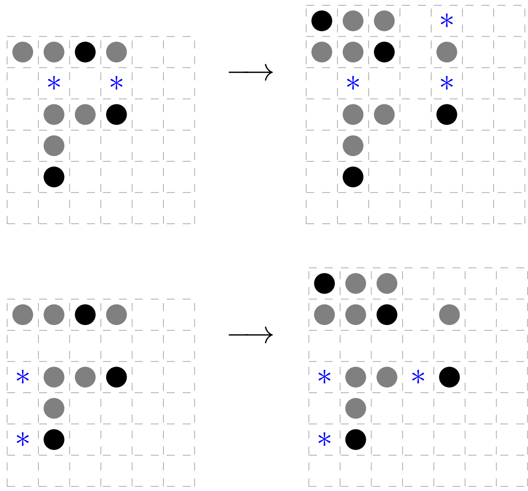
- *We can get* $\mathsf{rajcode}(w)$ *by prepending* $a + d_a(u)$ *to* $\mathsf{rajcode}(u)$.

- *To obtain* $\mathsf{rajcode}(w^{-1})$*, we just insert* $d_a(u)$ *between the* a^{th} *and* $(a+1)^{th}$ *entries of* $\mathsf{rajcode}(u^{-1})$*. Then increase the first* a *entries by* 1*.*

Consequently, $\mathsf{reg}(w) - \mathsf{reg}(u) = d_a(u)$*.*

Proof. Follows directly from the recursive constructions of $\mathsf{Rothe}(w)$ and $\mathsf{dark}(\mathsf{Rothe}(w))$. □

Example 17. Keep $w = (a, u)$ in Example 15. We show how the snow diagram and left snow diagram of w differ from those of u:



We have $d_a(u) = 1$. We obtain $\mathsf{rajcode}(w) = (4, 4, 2, 3, 1, 1)$ by prepending $a + d_a(u) = 4$ to $\mathsf{rajcode}(u) = (4, 2, 3, 1, 1)$. We obtain $\mathsf{rajcode}(w^{-1}) = (4, 5, 3, 1, 2)$ by inserting $d_a(u)$ after the a^{th} entry of $\mathsf{rajcode}(u^{-1}) = (3, 4, 2, 2)$ and then increase the first a entries by 1.

Notice that when $w = (a, u)$, $\mathsf{invcode}(w)$ can be obtained by prepending the number a to $\mathsf{invcode}(u)$. Thus, we also have a recursive formula for $\mathsf{IR}(w)$. For a monomial m, let $\overrightarrow{m}$ be the monomial obtained by turning each x_i in m into x_{i+1}.

Proposition 18. *Take* $w = (a, u) \in S_n$*. Let*

$$(M_0, \ldots, M_{\mathsf{reg}(w)}) = \mathsf{IR}(w), (m_0, \ldots, m_{\mathsf{reg}(u)}) = \mathsf{IR}(u).$$

Then $\mathsf{reg}(w) = \mathsf{reg}(u) + d_a(u)$ *and*

$$M_j = \begin{cases} x_1^a \overrightarrow{m_j} & \text{if } j = 0, 1, \ldots, \mathsf{reg}(u), \\ x_1^{a+j-\mathsf{reg}(u)} \times \overrightarrow{m_{\mathsf{reg}(u)}} & \text{if } j = \mathsf{reg}(u) + 1, \ldots, \mathsf{reg}(w). \end{cases}$$

Proof. Follows directly from the recursive formula of $\mathsf{rajcode}(w)$ and the definition of $\mathsf{IR}(\cdot)$. □

Example 19. Keep $w = (a, u)$ in Example 15. We have $\mathsf{reg}(u) = 2$ and $\mathsf{reg}(w) = \mathsf{reg}(u) + d_a(u) = 3$. Since

$$\mathsf{IR}(u) = (x^{(4,0,3,1,1)}, x^{(4,1,3,1,1)}, x^{(4,2,3,1,1)}),$$

we have

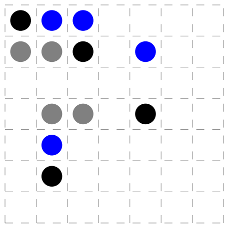
$$\mathsf{IR}(w) = (x^{(3,4,0,3,1,1)}, x^{(3,4,1,3,1,1)}, x^{(3,4,2,3,1,1)}, x^{(4,4,2,3,1,1)}).$$

4 Proof of main theorems

To prove our main theorems, we need to introduce a new permutation statistic.

Definition 20. For $w \in S_n$, its *movecode*, denoted as $\mathsf{movecode}(w)$, is a weak composition where $\mathsf{movecode}(w)_i$ is the number of cells in column i of $\mathsf{Rothe}(w)$ with no dark clouds strictly to its right.

Example 21. Take $w \in S_7$ with one-line notation 4617352. The following is $\mathsf{Rothe}(w)$, where the black cells are dark clouds and blue cells are non-dark cloud cells without dark clouds to their right.



Then $\mathsf{movecode}(w)$ is the number of black and blue cells in each column. Thus, we know it is $(1, 3, 2, 0, 2)$.

We have the following observation regarding this permutation statistic.

Proposition 22. *Take $w \in S_n$ and $c \in [n]$. Then*

$$\mathsf{rajcode}(w^{-1})_{c+1} - \max(\mathsf{movecode}(w)_{c+1} - 1, 0) = d_c(u) = \mathsf{rajcode}(w^{-1})_c - \mathsf{movecode}(w)_c.$$

Proof. We refer to cells in $\mathsf{dark}(\mathsf{Rothe}(w))$ as dark clouds. Consider the left snow diagram of w. In the diagram, there are four types of cells.

- Type 1: Dark clouds.
- Type 2: Cells that do not belong to $\mathsf{Rothe}(w)$.
- Type 3: Cells in $\mathsf{Rothe}(w)$ with a dark cloud in its row on its right.
- Type 4: Cells in $\mathsf{Rothe}(w)$ that is not a dark cloud and has no dark cloud in its row on its right.

