Digital Commons @ Assumption University

Honors Theses

Honors Program

2018

Schubert Polynomial Multiplication

Sara Amato
Assumption College

Follow this and additional works at: https://digitalcommons.assumption.edu/honorstheses

Part of the Algebraic Geometry Commons

Recommended Citation

Amato, Sara, "Schubert Polynomial Multiplication" (2018). *Honors Theses*. 34.
https://digitalcommons.assumption.edu/honorstheses/34

This Honors Thesis is brought to you for free and open access by the Honors Program at Digital Commons @ Assumption University. It has been accepted for inclusion in Honors Theses by an authorized administrator of Digital Commons @ Assumption University. For more information, please contact digitalcommons@assumption.edu.

SCHUBERT POLYNOMIAL MULTIPLICATION

SARA AMATO
A THESIS UNDER THE MENTORSHIP OF JOSEPH ALFANO

Abstract. Schur polynomials are a fundamental object in the field of algebraic combinatorics. The product of two Schur polynomials can be written as a sum of Schur polynomials using non-negative integer coefficients. A simple combinatorial algorithm for generating these coefficients is called the Littlewood-Richardson Rule. Schubert polynomials are generalizations of the Schur polynomials. Schubert polynomials also appear in many contexts, such as in algebraic combinatorics and algebraic geometry. It is known from algebraic geometry that the product of two Schubert polynomials can be written as a sum of Schubert polynomials using non-negative integer coefficients. However, a simple combinatorial algorithm for generating these coefficients is not known in general. Monk's Rule is a known algorithm that can be used in specific cases. This research seeks to identify more algorithms for the multiplication of Schubert polynomials. In this thesis, I will provide a brief overview of Schur polynomials and Schubert polynomials. Also, I will present diagrams called 'pipe-dreams' to illustrate Schubert polynomials and establish a connection to Schur polynomials. Our main result is in Schubert polynomial multiplication. I will present two algorithms for Schubert polynomial multiplication, which generalize Monk's rule in specific cases.

Contents

1. Symmetric Polynomials and Schur Polynomials

1.1. **Symmetric Polynomials.** We will define symmetric polynomials, initially, with positive integer indices. More generally, later in the paper, we will construct them to be indexed

Date: April 27, 2018.

by partitions of integers. A symmetric polynomial, denoted by s_k, is a polynomial in n variables, $\{x_1, x_2, \ldots, x_n\}$, such that if any of the variables are transposed, the same polynomial is obtained. For example, for $\{x_1, x_2, x_3\}$, there are $3! = 6$ permutations of subscripts; s_k would have to be unchanged if we apply any of the permutations to the subscripts. In this example, the symmetric group, S_3 is the set $\{123, 132, 213, 231, 312, 321\}$. In this notation, we write the set of outputs when we permute the sequence $1, 2, 3$.

Elementary symmetric polynomials, denoted by e_k are the sum of all k-letter monomials, each using distinct variables. For example, the elementary symmetric polynomials for $\{x_1, x_2, x_3\}$ are:

$$\begin{aligned} e_1 &= x_1 + x_2 + x_3 \\ e_2 &= x_1x_2 + x_1x_3 + x_2x_3 \\ e_3 &= x_1x_2x_3 \end{aligned}$$

The complete homogeneous polynomials, denoted by h_k are the sum of all k-letter monomials, with repeated variables permitted. For example, the complete homogeneous polynomials for $\{x_1, x_2, x_3\}$ are:

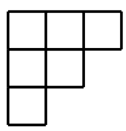
$$\begin{aligned} h_1 &= x_1 + x_2 + x_3 \\ h_2 &= x_1x_2 + x_1x_3 + x_2x_3 + x_1x_1 + x_2x_2 + x_3x_3 \\ h_3 &= x_1x_2x_3 + x_1x_1x_1 + x_1x_1x_2 + x_1x_1x_3 + x_1x_2x_2 + x_1x_3x_3 + x_2x_2x_2 + x_2x_2x_3 + x_2x_3x_3 + x_3x_3x_3. \end{aligned}$$

1.2. **Schur Polynomials.** Here is some terminology that will be helpful for the definition of a Schur polynomial:

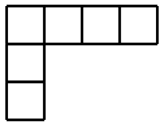
- A *partition* is denoted by λ, which we write as $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$, such that $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n \geq 0$. In other words λ is a weakly decreasing sequence of non-negative integers. A partition of n is a partition whose parts sum up to equal n. We omit 0's for convenience. For example, the partitions for $n = 6$ are the following:

$\lambda = 6$		
$\lambda = 5, 1$		
$\lambda = 4, 2$	$\lambda = 4, 1, 1$	
$\lambda = 3, 3$	$\lambda = 3, 2, 1$	$\lambda = 3, 1, 1, 1$
$\lambda = 2, 2, 2$	$\lambda = 2, 2, 1, 1$	$\lambda = 2, 1, 1, 1, 1$
$\lambda = 1, 1, 1, 1, 1, 1$		

- We define the elementary symmetric function $e_\lambda := \sum_{i=1}^{n} e_{\lambda i}$, where $e_0 = 1$.
- We define $h_\lambda := \sum_{i=1}^{n} h_{\lambda i}$, where $h_0 = 1$.
- A *Ferrers diagram* is a diagram representing partitions as patterns of dots or boxes. If a partition is $\lambda = (a, b, c)$, the first row will have a dots/boxes, the second row will have b dots/boxes, and the third row will have c dots/boxes. For example, the Ferrers diagram for $\lambda = (3, 2, 1)$ would look like:



Here is another example. This is the Ferrers diagram for $\lambda = (4, 1, 1)$:



- A *semi-standard Young tableau* is obtained by filling in the boxes of a Ferrers diagram with symbols from some alphabet. We require that each row has a weakly increasing sequence from left to right and each column has a strictly increasing sequence from top to bottom. The *weight* of the tableau is found by counting the number of times that each distinct number is repeated. For example, using the alphabet $\{1, 2, 3\}$, the semi-standard Young tableaux with shape $(2, 1, 1)$ are the following:

$$\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 2 \\ \cline{1-1} 3 \\ \cline{1-1}\end{array} \qquad \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 2 \\ \cline{1-1} 3 \\ \cline{1-1}\end{array} \qquad \begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1} 3 \\ \cline{1-1}\end{array}$$

Schur polynomials generalize elementary symmetric polynomials and complete homogeneous polynomials. Here is a combinatorial definition:

(1) Each semi-standard Young tableau, T, determines a monomial, x^T defined by the rule that constructs x^T by the following product:

$$\prod_{i \in T} x_i$$

(2) The Schur polynomial for partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$, is the sum of monomials, such that, $s_\lambda(x_1, x_2, \ldots, x_n) = \sum_T x^T$, where the summation is over all semi-standard Young tableaux, T, of shape, λ.