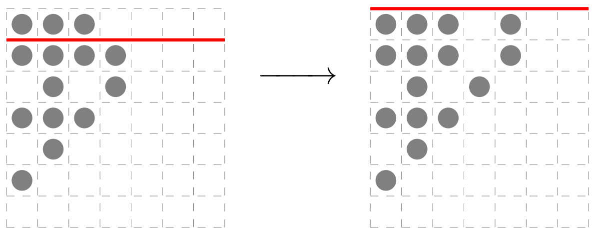
The number of type 1, 2 and 4 cells in column $c + 1$ is $d_c(w)$. The number of all cells in column $c + 1$ is $\mathsf{rajcode}(w^{-1})_{c+1}$. The number of type 3 cells in column $c + 1$ is $\max(\mathsf{movecode}(w)_c - 1, 0)$, so we have the first equation.

The number of type 2 and 3 cells in column c is $d_c(w)$. The number of all cells in column c is $\mathsf{rajcode}(w^{-1})_c$. The number of type 1 and 4 cells in column c is $\mathsf{movecode}(w)_c$, so we have the second equation. □

The main application of $\mathsf{movecode}(w)$ is to characterize the number of cells moved when our algorithm processes each column.

Proposition 23. *Take $v = (a, w) \in S_n$. During the last iteration of the algorithm that computes $\widehat{P}(v)$, the number of cells moved in column c is $\mathsf{movecode}(w)_c$ if $c > a$ and 0 otherwise.*

Example 24. Keep $v \in S_7$ with one-line notation 4617352. We have $v = (a, w)$ where $a = 3$ and $w \in S_6$ has one-line notation 516342. We have $\mathsf{movecode}(w) = (0, 2, 1, 2)$. During the last iteration of the algorithm, the bar is right above row 1. The algorithm moves 0 cells in column $c > 4$, since $\mathsf{movecode}(w)_c = 0$. The algorithm moves 2 cells in column 4 since $4 > a$ and $\mathsf{movecode}(w)_4 = 2$. It moves 0 cells in column $3, 2$, and 1 since $1, 2, 3 \leqslant a$.



We prove this proposition in §5. Our proof requires a few technical lemmas which also lead to the following result:

Corollary 25. *Consider the iteration when the bar is right above row i in our algorithm. Let D_1 (resp. D_2) be the diagram before (resp. after) processing one column. If the algorithm makes a move in this column, then $\mathsf{wt}(D_2)$ is obtained from increasing i^{th} entry of $\mathsf{wt}(D_1)$ by 1.*

Using Proposition 23 and Corollary 25, we can prove our main results. We start with Theorem 4.

Proof of Theorem 4. We induct on n. The base case ($n = 1$) is trivial. Let $w = (a, u) \in S_n$ with $n > 1$. By our inductive hypothesis, the algorithm made $\mathsf{reg}(u)$ K-ladder moves before the last iteration. By Proposition 23, in the last iteration of the algorithm, it makes a K-ladder move in column c if and only if $c > a$ and $\mathsf{movecode}(u)_c > 0$. This is exactly the number $d_a(u)$, which equals $\mathsf{reg}(w) - \mathsf{reg}(u)$ by Proposition 16. Thus, the algorithm to compute $\widehat{P}(w)$ makes $\mathsf{reg}(w)$ K-ladder moves in total.

Let

$$\mathsf{IR}(w) = (M_0, \ldots, M_{\mathsf{reg}(w)}), \mathsf{IR}(u) = (m_0, \ldots, m_{\mathsf{reg}(u)}).$$

By Proposition 18, for $i = 0, \ldots, \mathsf{reg}(u)$, we have $M_i = x_1^a \overrightarrow{m_i}$. When the algorithm makes the i^{th} K-ladder move, the bar has not reached row 1. Before the bar reaches row 1, the algorithm ignores the first row of the diagram, which has a cells, and behaves as if computing $\widehat{P}(u)$. Thus, the statement holds for $i = 0, 1, \ldots, \mathsf{reg}(u)$ by our inductive hypothesis.

For $i = \mathsf{reg}(u) + 1, \ldots, \mathsf{reg}(w)$, the i^{th} K-ladder move happens when the bar is above row 1. Let D be the diagram right after the $(i-1)^{\text{th}}$ K-ladder move and D' be the diagram right after the i^{th} K-ladder move. By Corollary 25, $x^{\mathsf{wt}(D')} = x_1 \cdot x^{\mathsf{wt}(D)}$, which concludes the proof. □

Proof of Theorem 2. By Theorem 4, the row weight of $\widehat{P}(w)$ is $\mathsf{rajcode}(w)$. For the column weight, we prove by induction on n. The base case $n = 1$ is trivial. Now assume $n > 1$ and $w = (a, u) \in S_n$. Let D be the diagram we have right before the last iteration of the algorithm computing $\widehat{P}(w)$. It can be obtained by shifting $\widehat{P}(u)$ downward by 1 and append a left-justified cells in the first row. By our inductive hypothesis, $\widehat{P}(u)$ has column weight $\mathsf{rajcode}(u^{-1})$. Now take $c \in [n-1]$ and consider three cases:

- Suppose $c > a + 1$. Consider the last iteration of the algorithm. By Proposition 23, the algorithm makes $\mathsf{movecode}(u)_c$ (resp. $\mathsf{movecode}(u)_{c-1}$) moves in column c (resp. $c - 1$). Thus, column c loses $\max(\mathsf{movecode}(u)_c - 1, 0)$ cells and then gain $\mathsf{movecode}(u)_{c-1}$ cells. By Proposition 22, $\widehat{P}(w)$ has

$$\mathsf{rajcode}_c(u^{-1}) - \max(\mathsf{movecode}(u)_c - 1, 0) + \mathsf{movecode}(u)_{c-1} = \mathsf{rajcode}_{c-1}(u^{-1})$$

  cells in column c. Finally, by Proposition 16, $\mathsf{rajcode}_{c-1}(u^{-1})$ is just $\mathsf{rajcode}_c(w^{-1})$.

- Suppose $c = a + 1$. By Proposition 23, the algorithm makes $\mathsf{movecode}(u)_c$ moves in column c, and makes 0 moves in column $c - 1$ if it exists. Thus, column c loses $\max(\mathsf{movecode}(u)_c - 1, 0)$ cells. By Proposition 22, $\widehat{P}(w)$ has

$$\mathsf{rajcode}_c(u^{-1}) - \max(\mathsf{movecode}(u)_c - 1, 0) = d_a(u)$$

  cells in column c. Finally, by Proposition 16, $d_a(u)$ is just $\mathsf{rajcode}_c(w^{-1})$.

- Suppose $c \in [a]$. By Proposition 23, the algorithm makes 0 moves in column c, and makes 0 moves in column $c - 1$ if it exists. Thus, $\widehat{P}(w)$ has $\mathsf{rajcode}(u^{-1})_c + 1$ cells in column c. Finally, by Proposition 16, $\mathsf{rajcode}(u^{-1})_c + 1$ is just $\mathsf{rajcode}_c(w^{-1})$. □

5 Proof of Proposition 23 and Corollary 25

Following §3, we derive a recursive way to compute $\mathsf{movecode}(w)$.

Lemma 26. *For $w \in S_n$, we write $w = (a, u)$. Then* $\mathsf{movecode}(w)$ *can be determined starting from* $\mathsf{movecode}(u)$. *First, insert a* 0 *between* $\mathsf{movecode}(u)_a$ *and* $\mathsf{movecode}(u)_{a+1}$. *Then start from the a^{th} entry and increase each entry by* 1 *from right to left. Whenever we change a* 0 *into a* 1, *we stop immediately. The resulting weak composition is* $\mathsf{movecode}(w)$.

Proof. Follows directly from the recursive constructions of $\mathsf{Rothe}(w)$ and $\mathsf{dark}(\mathsf{Rothe}(w))$. □

Example 27. Take $w \in S_7$ with one-line notation 4617352. We have $w = (3, u)$ where $u \in S_6$ has one-line notation 516342. We have $\mathsf{movecode}(u) = (0, 2, 1, 2)$. Then we insert a 0 between $\mathsf{movecode}(u)_3$ and $\mathsf{movecode}(u)_4$, obtaining $(0, 2, 1, 0, 2)$. We then increase entries by 1 from right to left, starting from the thrid entry. When we turn the 0 in the first entry into 1, we stop, obtaining $(1, 3, 2, 0, 2)$.

Our proofs rely on a simple operator on diagrams. We may break the algorithm into a sequence of this operator.

Definition 28. We define the operator $L_{i,c}$ on diagrams. Take diagram D and put a bar above row i in D. We ignore everything above the bar, imagining row i is the top-most row. Then we scan through cells in column c from top to bottom. Whenever we see a cell at which we can perform a ladder move, we perform a ladder move. After going through this column, if we made a move, turn the last move into a K-ladder move.

With this notion, applying the algorithm on $w \in S_n$ can be rewritten as

$$\widehat{P}(w) = (L_{1,1} \cdots L_{1,n-2}) \cdots (L_{n-3,1} L_{n-3,2})(L_{n-2,1})(\overleftarrow{\mathsf{Rothe}(w)}). \tag{1}$$

In words, we iterate through $i = n-2, \ldots, 2, 1$. For each i, we iterate through $c = n-1-i, \ldots, 2, 1$ and apply $L_{i,c}$.

We start by observing a straightforward recursive property of this operator.

Remark 29. Fix $i, c \in \mathbb{Z}_{>0}$ and let D be a diagram. Suppose $(i, c) \notin D$ and $(i, c+1) \notin D$.

- Suppose $(i+1, c) \in D$ and $(i+1, c+1) \notin D$. Let D' be the diagram obtained by moving $(i+1, c)$ to $(i, c+1)$ in D. If $L_{i+1,c}(D') \neq D'$, we know $L_{i,c}(D) = L_{i+1,c}(D')$. Otherwise, $L_{i,c}(D) = D' \sqcup \{(i+1, c)\}$. Informally, in this case, $L_{i,c}$ behaves as if $L_{i+1,c}$ after the ladder move on $(i+1, c)$.
- Suppose $(i+1, c) \in D$ and $(i+1, c+1) \in D$. Then intuitively, $L_{i,c}$ behaves as if row $i+1$ is ignored: Let D' be obtained from D by removing $(i+1, c)$ and $(i+1, c+1)$. If $(i+1, c+1) \notin L_{i+1,c}(D')$, $L_{i,c}(D) = L_{i+1,c}(D') \sqcup \{(i+1, c), (i+1, c+1)\}$. Otherwise, $L_{i,c}(D) = L_{i+1,c}(D') \sqcup \{(i+1, c), (i, c+1)\}$.

We are primarily interested in applying $L_{i,c}$ to a diagram in the following case.

Definition 30. We say the operator $L_{i,c}$ *acts initially on* D if D is fixed by $L_{i+1,c}$.

Eventually, we will show all $L_{i,c}$ in our algorithm acts initially. We first derive a few properties when $L_{i,c}$ acts initially on D.

Lemma 31. *Suppose $L_{i,c}$ acts initially on D and $L_{i,c}$ moves at least one cell. We let $(r_1, c), \ldots, (r_k, c)$ be the cells moved where $r_1 < \cdots < r_k$. Let $r_0 = i$. Then we know the cell (r_j, c) is moved to $(r_{j-1}, c+1)$ for $j \in [k]$. Thus, $\mathsf{wt}(L_{i,c}(D))$ is obtained from $\mathsf{wt}(D)$ by adding 1 to the i^{th} entry.*

Proof. If $L_{i,c}$ moves (r_1, c) to $(r', c+1)$ for some $r' > i$, then $L_{i+1,c}$ will also move (r_1, c) to $(r', c+1)$. This contradicts our assumption that $L_{i,c}$ acts initially on D. Thus, $L_{i,c}$ moves (r_1, c) to $(i, c+1)$.

For $j > 1$, when (r_j, c) moves, (r_{j-1}, c) and $(r_{j-1}, c+1)$ must both be empty since the cell in (r_{j-1}, c) just performed a ladder move. Therefore (r_j, c) must be moved to $(r', c+1)$ for some $r' \geqslant r_{j-1}$. However, $r' > r_{j-1}$ contradicts the assumption that $L_{i,c}$ acts initially on D, so $r' = r_{j-1}$. □

To better describe the effect of $L_{i,c}$ when it acts initially, we introduce the following notion.

Definition 32. The *(i,c)-initial segment* of a diagram D is the set of (r,c) such that $(r',c) \in D$ for all $i \leqslant r' \leqslant r$.