For example, let $\{x_1, x_2, x_3\}$ be our alphabet. Here are the semi-standard young tableaux of shape $(3, 0, 0)$:

$$\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline\end{array} \quad \begin{array}{|c|c|c|}\hline 2 & 2 & 2 \\ \hline\end{array} \quad \begin{array}{|c|c|c|}\hline 3 & 3 & 3 \\ \hline\end{array} \quad \begin{array}{|c|c|c|}\hline 1 & 1 & 2 \\ \hline\end{array} \quad \begin{array}{|c|c|c|}\hline 1 & 1 & 3 \\ \hline\end{array}$$

$$\begin{array}{|c|c|c|}\hline 1 & 2 & 2 \\ \hline\end{array} \quad \begin{array}{|c|c|c|}\hline 1 & 3 & 3 \\ \hline\end{array} \quad \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline\end{array} \quad \begin{array}{|c|c|c|}\hline 2 & 2 & 3 \\ \hline\end{array} \quad \begin{array}{|c|c|c|}\hline 2 & 3 & 3 \\ \hline\end{array}$$

Hence, the following Schur polynomial can be obtained by summing the tableaux monomials:

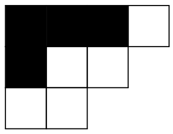
$$s_{(3,0,0)} = x_1^3 + x_2^3 + x_3^3 + x_1^2 x_2 + x_1^2 x_3 + x_1 x_2^2 + x_1 x_3^2 + x_1 x_2 x_3 + x_2^2 x_3 + x_2 x_3^2$$

1.3. **Littlewood-Richardson Rule.** This rule gives a combinatorial description of the coefficients that arise when decomposing a product of Schur functions. The product of two Schur functions can be written as a linear combination of Schur polynomials with non-negative integral coefficients. These coefficients are given by the *Littlewood-Richardson Rule*, which is:

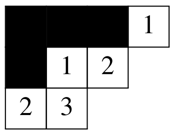
$$s_\lambda s_\mu = \sum_\nu c^\nu_{\lambda,\mu} s_\nu$$

where λ and μ identify the Schur functions being multiplied and ν identifies the Schur function for which the coefficient is being found for. Further, the L-R rule states that $c^\nu_{\lambda,\mu}$ is the number of L-R tableaux of skew shape ν/λ and weight μ.

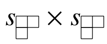
The *skew shape*, ν/λ is the shape obtained by drawing ν and then deleting λ. For example let $\nu = (4, 3, 2)$ and $\lambda = (3, 1)$. We obtain the following skew shape:



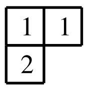
The *L-R tableau* is a filling of the skew shape of the Ferrers diagram with numbers whose multiplicity is given by weight μ, such that the entries are strictly increasing in each column and the entries are weakly increasing in each row. There is one final requirement. When reading the entries from right to left and then top to bottom we need to obtain an *L-R word*, which means that in every initial part of the sequence any number i occurs at least as often as the number $i + 1$. Here is the L-R tableau for our previous example.



Here is a diagram of the Littlewood-Richardson Rule with Schur functions. This illustrates that our skew shape will have weight, μ, and will start with λ and add onto it 1's and 2's from μ. We are multiplying:

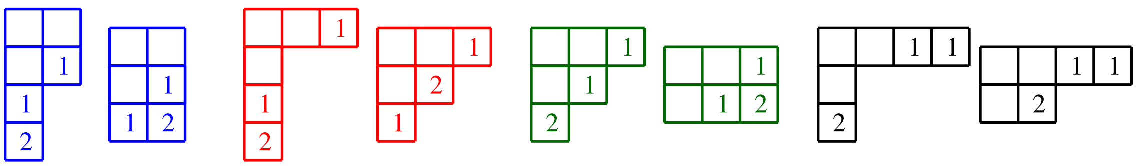


(1) Call the first Schur function s_λ and the second, s_μ. Fill s_μ with its most elementary filling:



Start with shape, $\lambda =$ .

(2) Append to λ, the broken rows, which is a collection of entries, no two of which are in the same column. First, do this with 1's and then with 2's, such that we obtain an L-R word, which means that in every initial part of the sequence any number i occurs at least as often as the number $i + 1$.



The resulting Schur polynomials are $s_{4,2} + s_{4,1,1} + s_{3,3} + 2s_{3,2,1} + s_{3,1,1,1} + s_{2,2,2} + s_{2,2,1,1}$.

1.4. **Pieri's Rule.** Pieri's formula is a specialized version of the L-R rule to the case where s_μ is the Schur function where μ contains just one row, call its length k. Pieri's formula describes the product of a Schur polynomial by a complete homogeneous symmetric function. In terms of Schur functions, s_λ, indexed by partition λ states that:

$$s_\lambda h_k = \sum_\nu s_\nu$$

where shape ν/λ has no two cells in the same column. Pieri's rule for multiplying an elementary symmetric polynomial with a Schur polynomial is:

$$s_\lambda e_k = \sum_\nu s_\nu$$

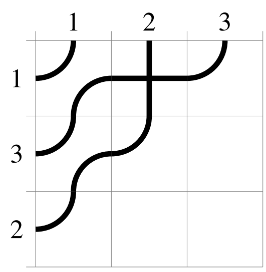
where ν is summed over all shapes such that ν/λ has no two cells in the same row.

2. Schubert Polynomials

A Schubert polynomial, written as $\mathfrak{S}_w$, is a mathematical object in the field of combinatorics. Schubert polynomials have their basis in permutations because every Schubert polynomial is indexed by a specific ordering of numbers. For example, to find all Schubert polynomials for $n = 3$, we take the numbers $1, 2, 3$ and find every possible ordering of these numbers. Hence, there are $3 \times 2 \times 1$ or 6 Schubert polynomials in $n = 3$. They are $\mathfrak{S}_{123}$, $\mathfrak{S}_{132}$, $\mathfrak{S}_{213}$, $\mathfrak{S}_{231}$, $\mathfrak{S}_{312}$, and $\mathfrak{S}_{321}$.

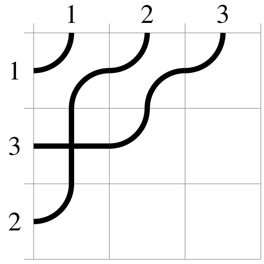
Further, a Schubert polynomial can be constructed using a combinatorial object called a pipe-dream. A pipe-dream is a $n \times n$ square in which the numbers $\{1, 2, 3, \ldots, n\}$, go along the top of the square reading left to right. The Schubert polynomial's permutation w goes down the left side reading from top to bottom. The square's entries get filled with crosses $+$ or elbows ⌟⌜, matching equal numbers on the top of the diagram to their matches on the side of the diagram. The only condition is that no two pipes cross twice. We make graphs for all admissible pipe-dreams for w and add the monomials constructed from each one together. The monomial constructed from a given pipe-dream is $x_0^{a_0} x_1^{a_1} \ldots x_{n-1}^{a_{n-1}}$ where a_i is the number of crosses in the i^{th} row.

Here is a pipe-dream for $w = 132$.



This pipe-dream represents the monomial $x_0^1 x_1^0 x_2^0$, which we simplify to x_0, since there is one cross in the first row.