This notion characterizes the destination of cells moved by $L_{i,c}$ when it acts initially.

Lemma 33. *Suppose $L_{i,c}$ acts initially on D. Then it moves cells to the $(i,c+1)$-initial segment of $L_{i,c}(D)$.*

Proof. Let $(r_1,c),(r_2,c),\dots,(r_k,c)$ where $r_1 < r_2 < \cdots < r_k$ be the cells of D moved by $L_{i,c}$. Let $r_0 = i$. By Lemma 31, for $j \in [k]$, (r_j,c) is moved to $(r_{j-1},c+1)$. We show (r_{j-1},c) is in the $(j,c+1)$-initial segment of $L_{i,c}(D)$ by induction on j. For the base case, $(r_0,c+1) = (i,c+1)$ is clearly in the $(j,c+1)$-initial segment of $L_{i,c}(D)$

For $j > 1$. assume $(r_{j-2},c+1)$ is in the $(i,c+1)$-initial segment of $L_{i,c}(D)$. Since (r_{j-1},c) is moved to $(r_{j-2},c+1)$, we know $(r',c+1) \in L_{i,c}(D)$ for any $r_{j-2} < r' < r_{j-1}$. Thus, $(r_{j-1},c+1)$ is in the $(i,c+1)$-initial segment of $L_{i,c}(D)$. □

We can also use "initial segment" to characterize what cells can be moved by $L_{i,c}$ when it acts initially.

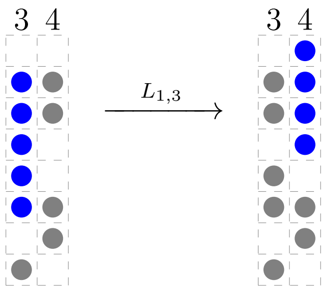
Lemma 34. *Suppose $L_{i,c}$ acts initially on D. If $(i,c) \in D$, then D is fixed by $L_{i,c}$. Otherwise, a cell $(r,c) \in D$ is moved by $L_{i,c}$ if and only if it is in the $(i+1,c)$-initial segment of D and $(r,c+1) \notin D$.*

Proof. The lemma is immediate when $(i,c) \in D$. Otherwise, let $(r_1,c),\dots,(r_k,c) \in D$ be the cells moved by $L_{i,c}$ where $r_1 < \cdots < r_k$. Let $r_0 = i$. Clearly, $(r_j,c+1) \notin D$ for each $j \in [k]$. We prove (r_j,c) is in the $(i+1,c)$-initial segment of D by induction. First, by Lemma 31, (r_1,c) is moved to $(r_0,c+1)$, so $(r',c) \in D$ for $r_0 = i < r' < r_1$. In other words, (r_1,c) is in the $(i+1,c)$-initial segment of D. For $j > 1$, by Lemma 31, (r_j,c) is moved to $(r_{j-1},c+1)$, so $(r',c) \in D$ for $r_{j-1} < r' < r_j$. The inductive step is finished since (r_{j-1},c) is in the $(i+1,c)$-initial segment of D.

Now assume (r,c) is a cell in the $(i+1,c)$-initial segment of D and $(r,c+1) \notin D$. Assume toward contradiction that (r,c) is not moved by $L_{i,c}$. Take the smallest such r. Since $L_{i,c}$ moves (r_j,c) to (r_{j-1},c), we know $(r',c+1) \in D$ for any $r_{j-1} < r' < r_j$. Thus, we cannot have $r_{j-1} < r < r_j$ for $j \in [k]$. Since (r,c) is not moved, we know r is not $r_1,\dots,r_k$. Thus, $r > r_k$. By the minimality of r, $(r',c),(r',c+1) \in D$ for $r_k < r' < r$. Thus, $L_{i,c}$ moves (r_k,c), it can perform a ladder move at (r,c). Contradiction. □

The following example is a demonstration of the previous two lemmas related to initial segments.

Example 35. Let D be a diagram whose column 3 and 4 look like the picture on the left. Notice that D will be fixed by $L_{2,3}$. After applying $L_{1,3}$, these two columns look like the picture on the right:



We color the $(2,3)$-initial segment of D and $(1,4)$-initial segment of $L_{1,3}(D)$. Notice that $L_{1,3}$ moves cells to the $(1,4)$-initial segment of $L_{1,3}(D)$. Also notice that a cell in column 3 is moved if and only if it is in the $(2,3)$-initial segment of D and has no cell on its right.

We also have the "converse statement" of Lemma 34.

Lemma 36. *Suppose $(i,c) \notin D$. If $L_{i,c}$ only moves cells in the $(i+1,c)$-initial segment of D, then it acts initially on D.*

Proof. Suppose to the contrary that D is not fixed by $L_{i+1,c}$. Let (r,c) be the first cell moved by $L_{i+1,c}$. Clearly, (r,c) is not in the $(i+1,c)$-initial segment of D and it will also be moved by $L_{i,c}$. □

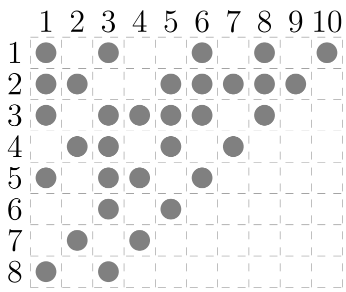
We introduce more definitions that capture the structure of columns for intermediate diagrams during our algorithm.

Definition 37. We say a diagram D is *(i,c)-paired* if the following are satisfied:

- Take any cell $(R,c) \in D$ with $i \leqslant R$ and $(R,c+1) \notin D$. There exists $(r,c+1) \in D$ with $i \leqslant r < R$ and $(r,c) \notin D$. Moreover, $(r',c),(r',c+1) \in D$ for any $r < r' < R$.
- Take any cell $(r,c+1) \in D$ with $i \leqslant r$ and $(r,c) \notin D$. There exists $(R,c) \in D$ with $r < R$ and $(R,c+1) \notin D$. Moreover, $(r',c),(r',c+1) \in D$ for any $r < r' < R$.

Remark 38. Notice that if D is (i,c)-paired, then $L_{i,c}$ fixes D.

Example 39. Consider the following diagram D.

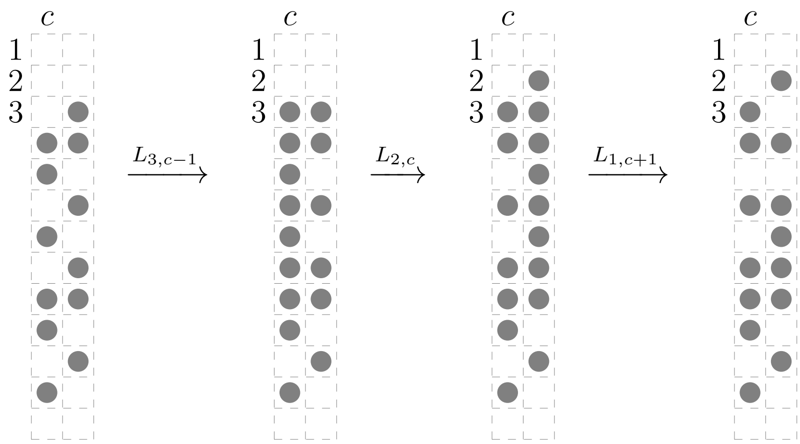


Then D has the following properties: $(1,5)$-paired, $(1,9)$-paired, $(4,1)$-paired, $(6,1)$-paired.

We have the following lemma regarding this new notion.

Lemma 40. *Let diagram D be $(3,c)$-paired and $(2,c+1) \notin D$. We consider the actions of $L_{1,c+1}L_{2,c}L_{3,c-1}$ on D. Assume $L_{3,c-1}$ and $L_{2,c}$ act initially. Let $(r_1,c),\ldots,(r_m,c)$ be the cells moved by $L_{2,c}$ with $r_1 < \cdots < r_m$ and let $r_0 = 2$. We further assume $L_{1,c+1}$ moves $(r'_1,c+1),\ldots,(r'_m,c+1)$ with $r_{i-1} \leqslant r'_i < r_i$. Then $D' = L_{1,c+1}L_{2,c}L_{3,c-1}(D)$ is $(2,c)$-paired.*

Example 41. Consider the action of $L_{1,c+1}L_{2,c}L_{3,c-1}$ on D whose column c and $c+1$ are depicted in the left-most figure. We see D is $(3,c)$-paired. The action of $L_{2,c}$ and $L_{1,c+1}$ satisfy the condition in Lemma 40: For instance, $L_{2,c}$ moves $(5,c)$ to $(2,c+1)$ and there is a unique cell $(r,c+1)$ moved by $L_{1,c+1}$ with $2 \leqslant r < 5$, namely $(3,c+1)$. Then by the Lemma, we know $L_{1,c+1}L_{2,c}L_{3,c-1}(D)$, whose column c and $c+1$ are depicted in the right-most figure, is $(2,c)$-paired.



Proof. Say (t,c) is the bottom-most cell in the $(2,c)$-initial segment of $L_{3,c-1}(D)$. Since $L_{3,c-1}$ acts initially on D, it will only move cells to the $(2,c)$-initial segment by Lemma 33. Since $L_{2,c}$ acts initially on D, it will only move cells in the $(2,c)$-initial segment by Lemma 34. Then by our assumption in the lemma, $L_{1,c+1}$ also moves cells above row t. Thus, D and D' agreed under row t in column c and $c+1$. Now we check D' is $(2,c)$-paired.

Take (R,c) in D' such that $R \geqslant 2$ and $(R,c+1) \notin D'$. We find the r satisfying the condition in the definition of $(2,c)$-paired by considering two cases.

- If $R > t$, then $(R,c) \in D$ and $(R,c+1) \notin D$. Since D is $(3,c)$-paired, we can find $(r,c+1) \in D$ such that $2 \leqslant r < R$, $(r,c) \notin D$ and $(r',c),(r',c+1) \in D$ for $r < r' < R$. It remains to show $r > t$. If not, (r,c) is in the $(2,c)$-initial segment of $L_{3,c-1}(D)$, then so is (R,c), contradicting $R > t$.

- If $R \leqslant t$, then $(R,c) \in L_{3,c-1}(D)$. If $(R,c+1) \notin L_{3,c-1}(D)$, by Lemma 34, $L_{2,c}$ moves (R,c). Since (R,c) is in D', we know it is the last cell moved by $L_{2,c}$, so $R = r_m$. By Lemma 31, $L_{2,c}$ moves (r_m,c) to $(r_{m-1},c+1)$. We have $(r_{m-1},c) \notin D'$.

By our assumption on $L_{1,c-1}$, it does not make a ladder move on cells between row r_{m-1} and row r_m. Thus, we may pick $r = r_{m-1}$.

Now assume $(R, c+1) \in L_{3,c-1}(D)$. Then, $L_{1,c+1}$ moves $(R, c+1)$, so $R = r'_i$ for some $i \in [m-1]$. We know $L_{2,c}$ moves (r_i, c) to $(r_{i-1}, c+1)$. By our assumption on $L_{1,c+1}$, $r_{i-1} < r'_i$ and $L_{1,c+1}$ does not make a move between row r_{i-1} and r'_i. Thus, we may pick $r = r_{i-1}$.

Take $(r, c+1)$ in D' such that $r \geqslant 2$ and $(r, c) \notin D'$. We find the R satisfying the condition in the definition of $(2, c)$-paired by considering two cases.

- If $r > t$, then $(r, c+1) \in D$ and $(r, c) \notin D$. Moreover, since $(2, c+1) \notin D$, we know $r \geqslant 3$. Since D is $(3, c)$-paired, we can find $R > r > t$ such that $(R, c) \in D$, $(R, c+1) \notin D$ and $(r', c), (r', c+1) \in D$ for $r < r' < R$.
- If $r \leqslant t$, then $(r, c) \in L_{3,c-1}(D)$. We know $L_{2,c}$ performs a ladder move on (r, c), so $r = r_i$ for some $i \in [m-1]$. We know $r_i < r'_{i+1} < r_{i+1}$ and $(r', c), (r', c+1) \in L_{2,c}L_{3,c-1}(D)$ for $r_i < r' < r_{i+1}$. If $i+1 < m$, then $L_{1,c+1}$ makes a ladder move on $(r'_{i+1}, c+1)$. We have $(r'_{i+1}, c) \in D'$ and $(r_{i+1}, c+1) \notin D'$. We may pick $R = r'$. If $i+1 = m$, then $L_{1,c+1}$ makes a K-ladder move on $(r'_{i+1}, c+1)$. We may pick $R = r_{i+1}$. □

The last piece of our preparation work is the following observation.