Here is another pipe-dream for $w = 132$.



This pipe-dream represents the monomial $x_0^0 x_1^1 x_2^0$, which we simplify to x_1, since there is one cross in the second row. These are the only two pipe-dreams that satisfy the requirements; therefore, the Schubert polynomial, $\mathfrak{S}_{132}$ is $x_0 + x_1$.

Here is some terminology that will be helpful for the algebraic definition of a Schubert polynomial:

- S_n is the notation for the symmetric group, which is the collection of all permutations of $\{1, 2, \ldots, n\}$.
- Let $[n] = \{1, 2, \ldots, n\}$ for any positive integer n. We say that σ is a *permutation* of $[n]$ if σ is a function from $[n]$ to $[n]$ that is a one-to-one correspondence. In other words, σ is an ordering of $\{1, 2, \ldots, n\}$. Here is an example of a permutation σ for the case $n = 5$, written in *two-line notation*, that is each column is an input-output pair, $\begin{bmatrix} \text{i} \\ \sigma(i) \end{bmatrix}$.

$$\sigma = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 3 & 5 & 1 & 2 \end{bmatrix}$$

  Here is another example of a permutation, which we will call τ.

$$\tau = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 5 & 4 \end{bmatrix}$$

  We define the product of permutations σ and τ by the rule: $\tau\sigma$ is the permutation whose output is given by $(\tau\sigma)(i) = \tau(\sigma(i))$ for every i in $\{1, 2, \ldots, n\}$. In other words, we plug-in the output of function σ into function τ.

$$\tau\sigma = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 3 & 5 & 1 & 2 \\ 5 & 1 & 4 & 2 & 3 \end{bmatrix}$$

  Written in two-line notation, that is writing each input-output of $\tau\sigma$.

$$\tau\sigma = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 4 & 2 & 3 \end{bmatrix}$$

  and in *one-line notation*, that is the sequence of outputs if we assume the inputs are written in numerical order as $\{1, 2, \ldots, n\}$, $\tau\sigma = 51423$.
- We call the permutation that sends i to j (and vice versa) and sends every other number to itself the transposition $t_{i,j}$. We denote the *transposition* of i and $i + 1$ by s_i, which we call an *adjacent transposition*. The *length* of permutation σ is the minimum number of adjacent transpositions, s_i, needed to multiply together to equal σ.

  For example, let us find the length of $\tau\sigma$ from our previous example. We start by looking at the original ordering 12345. First, we need to move the 5 to the front. This can be done using a sequence of adjacent swaps. We will call each of these swaps, s_i. We are letting our s_i denote a transposition in the i^{th} position. To move 5 to the front we can swap 5 with the number adjacent to it, 4. This is denoted by s_4. Now, we have 12354. Now, we swap 5 with 3, since these two are now adjacent. This is denoted by s_3. Now, we have 12534. Next, we swap 5 with 2, since they are now in adjacent positions, which is denoted by s_2. This yields 15234. Finally we swap 5 with 1, since they are now in adjacent positions, which is denoted by s_1. This yields 51234. We, now, have to get 4 into the third spot. To do this, we first have to swap 4 with 3, which is denoted by s_4 and yields 51243. Finally, we swap 4 with 2, since they are now in adjacent positions, which is denoted by s_3. Now, we have 51423, which is $\tau\sigma$. The length of $\tau\sigma$ is found by adding up all of the adjacent transpositions we performed. The transpositions we performed were, s_4,s_3,s_2,s_1,s_4, and s_3. Thus, the length is 6.

- A *reduced word* is found by putting together all of the adjacent transpositions we performed to achieve the permutation. So, a reduced word for $\tau\sigma$ is $s_3 s_4 s_1 s_2 s_3 s_4$. We use this ordering because $\tau\sigma$ and each s_i are functions.
- We often use w to stand for permutation. It can be proved that the length of our permutation, w, is well-defined: every reduced word of w has the same length. However, there may be more than one satisfactory reduced word for a given permutation. Let w_0 be the permutation 321 in S_3. To find the reduced word and length we have to perform the adjacent transpositions. We start with the initial ordering 123. We apply s_2. The output is 132. Then we apply s_1. The output is 312. Then we apply s_2. The output is 321. The length of w_0 is 3 and the reduced word is $s_2 s_1 s_2$. We can also perform the following adjacent transpositions. We start with 123. We apply s_1. The output is 213. Then we apply s_2. The output is 231. Then we apply s_1. The output is 321. Therefore, the length of w_0 is still 3, but the reduced word is $s_1 s_2 s_1$.
- The *inverse* of σ is the inverse function, that is the one-to-one correspondence that sends the sequence of outputs to the sequence of inputs. The two-line notation of σ^{-1} is obtained by swapping the second and first lines.

$$\sigma^{-1} = \begin{bmatrix} 4 & 3 & 5 & 1 & 2 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

which we write in more standard form where the inputs are ordered $1, 2, \ldots, n$.

$$\sigma^{-1} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 2 & 1 & 3 \end{bmatrix}$$

or in one line notation as, 45213.

Here is an algebraic definition for Schubert polynomials:

(1) Let w_0, in $S_n = n, n-1, \ldots, 2, 1$. We define the Schubert polynomial, $\mathfrak{S}_{w_0} = x_1^{n-1} x_2^{n-2} \ldots x_{n-1}^1 x_n^0$.
(2) Define the difference quotient, $\mathfrak{d}_i$, which acts on the polynomial $P(x_1, x_2, \ldots, x_n)$, by the following rule. Let

$$\mathfrak{d}_i P(x_1, x_2, \ldots, x_n) = \frac{P(x_1, \ldots, x_i, x_{i+1}, \ldots, x_n) - P(x_1, \ldots, x_{i+1}, x_i, \ldots, x_n)}{x_i - x_{i+1}}$$

for any $i \in \{1, 2, \ldots, n-1\}$.

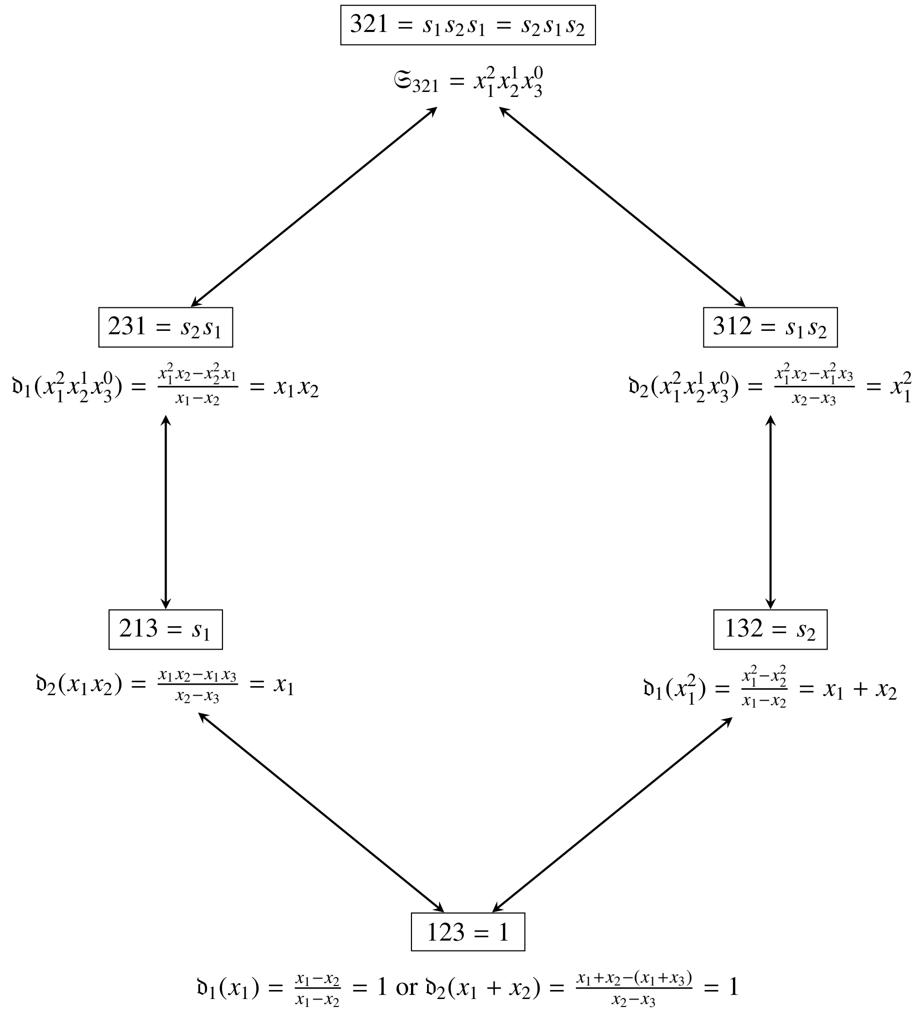
For any permutation w and for any i in $\{1, 2, \ldots, n-1\}$, we define the Schubert polynomial that is indexed by ws_i by the rule:

$$\mathfrak{d}_i \mathfrak{S}_w = \mathfrak{S}_{ws_i}$$

as long as ws_i has smaller length as a permutation than w.

The following diagram provides an example of using the divided difference operator for each of the permutations of $n = 3$.

FIGURE 1. The Schubert polynomials of $n = 3$ by Divided Difference



Please note: To each of these Schubert polynomials, we adapt the answer to our notation, which starts with x_0, instead of x_1. Please note: Some of the sources that will be referenced in this thesis will start their variable sequence with x_1, as in the above definition. Our computing software, SageMath, starts its variable sequence with x_0. In this paper, we adopt the latter convention.

2.1. **Monk's rule.** Suppose we want to multiply an arbitrary Schubert polynomial, $\mathfrak{S}_w$ by the Schubert polynomial $\mathfrak{S}_{s_r}$. w is an arbitrary permutation in S_n whose output sequences are $w_1, w_2, \ldots, w_n$. s_r is the permutation that exchanges a number, r, and the subsequent number, $r + 1$. An example of s_r would be 132 because this permutation exchanges the number 2 with 3.

Given a specific w and a specific r, we consider all ordered pairs, (i, j), such that $i \leq r < j$, with the restriction that $j \leq n + 1$ with n being the number of elements in the permutation. In our example of 132, $n = 3$. For each (i, j) pair, we determine if it is admissible. It is admissible if $w_i < w_j$. Also, for each value k that lies in the interval $i < k < j$, the value of w_k must be either greater than w_j or less than w_i. The inequality would have to be $w_k < w_i < w_j$ or $w_i < w_j < w_k$. For each admissible (i, j) pair we construct an output, which we will call w', which is obtained from w by swapping w_i and w_j. Monk's rule asserts that the product $\mathfrak{S}_w\ \mathfrak{S}_{s_r}$ is the sum of all w' that were constructed in the previous step.

Let's try an example. We take s_r to be 132. Observe $r = 2$. Let the Schubert polynomial that it is multiplying be $\mathfrak{S}_{213}$. To find w', we first have to list every (i, j) pair that satisfies the inequality $i \leq 2 < j$ with $j \leq 4$.

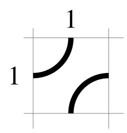
- $(1, 3)$. This would swap the numbers 2 and 3 in our original permutation. Since there is no entry located between entries 1 and 3 that has a value between 2 and 3, we can swap the i and j values to obtain $w' = 312$.
- $(1, 4)$. The permutation 213 in $n = 3$ is the same as the permutation 2134 in $n = 4$. We can consider an entry one past the n^{th} position when we are applying Monk's rule. This would swap the numbers 2 and 4 in our original permutation. Since entry 3 is located in between entries 1 and 4 and has a value between 2 and 4, this (i, j) pair is not admissible.
- $(2, 3)$. This would swap the numbers 1 and 3 in our original permutation. Since there is no entry located in between entries 2 and 3, this ordered pair is admissible and we can swap w_i and w_j to obtain the permutation $w' = 231$.
- $(2, 4)$. This would swap the numbers 1 and 4 in our original permutation. Since entry 3 is located in between entries 2 and 4 and has a value between 1 and 4, this (i, j) pair is not admissible.

Therefore, $\mathfrak{S}_{132}\mathfrak{S}_{213} = \mathfrak{S}_{312} + \mathfrak{S}_{231}$. Monk's rule is an algorithm that has been proven by multiplying polynomials with algebra and expressing the result as a sum of Schubert polynomials. This is a nice property of Schubert polynomials with regard to products and our research seeks to find other nice multiplication algorithms.

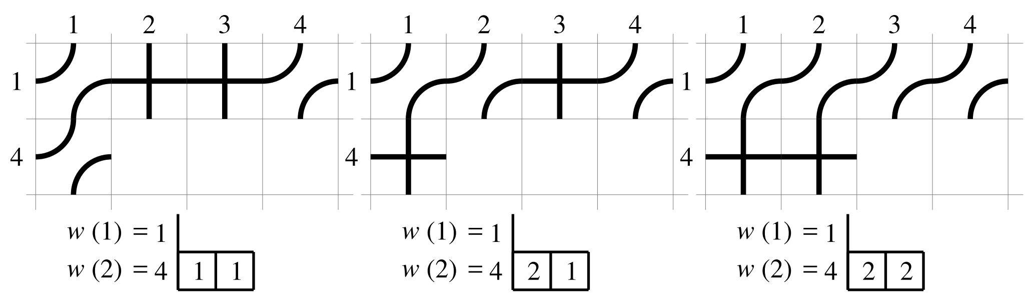
3. Connection between Schur polynomials and Schubert polynomials

It is well known that every Schur polynomial is a Schubert polynomial. Here we present a bijection that accounts for going from a Schubert polynomial, $\mathfrak{S}_w$, where w is the permutation with exactly one descent, to a certain Schur polynomial. More precisely, every Schur polynomial is equal to a Schubert polynomial, $\mathfrak{S}_w$, where w is a permutation with exactly one descent. A descent is a pair $(i, i + 1)$, such that $i < i + 1$ and $w_i > w_{i+1}$. Let $w = 14823567$