Remark 42. Notice that $L_{i,c}$ and $L_{i',c'}$ commute if $|c - c'| > 1$. Therefore, we know applying

$$L_{1,1}L_{1,2}\cdots L_{1,n-2} \quad L_{2,1}L_{2,2}\cdots L_{2,n-3}$$

is the same as applying

$$L_{1,1}\ \ L_{1,2}L_{2,1}\ \ L_{1,3}L_{2,2}\ \ \cdots\ \ L_{1,n-4}L_{2,n-3}\ \ L_{1,n-2}L_{2,n-3}.$$

Moreover, each $L_{i,c}$ behaves the same in both expressions.

Now we embark on proving Proposition 23 and Corollary 25. We start by introducing two claims which will imply Proposition 23 and Corollary 25 respectively. For a diagram D, let $D^{\downarrow k}$ be the diagram obtained by shifting all cells of D downward by k. We claim:

- Claim 1: Take $N \in \mathbb{Z}_{>0}$ and $w \in S_N$. Consider

  $$(L_{1,2}L_{2,1})\cdots(L_{1,N-2}L_{2,N-3})(L_{1,N}L_{2,N-1})(\widehat{P}(w)^{\downarrow 2}). \tag{2}$$

  Take any $c \in [N-1]$. Then $L_{2,c}$ and $L_{1,c+1}$ move the same number of cells. More specifically, suppose $L_{2,c}$ moves a cell (r, c) to $(\hat{r}, c+1)$. Then there exists a unique r' such that $\hat{r} \leqslant r' < r$ and $(r', c+1)$ is moved by $L_{1,c+1}$. In addition, after the action of $L_{1,c+1}$, the diagram is $(2, c)$-paired.

- Claim 2: Take $N \in \mathbb{Z}_{>0}$ and $w \in S_N$. Consider

  $$L_{1,1}\cdots L_{1,N-1}(\widehat{P}(w)^{\downarrow 1}).$$

  Each $L_{1,c}$ acts initially.

We will inductively show both claims hold for all N. The induction is based on Lemma 43 and Lemma 44.

Lemma 43. *Suppose Claim 1 and Claim 2 hold for $N \leqslant n$, then Claim 2 holds for $N = n+1$.*

Proof. Suppose $w = (b, u) \in S_{n+1}$. Let D be the diagram obtained by putting b left-justified cells in the second row of $\widehat{P}(u)^{\downarrow 2}$. Then $\widehat{P}(w)^{\downarrow 1} = L_{2,1}L_{2,2}\cdots L_{2,n-1}(D)$ and each $L_{2,c}$ acts initially by Claim 2 for u. By Remark 42, we may write $L_{1,1}\cdots L_{1,N-1}(\widehat{P}(w)^{\downarrow 1})$ as

$$L_{1,1}\cdots L_{1,N-1}\, L_{2,1}\cdots L_{2,n-1}(D) = (L_{1,2}L_{2,1})\cdots(L_{1,N-2}L_{2,N-3})(L_{1,N}L_{2,N-1})(D). \quad (3)$$

Clearly, for $c \leqslant b$, $L_{1,c}$ acts initially on $\widehat{P}(w)^{\downarrow 1}$. Now take $c > b$. We know the $L_{1,c}$ behaves the same in both sides of (3). By Lemma 36, it is enough to show each $L_{1,c}$ on the right hand side moves cells in the $(2, c)$-initial segment. Since $L_{2,c-1}$ acts initially, by Lemma 33, $L_{2,c-1}$ moves cells into the $(2, c)$-initial segment. Then by Claim 1 of u, $L_{1,c}$ moves cells in the $(2, c)$-initial segment. □

Lemma 44. *Suppose Claim 1 holds for $N \leqslant n$ and Claim 2 holds for $N \leqslant n+1$, then Claim 1 holds for $N = n+1$.*

Proof. Since Claim 2 holds for $N \leqslant n+1$, each $L_{1,c}$ and $L_{2,c}$ in (2) acts initially by Remark 42. We prove Claim 1 by induction on $c = n, \ldots, 2, 1$. The base case with $c = n$ is trivial.

Suppose $c \in [n-1]$. Let D' be the diagram right before applying $L_{2,c}$ in (2). By our inductive hypothesis for $c+1$, D' is $(2, c+1)$-paired. Now apply $L_{2,c}$ to D'. Let $(r_1, c), \cdots, (r_k, c)$ be the cells moved by $L_{2,c}$. Let $r_0 = 2$. For $j \in [k]$, by Lemma 31, (r_j, c) is moved to $(r_{j-1}, c+1)$. By Lemma 33, $(r_{j-1}, c+1)$ is in the $(2, c+1)$-initial segment of $L_{2,c}(D)$. We consider two cases.

- If $(r_{j-1}, c+2) \notin D'$, then $(r_{j-1}, c+1)$ will be moved by $L_{1,c+1}$ by Lemma 34. For $r_{j-1} < r' < r$, by D' is $(2, c+1)$-paired, we know $(r', c+1), (r', c+2) \in D'$. By Lemma 34, $L_{1,c+1}$ will not move $(r', c+1)$.
- Now assume $(r_{j-1}, c+2) \in D'$. Since D' is $(2, c+1)$-paired and $(r_{j-1}, c+) \notin D'$, we can find $R > r_{j-1}$ such that $(R, c+1) \in D'$, $(R, c+2) \notin D$ and $(r', c+1), (r', c+2) \in D'$ for any $r_{j-1} < r' < R$. We know $(r_j, c+1) \notin D'$, so $R < r_j$. For $R < r' < r_j$, since $(r', c+1) \in D'$ and D' is $(2, c+1)$-paired, we must have $(r', c+2) \in D'$. By 34, $(R, c+1)$ is the unique cell moved during $L_{1,c+1}$ between row r_{j-1} and row r_j.

Now we show $L_{1,c+1}$ and $L_{2,c}$ move the same number of cells, we already know $L_{1,c+1}$ makes exactly one move between row r_{j-1} and row r_j inclusively for $j \in [k]$. We just need to show $L_{1,c+1}$ does not move any $(r, c+1)$ for any $r > r_k$. Notice that $(r_k, c+1) \notin L_{2,c}(D')$, so $(r, c+1)$ is not in the $(2, c+1)$-initial segment of $L_{2,c}(D')$. By Lemma 34, $(r, c+1)$ will not be moved.

It remains to check $L_{1,c+1}L_{2,c}(D')$ is $(2,c)$-paired. Write w as (b,u). Let D be the diagram obtained by putting b left-justified cells in row 3 of $\widehat{P}(u)^{\downarrow 3}$. Then

$$\widehat{P}(w)^{\downarrow 2} = L_{3,1}L_{3,2}\cdots L_{3,n-1}(D).$$

By Remark 42,

$$\begin{aligned}&(L_{1,2}L_{2,1})\cdots(L_{1,n+1}L_{2,n})(\widehat{P}(w)^{\downarrow 2})\\=&(L_{1,2}L_{2,1})\cdots(L_{1,n+1}L_{2,n})(L_{3,1}L_{3,2}\cdots L_{3,n-1})(D)\\=&(L_{1,2}L_{2,1})(L_{1,3}L_{2,2}L_{3,1})\cdots(L_{1,n+1}L_{2,n}L_{3,n-1})(D).\end{aligned}$$

If $c > b$, then $(3,c) \notin D$. By claim 1 of u, after $L_{2,c+1}$ the diagram is $(3,c)$-paired. Therefore, by Lemma 40, after $L_{1,c+1}$ the diagram is $(2,c)$-paired.

Now consider $c \leqslant b$, so $(3,c) \in D$. We consider three cases:

- Case 1: $(3,c)$ is moved by $L_{2,c}$ and not the last cell moved by $L_{2,c}$. Then $L_{2,c}$ performs a ladder move on $(3,c)$ moving it to $(2,c+1)$. Later, $L_{1,c+1}$ will move $(2,c+1)$. Since $L_{1,c+1}$ and $L_{2,c}$ moves the same number of cells, we know $L_{1,c+1}$ makes a ladder move on $(2,c+1)$. By Remark 29, the action of $L_{1,c+1}L_{2,c}$ is the same as first moving $(3,c)$ to $(1,c+2)$, and then perform $L_{2,c+1}L_{1,c+2}$. By Claim 1 of u, the diagram after applying $L_{1,c+1}$ is $(3,c)$-paired. Since $(2,c), (2,c+1)$ are not in the diagram, it is $(2,c)$-paired.

- Case 2: $(3,c)$ is the last cell moved by $L_{2,c}$. Then $L_{2,c}$ performs a K-ladder move on $(3,c)$ moving it to $(2,c+1)$. Later, $L_{1,c+1}$ will move $(2,c+1)$. Since $L_{1,c+1}$ and $L_{2,c}$ moves the same number of cells, we know $L_{1,c+1}$ makes K-ladder move on $(2,c+1)$. By Remark 29, the action of $L_{1,c+1}L_{2,c}$ can be described as follows: Remove $(3,c)$, perform $L_{2,c+2}L_{3,c}$, and then add cells $(3,c)$, $(2,c+1)$ and $(1,c+2)$. By Claim 1 of u, before adding those three cells, the diagram is $(3,c)$-paired. Thus, after adding these three cells, the diagram is $(2,c)$-paired.

- If $(3,c)$ is not moved by $L_{2,c}$, then $(3,c+1) \in D$. By Remark 29, applying $L_{1,c+1}L_{2,c}$ is the same as applying $L_{2,c+2}L_{3,c}$ while ignoring row 3. By Claim 1 of u, after the action of $L_{1,c+1}$, the diagram is $(2,c)$-paired. ☐

Lemma 45. *Claim 1 and 2 hold for all $N \in \mathbb{Z}_{>0}$.*

Proof. The claims are obvious when $N = 1$. Then we prove by induction on N. The inductive step is given by Lemma 43 and Lemma 44. ☐

Corollary 46. *In (1), each $L_{i,c}$ acts initially.*

Proof. Suppose $w \in S_n$ and we prove the corollary by induction on n. Suppose $w = (b,u)$. Since the corollary holds for u, we know $L_{i,c}$ in (1) acts initially when $i > 2$. Finally, each $L_{1,c}$ acts initially by Claim 2. ☐

Now we may prove the main results of this subsection using the two claims.

Proof of Proposition 23. We induct on n. The base case $n = 2$ is trivial. Now suppose $n > 2$ and take $v = (a, w) \in S_n$. Let D be the diagram obtained by putting a left-justified cells in row 1 of $\widehat{P}(w)^{\downarrow 1}$. The last iteration to compute $\widehat{P}(v)$ is to apply $L_{1,1} \cdots L_{1,n-2}L_{1,n-2}$ on D. For $c \in [a]$, since $L_{1,c}$ acts initially and $(1, c) \in D$, $L_{1,c}$ does not move any cells.

Now assume $c > a$. We want to show $L_{1,c}$ moves exactly $\mathsf{movecode}(w)_c$ cells. Let $w = (b, u)$ and let D' be the diagram obtained by putting b left-justified cells in the row 2 of $\widehat{P}(u)^{\downarrow 2}$. Then,

$$\begin{aligned}
&L_{1,1} \cdots L_{1,n-2}L_{1,n-1}(D)\\
=&(L_{1,1} \cdots L_{1,n-2}L_{1,n-1})(L_{2,1} \cdots L_{2,n-3}L_{2,n-2})(D')\\
=&(L_{1,1})(L_{1,2}L_{2,1}) \cdots (L_{1,n-1}L_{2,n-2})(D').
\end{aligned}$$

For $c > b$, by our induction hypothesis, applying $L_{2,c}$ moves exactly $\mathsf{movecode}(u)_c$ cells. Then by Claim 1, applying $L_{1,c+1}$ to D also moves exactly $\mathsf{movecode}(u)_c$ cells. Therefore the number of cells moved by $L_{1,c+1}$ is $\mathsf{movecode}(u)_c = \mathsf{movecode}(w)_{c+1}$. Now clearly each $L_{2,c}$ does not move any cells for $c \in [b]$. We know $L_{1,b+1}$ also moves no cells since the $(2, b+1)$-initial segment is empty. Therefore $L_{1,b+1}$ moves $0 = \mathsf{movecode}(w)_{b+1}$ cells.