(1) Construct every pipe-dream for $w(1) = 1$.
 (a) There is only one way:

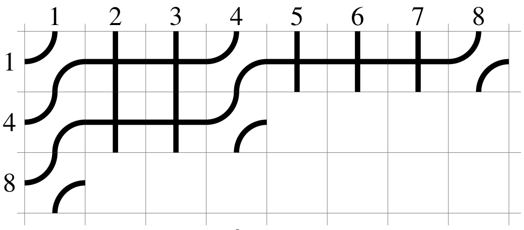


 (b) Then construct a pipe for $w(2) = 4$. There are three ways. Also, after the pipes are constructed, pause here to record, for each pipe we have constructed, the sequence of rows where is has a crossing, $+$

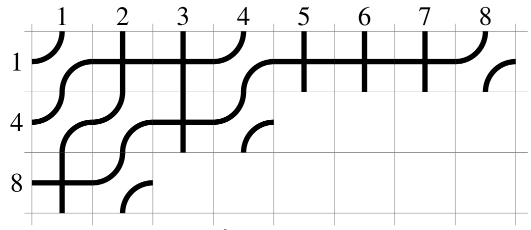


Observe that for $w(2)$ the sequence of row entries we have recorded is every possibly weakly decreasing sequence with values in the interval $\{1, 2\}$, whose sequence-length is 2 (since the pipe that connects number 4 must cross number 3 and number 2, so there are 2 crossings).

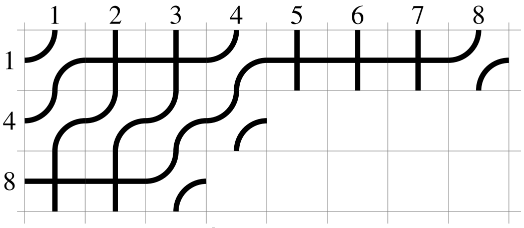
(c) Now, construct a pipe for $w(3) = 8$. There are many ways:



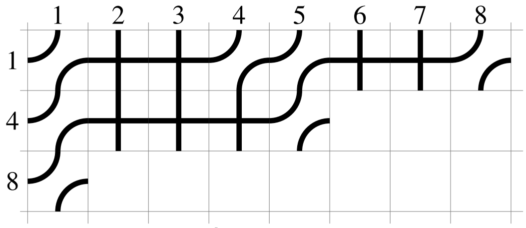
$w(1) = 1$					
$w(2) = 4$	1	1			
$w(3) = 8$	2	2	1	1	1



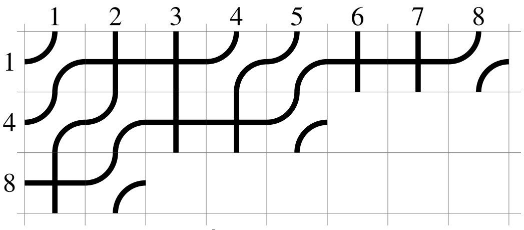
$w(1) = 1$					
$w(2) = 4$	1	1			
$w(3) = 8$	3	2	1	1	1



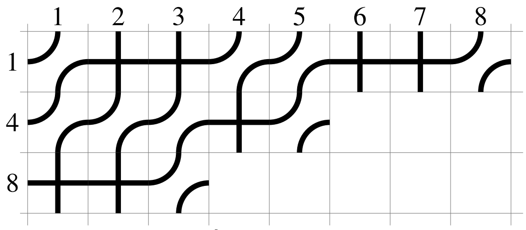
$w(1) = 1$					
$w(2) = 4$	1	1			
$w(3) = 8$	3	3	1	1	1



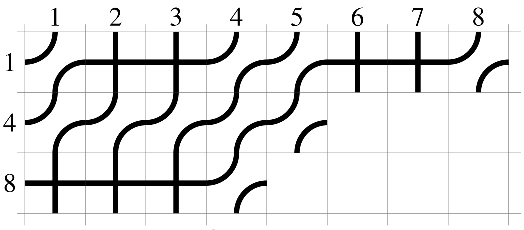
$w(1) = 1$					
$w(2) = 4$	1	1			
$w(3) = 8$	2	2	2	1	1



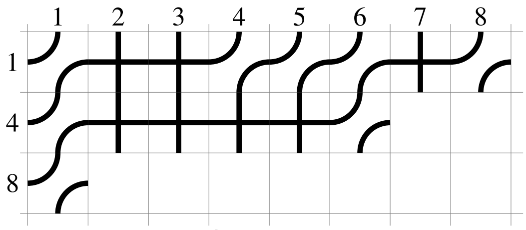
$w(1) = 1$					
$w(2) = 4$	1	1			
$w(3) = 8$	3	2	2	1	1



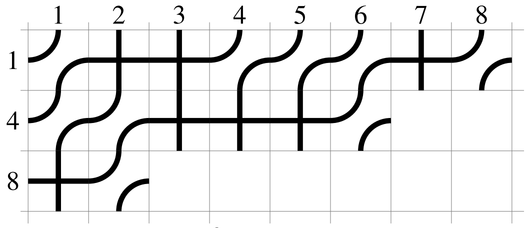
$w(1) = 1$					
$w(2) = 4$	1	1			
$w(3) = 8$	3	3	2	1	1



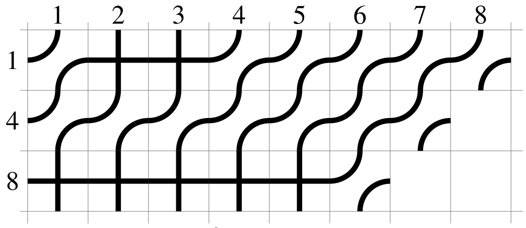
$w(1) = 1$					
$w(2) = 4$	1	1			
$w(3) = 8$	3	3	3	1	1



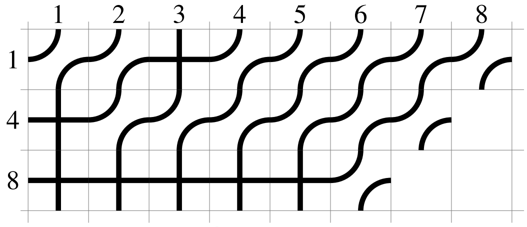
$w(1) = 1$					
$w(2) = 4$	1	1			
$w(3) = 8$	2	2	2	2	1



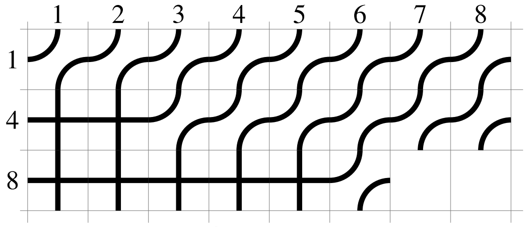
$w(1) = 1$					
$w(2) = 4$	1	1			
$w(3) = 8$	3	2	2	2	1



$w(1) = 1$					
$w(2) = 4$	1	1			
$w(3) = 8$	3	3	3	3	3



$w(1) = 1$					
$w(2) = 4$	2	1			
$w(3) = 8$	3	3	3	3	3



$w(1) = 1$					
$w(2) = 4$	2	2			
$w(3) = 8$	3	3	3	3	3

Note to reader: This is only a partial listing of all of the pipes that can be achieved. Observe that for $w(3)$, the sequence of row entries is every possible weakly decreasing sequence in $\{1, 2, 3\}$, whose sequence length is 5. Each column in this tableau is strictly increasing(since pipe number 2 must first be crossed by number 4 before it gets crossed by number 8...)

(d) Now, construct a pipe for the remaining numbers $2, 3, 5, 6, 7$. These have no additional crossings.

(2) Now apply the one-to-one operation; negate each entry.

(3) Now apply the one-to-one operation; invert the recording tableau.In our example, the $w(1)$ row will now be below the $w(2)$ row, which will now be below the $w(3)$ row.

(4) Now apply the one-to-one operation; add to each entry $r + 1$. r is the location of the descent in w. In this case, $r = 3$, so $r + 1 = 4$.

2	2	3	3	3
3	3			

1	2	3	3	3
3	3			

1	1	3	3	3
3	3			

2	2	2	3	3
3	3			

1	2	2	3	3
3	3			

1	1	2	3	3
3	3			

1	1	1	3	3
3	3			

2	2	2	2	3
3	3			

1	2	2	2	3
3	3			

1	1	1	1	1
3	3			

1	1	1	1	1
2	3			

1	1	1	1	1
2	2			

...

This is a partial listing of all tableaux with shape $(w(r)-r, w(r-1)-(r-1), \ldots, w(2)-2, w(1)-1)$. These are weakly increasing in each row, and strictly increasing in each column, and whose entries in the top row lie in $\{1, 2, \ldots, r\}$, and in the next row lie in $\{2, \ldots, r\}$, etc. To state the correspondence precisely: the Schubert polynomial indexed by w, where w is a permutation in S_n with exactly one descent at location r, is equal to the Schur function, s_λ, where λ is the permutation whose parts are $(w_r - r, w_(r-1) - (r-1), \ldots, w_2 - 2, w_1 - 1)$

4. Specific Algorithms for cases $n = 3$ and $n = 4$

In this section, we investigate the product of two Schubert polynomials, $\mathfrak{S}_w \mathfrak{S}_\sigma$, and write algorithms to construct the output, in the cases where σ is any permutation in the symmetric group S_3 or S_4, and $\mathfrak{S}_w$ is arbitrary. Let us start with the Schubert polynomials of $n = 3$.

4.1. **The Trivial case: σ is the identity permutation.** Consider the case where σ is the identity element of S_3. In two-line notation, we write:

$$\sigma = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}.$$

This Schubert polynomial $\mathfrak{S}_{123}$ is equal to 1. So the corresponding multiplication rule is trivial: $\mathfrak{S}_w \mathfrak{S}_{123} = \mathfrak{S}_w$.

4.2. **Other Easy cases: σ is a transposition of two consecutive numbers.** Next consider the cases where σ is a transposition s_i that permutes the consecutive numbers i and $i+1$. The multiplication rules are given by Monk's formula, which we will write explicitly.

4.2.1. *the case where $\sigma = s_1$.* In the case where σ permutes 1 and 2, we have

$$\sigma = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{bmatrix}.$$

This Schubert polynomial $\mathfrak{S}_{213}$ is equal to x_0, in the alphabet $\{x_0, x_1, \ldots\}$. The corresponding multiplication rule is given by Monk's formula:

$$\mathfrak{S}_w \mathfrak{S}_{213} = \sum_{\substack{i \le 1 < j \\ l(wt_{ij}) = l(w)+1}} \mathfrak{S}_{wt_{ij}},$$

where t_{ij} denotes the transposition that permutes the numbers i and j, and $l(w)$ denotes the length of w as a product of transpositions of adjacent numbers.

4.2.2. *the case where $\sigma = s_2$.* In the case where σ permutes 2 and 3, we have

$$\sigma = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{bmatrix}.$$

This Schubert polynomial $\mathfrak{S}_{132}$ is equal to $x_0 + x_1$. The corresponding multiplication rule is given by Monk's formula:

$$\mathfrak{S}_w \mathfrak{S}_{132} = \sum_{\substack{i \le 2 < j \\ l(wt_{ij}) = l(w)+1}} \mathfrak{S}_{wt_{ij}}.$$

4.3. **Intermediate cases: σ is a product of two transpositions s_i.** Now consider the cases where σ is a permutation whose length is 2.

4.3.1. *the case where* $\sigma = 312$. We first examine the case is where σ equals 3 1 2, that is

$$\sigma = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix}.$$

This Schubert polynomial $\mathfrak{S}_{312}$ is equal to x_0^2. Multiplying $\mathfrak{S}_w$ by this polynomial is equivalent to multiplying by x_0 twice, so a simple procedure for evaluating the product $\mathfrak{S}_w\mathfrak{S}_{312}$ is to evaluate $\mathfrak{S}_{213}(\mathfrak{S}_{213}\mathfrak{S}_w)$. In other words, one performs the following algorithm.

(1) Apply Monk's formula, with $r = 1$, to evaluate the product $\mathfrak{S}_{213}\mathfrak{S}_w$.
(2) Then apply Monk's formula, with $r = 1$, to evaluate the product of $\mathfrak{S}_{213}$ with each of the Schubert polynomials of the output of the previous step.

$$\mathfrak{S}_w\mathfrak{S}_{312} = \sum_{\substack{a\le 1<b \\ l(wt_{ab})=l(w)+1}} \left(\sum_{\substack{c\le 1<d \\ l(wt_{ab}t_{cd})=l(wt_{ab})+1}} \mathfrak{S}_{wt_{ab}t_{cd}} \right).$$

4.3.2. *the case where* $\sigma = 231$. We next examine the case where σ equals 2 3 1, that is

$$\sigma = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{bmatrix}.$$

This Schubert polynomial $\mathfrak{S}_{231}$ is equal to x_0x_1. Multiplying $\mathfrak{S}_w$ by this polynomial is equivalent to multiplying by the difference $(x_0 + x_1)x_0 - x_0^2$, so a procedure for evaluating the product $\mathfrak{S}_w\mathfrak{S}_{231}$ is to evaluate $\mathfrak{S}_{132}(\mathfrak{S}_{213}\mathfrak{S}_w) - \mathfrak{S}_{213}(\mathfrak{S}_{213}\mathfrak{S}_w)$.
Proposition. It is sufficient to perform the following algorithm.