Let c_0 be the largest in $[b]$ such that $\mathsf{movecode}(u)_{c_0} = 0$. Say $c_0 = 0$ if no such c_0 exists. For $c \in [b]$, by Lemma 26, we have

$$\mathsf{movecode}(w)_c = \begin{cases} \mathsf{movecode}(u)_c + 1 & \text{if } c \geqslant c_0. \\ \mathsf{movecode}(u)_c & \text{otherwise.} \end{cases}$$

We first inductively show that for $c = b, \ldots, c_0 + 1$, there is no cell at $(2, c+1)$ right before the action of $L_{1,c}$, so $L_{1,c}$ moves $(2, c)$. Moreover, $L_{1,c}$ moves $\mathsf{movecode}(w)_c > 2$ cells, so the move on $(2, c)$ is a ladder move. For $c = b$, we know $(2, b+1)$ is always empty. For $c_0 < c < b$, we know $L_{1,c+1}$ makes a ladder move on $(2, c+1)$, so $(2, c+1)$ is empty right before the action of $L_{1,c}$. Now for $c = b, \ldots, c_0 + 1$, after $L_{1,c}$ moves $(2, c)$, it behaves as if $L_{2,c}$ by Remark 29. Thus, the total number of cells moved is $\mathsf{movecode}(u)_c + 1 = \mathsf{movecode}(w)_c$.

Now consider L_{1,c_0} when $c_0 > 0$. Right before its action, $(2, c_0 + 1)$ is empty. Thus, L_{1,c_0} will first move $(2, c_0)$ to $(1, c_0 + 1)$. After that, the number of cells it moves is $\mathsf{movecode}(u)_{c_0}$, which is zero. Thus, the move on $(2, c_0)$ is a K-ladder move. Also, L_{1,c_0} moves $1 = \mathsf{movecode}(w)_{c_0}$ cell.

Finally, we prove by induction that for $c = c_0 - 1, \ldots, 1$, right before the action of $L_{1,c}$, the diagram contains $(2, c)$ and $(2, c+1)$. For the base case, right before the action of L_{1,c_0-1}, we know $(2, c_0)$ is in the diagram. Now assume right before the action of $L_{1,c}$, the diagram contains $(2, c)$ and $(2, c+1)$ for some $c < c_0$. Then $L_{1,c}$ will not move $(2, c)$. After the action of $L_{1,c}$, we know $(2, c)$ is still in the diagram. The inductive step is finished. Now by Remark 29, the action of $L_{1,c}$ moves the same number of cells as $L_{2,c}$ on the diagram without $(2, c)$ and $(2, c+1)$. Thus, $L_{1,c}$ makes $\mathsf{movecode}(u)_c = \mathsf{movecode}(w)_c$ moves. □

Proof of Corollary 25. Implied by Corollary 46 and Lemma 31. □

Acknowledgements

We thank Jianping Pan and Brendon Rhoades for carefully reading an earlier version of this paper and giving many useful comments. We are especially grateful to Anna Weigandt for suggesting this problem. This project was performed as a Research Experience for Undergraduates at UC San Diego in summer 2023.

References

[BB93] Nantel Bergeron and Sara Billey. RC-graphs and Schubert polynomials. *Experimental Mathematics*, 2(4):257–269, 1993.

[BJS93] Sara C Billey, William Jockusch, and Richard P Stanley. Some combinatorial properties of Schubert polynomials. *Journal of Algebraic Combinatorics*, 2(4):345–374, 1993.

[DMSD22] Matt Dreyer, Karola Mészáros, and Avery St. Dizier. On the degree of Grothendieck polynomials. arXiv:2209.00687, 2022.

[FK94] Sergey Fomin and Anatol N Kirillov. Grothendieck polynomials and the Yang-Baxter equation. In *Proc. formal power series and alg. comb*, pages 183–190, 1994.

[FUL92] W FULTON. Flags, Schubert polynomials, degeneracy loci, and determinantal formulas. *Duke Math. J.*, 65:381–420, 1992.

[Haf22] Elena S Hafner. Vexillary Grothendieck polynomials via bumpless pipe dreams. arXiv:2201.12432, 2022.

[KM05] Allen Knutson and Ezra Miller. Gröbner geometry of Schubert polynomials. *Annals of Mathematics*, pages 1245–1318, 2005.

[KMY09] Allen Knutson, Ezra Miller, and Alexander Yong. Gröbner geometry of vertex decompositions and of flagged tableaux. 2009.

[LRS06] Cristian Lenart, Shawn Robinson, and Frank Sottile. Grothendieck polynomials via permutation patterns and chains in the Bruhat order. *American Journal of Mathematics*, 128(4):805–848, 2006.

[LS82] 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.

[PSW21] Oliver Pechenik, David E Speyer, and Anna Weigandt. Castelnuovo-Mumford regularity of matrix Schubert varieties. arXiv:2111.10681, 2021.

[PY23] Jianping Pan and Tianyi Yu. Top-degree components of Grothendieck and Lascoux polynomials. arXiv:2302.03643, 2023.

[Ram85] Annamalai Ramanathan. Schubert varieties are arithmetically Cohen-Macaulay. *Inventiones mathematicae*, 80(2):283–294, 1985.

[RRR+21] Jenna Rajchgot, Yi Ren, Colleen Robichaux, Avery St. Dizier, and Anna Weigandt. Degrees of symmetric Grothendieck polynomials and Castelnuovo-Mumford regularity. *Proceedings of the American Mathematical Society*, 149(4):1405–1416, 2021.

[RRW23] Jenna Rajchgot, Colleen Robichaux, and Anna Weigandt. Castelnuovo-Mumford regularity of ladder determinantal varieties and patches of Grassmannian Schubert varieties. *J. Algebra*, 617:160–191, 2023.

[WY18] Anna Weigandt and Alexander Yong. The prism tableau model for Schubert polynomials. *Journal of Combinatorial Theory, Series A*, 154:551–582, 2018.