(1) Apply Monk's formula with $r = 1$ to our given Schubert polynomial $\mathfrak{S}_w$.
(2) Then apply Monk's formula, with $r = 2$ and restrict to the specific case $i = 2$, to each of the Schubert polynomials $\mathfrak{S}_{w'}$ of the output of the previous step.
(3) From this final output, discard each Schubert polynomial $\mathfrak{S}_{w''}$ for which the sequence of steps we have just performed, $w \mapsto w' \mapsto w''$, permutes exactly three values according to the form $(A, C, \dots, B, \dots) \mapsto (C, A, \dots, B, \dots) \mapsto (C, B, \dots, A, \dots)$ where $A < B < C$. The first ellipsis represents any sequence of numbers not in the interval $[A, B]$, the second ellipsis represents any sequence of numbers, and each ellipsis sequence may be empty.

Proof. We seek to evaluate the multiplication of an arbitrary Schubert polynomial, $\mathfrak{S}_w$, by a product. This product is $((x_0 + x_1) - (x_0))x_0$. This equals $(\mathfrak{S}_{132} - \mathfrak{S}_{213})\mathfrak{S}_{213}\mathfrak{S}_w$. This can be done by the following algebraic procedure.

(1) Evaluate the product $\mathfrak{S}_{213}$ $\mathfrak{S}_w$ by applying Monk's formula with $r = 1, i = 1$ to $\mathfrak{S}_w$. The output is a sum of Schubert polynomials; let $\mathfrak{S}_{w'}$ denote any arbitrary one of these terms.
(2) Evaluate $(\mathfrak{S}_{132} - \mathfrak{S}_{213})(\mathfrak{S}_{213}\mathfrak{S}_w)$ by applying the following.
    (a) Evaluate $\mathfrak{S}_{213}(\mathfrak{S}_{213}\mathfrak{S}_w)$ by applying Monk's formula with $r = 1$ to each $\mathfrak{S}_{w'}$. We obtain, for each input $\mathfrak{S}_{w'}$, an output that is a sum of Schubert polynomials; let $\mathfrak{S}_{w''}$ denote any arbitrary one of these terms.
    (b) Evaluate $\mathfrak{S}_{132}(\mathfrak{S}_{213}\mathfrak{S}_w)$ by applying Monk's formula with $r = 2$ to each $\mathfrak{S}_{w'}$. Again we obtain, for each $\mathfrak{S}_{w'}$, a sum of Schubert polynomials.
    (c) Subtract the output of this Step 2a from the output of this Step 2b.

We have a sequence $w \mapsto w' \mapsto w''$. Let's use this sequence for step $2a$. We have a sequence that we will call $W \mapsto W' \mapsto W''$, which we will use for step $2b$ (the ones that we will "subtract"). In this algebraic procedure, let us pause at each step to see more clearly the terms that we will subtract. There are two cases:

(1) Case 1: Suppose that in step 2a we have used an ordered pair, call it (i, j) such that $i = 1$ and $j \geq 3$. We have a sequence $w \mapsto w' \mapsto w''$. In step 2b, this same ordered pair is admissible and, beginning with the same input, we obtain an identical sequence, $W \mapsto W' \mapsto W''$. These terms cancel in step 2c. Note: This is why in our algorithm, when we perform Monk's rule with $r = 2$, we restrict to $i = 2$.

(2) Case 2: Suppose that in step 2a, to exhaust all cases, we use the ordered pair (i, j), where $i = 1$ and $j = 2$. For this to be the admissible (i, j) pair in our first step, we must have used the ordered pair (i, j) such that $i = 1$ and $j \geq 3$. The sequence $w \mapsto w' \mapsto w''$ must have the form $(A, C, \ldots B, \ldots) \mapsto (B, C, \ldots, A, \ldots) \mapsto (C, B, \ldots, A, \ldots)$ for $A < B < C$. In step 2b a sequence arises, which begins with the same input and we are assuming cancellation, so it has the same output. The sequence that we obtain, $W \mapsto W' \mapsto W''$, has the form $(A, C, \ldots, B, \ldots) \mapsto (C, A, \ldots, B, \ldots) \mapsto (C, B, \ldots, A, \ldots)$, where $A < B < C$. These terms cancel in step 2c, so in our algorithm, we cancel this sequence $(A, C, \ldots, B, \ldots) \mapsto (C, A, \ldots, B, \ldots) \mapsto (C, B, \ldots, A, \ldots)$.

4.4. **The final case: σ is a product of three transpositions s_i.** Lastly consider the case where σ equals 3 2 1, that is

$$\sigma = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix}.$$

This Schubert polynomial $\mathfrak{S}_{321}$ is equal to $x_0^2 x_1$. Multiplying $\mathfrak{S}_w$ by this polynomial is equivalent to multiplying by $((x_0 + x_1) - x_0)x_0$ then multiplying by x_0. So a procedure for evaluating the product $\mathfrak{S}_w \mathfrak{S}_{321}$ is to evaluate $\mathfrak{S}_{213}(\mathfrak{S}_{132} - \mathfrak{S}_{213})(\mathfrak{S}_{213}\mathfrak{S}_w) = (\mathfrak{S}_{132} - \mathfrak{S}_{213})\mathfrak{S}_{213}\mathfrak{S}_w$.

It is sufficient to perform the following algorithm.

(1) Apply Monk's formula with $r = 1$ to our given Schubert polynomial $\mathfrak{S}_w$.
(2) Then apply Monk's formula, with $r = 2$ and restricting to the specific case $i = 2$, to each of the Schubert polynomials $\mathfrak{S}_{w'}$ of the output of the previous step.
(3) From this final output discard each Schubert polynomial $\mathfrak{S}_{w''}$ for which the sequence of steps we have just performed, $w \mapsto w' \mapsto w''$, permutes exactly three values according to the form $(A, C, \ldots, B, \ldots) \mapsto (C, A, \ldots, B, \ldots) \mapsto (C, B, \ldots, A, \ldots)$ where $A < B < C$, the first ellipsis represents any sequence of numbers not in the interval $[A, B]$, the second ellipsis represents any sequence of numbers, and each ellipsis sequence may be empty.
(4) Apply Monk's formula, with $r = 1$, to each of the Schubert polynomials of the output of the previous step.

4.5. **n=4.** In the $n = 4$ case we have three different permutations that we constructed multiplication algorithms for.

- $\mathfrak{S}_{1342} = x_0x_1 + x_0x_2 + x_1x_2$ This equals $(x_0 + x_1)(x_0 + x_1 + x_2) - (x_0^2 + x_0x_1 + x_1^2)$. This composition is achieved by computing the product of the Schubert polynomials $\mathfrak{S}_{1324}$ and $\mathfrak{S}_{1243}$ and then subtracting $\mathfrak{S}_{1423}$.
  Proposition. It is sufficient to perform the following algorithm for computing the product $\mathfrak{S}_{1342}$ $\mathfrak{S}_w$.
  (1) Apply Monk's formula, with r=2 to Schubert polynomial $\mathfrak{S}_w$.
  (2) Apply Monk with r=3 to each term $\mathfrak{S}_{w'}$ of the output of the previous step.
  (3) Discard each $\mathfrak{S}_{w''}$ such that one or more of the following conditions are satisfied.
    – An i-value, in Monk's rule, is used more than once.

- The sequence of steps we just performed permutes three values according to the form $(A,\_,C,\dots,B,\dots) \mapsto (C,\_,A,\dots,B,\dots) \mapsto (C,\_,B,\dots,A,\dots)$ or $(\_,A,C,\dots,B,\dots) \mapsto (\_,C,A,\dots,B,\dots) \mapsto (\_,C,B,\dots,A,\dots)$ where the values satisfy the inequality $A < B < C$ and each of the ellipses $(\dots)$ may vary in length.
- The sequence of steps we just performed permutes $\{a,b,x,y\}$ according to the form $(a,x,\dots,b,\dots y) \mapsto (b,x,\dots,a,\dots y) \mapsto (b,y,\dots,a,\dots x)$, or $(a,x,\dots,y,\dots b) \mapsto (a,y,\dots,x,\dots b) \mapsto (b,y,\dots,x,\dots a)$, where the values $\{a,b,x,y\}$ are arbitrary and each of the ellipses may vary in length.

Proof. We seek to evaluate the product ($\mathfrak{S}_{1243}$ $\mathfrak{S}_{1324}$ - $\mathfrak{S}_{1423}$)$\mathfrak{S}_w$. This can be conducted by the following algebraic procedure.

(1) Evaluate the product $\mathfrak{S}_{1243}$ $\mathfrak{S}_{1324}$ $\mathfrak{S}_w$ by first applying Monk's formula with r=2 to $\mathfrak{S}_w$. This accounts for multiplying by $\mathfrak{S}_{1324}$. The output is a sum of Schubert polynomials; let $\mathfrak{S}_{w'}$ denote any arbitrary one of these terms. Then apply Monk with r=3 to $\mathfrak{S}_{w'}$. This accounts for multiplying by $\mathfrak{S}_{1243}$. The output is a sum of Schubert polynomials; let $\mathfrak{S}_{w''}$ denote any arbitrary one of these.

(2) Evaluate the product $\mathfrak{S}_{1423}$ $\mathfrak{S}_w$. To do this perform the following algorithm (which will be proven later in the thesis)
   - Apply Monk's formula with r=2, i=1 or i=2 to Schubert polynomial $\mathfrak{S}_W$.
   - If in step 1 Monk r=2,i=1 was performed, apply Monk with r=2, i=1 to each term $\mathfrak{S}_{W'}$ of the output of the previous step.
   - If in step 1 Monk r=2,i=2 was performed, apply Monk with r=2, i=1 or i=2 to each term $\mathfrak{S}_{W'}$ of the output of the previous step.
   - Discard from 2b each $\mathfrak{S}_{W''}$ such that the sequence of steps we just performed permutes three values according to the form $(A,B,\dots,C,\dots) \mapsto (A,C,\dots,B,\dots) \mapsto (B,C,\dots,A,\dots)$ where the values satisfy the inequality $A < B < C$. It must be noted that capital W is the same as w, but is used to distinguish between the multiplication and subtraction steps.

Our final outputs achieved from performing step 1 include the following cases

a. The case where the sequence $w \mapsto w' \mapsto w''$ starts with Monk's formula being applied to $\mathfrak{S}_w$ with r=2, i=1 with j $\geqslant$ 3. And then to this output Monk's formula is applied with r=3, i=1 with j $\geqslant$ 4.
b. The case where the sequence $w \mapsto w' \mapsto w''$ starts with Monk's formula being applied to $\mathfrak{S}_w$ with r=2, i=1. And then to this output Monk's formula is applied with r=3, i=2.
c. The case where the sequence $w \mapsto w' \mapsto w''$ starts with Monk's formula being applied to $\mathfrak{S}_w$ with r=2, i=1. And then to this output Monk's formula is applied with r=3, i=3.
d. The case where the sequence $w \mapsto w' \mapsto w''$ starts with Monk's formula being applied to $\mathfrak{S}_w$ with r=2, i=2. And then to this output Monk's formula is applied with r=3, i =1.
e. The case where the sequence $w \mapsto w' \mapsto w''$ starts with Monk's formula being applied to $\mathfrak{S}_w$ with r=2, i=2 with j $\geqslant$ 3. And then to this output Monk's formula is applied with r=3, i=2 with j $\geqslant$ 4.
f. The case where the sequence $w \mapsto w' \mapsto w''$ starts with Monk's formula being applied to $\mathfrak{S}_w$ with r=2, i=2. And then to this output Monk's formula is applied with r=3, i =3.

Our final outputs achieved from performing step 2 include the following cases.

i. The case where the sequence $W \mapsto W' \mapsto W''$ starts with Monk's formula being applied to $\mathfrak{S}_W$ with r=2, i=1 with j $\geqslant$ 3. And then to this output Monk's formula is applied with r=2, i=1 with j=3 or j $\geqslant$ 4.

ii. The case where the sequence $W \mapsto W' \mapsto W''$ starts with Monk's formula being applied to $\mathfrak{S}_W$ with r=2, i=2 with j $\geqslant$ 3. And then to this output Monk's formula is applied with r=2, i=1 with j $\geqslant$ 3.

iii. The case where the sequence $W \mapsto W' \mapsto W''$ starts with Monk's formula being applied to $\mathfrak{S}_W$ with r=2, i=2 with j $\geqslant$ 3. And then to this output Monk's formula is applied with r=2, i=2 with j $\geqslant$ 4 or j=3.

The following cases cancel trivially

(1) Step a and the case in step *i* where the sequence $W \mapsto W' \mapsto W''$ starts with Monk's formula being applied to $\mathfrak{S}_W$ with r=2, i=1 with j $\geqslant$ 3. And then to this output Monk's formula is applied with r=2, i=1 with j $\geqslant$ 4.

(2) Step e and the case in step *iii* where the sequence $W \mapsto W' \mapsto W''$ starts with Monk's formula being applied to $\mathfrak{S}_W$ with r=2, i=2 with j $\geqslant$ 3. And then to this output Monk's formula is applied with r=2, i=2 with j $\geqslant$ 4.

This is why in our algorithm, we cancel repeated *i* values.

Some W'' are left over and need to be canceled from w'' so that there are no negative Schubert polynomials among our outputs. The following are patterns that cancel in our algorithm.

(1) In case *i* where we apply Monk's formula with r=2, i=1 with j $\geqslant$ 3 and then apply Monk with r=2, i=1 with j=3 the following permutation occurs from doing these steps: $(A, \_, C, \ldots, B, \ldots) \mapsto (B, \_, C, \ldots, A, \ldots) \mapsto (C, \_, B, \ldots, A, \ldots)$ where the values satisfy the inequality $A < B < C$. These steps are admissible because of Monk's rule. A similar pattern arises in step c. This pattern is $(A, \_, C, \ldots, B, \ldots) \mapsto (C, \_, A, \ldots, B, \ldots) \mapsto (C, \_, B, \ldots, A, \ldots)$ where the values satisfy the inequality $A < B < C$. The ending permutation is the same. Thus, we cancel this sequence in our algorithm.

(2) In case *iii* where we apply Monk's formula with r=2, i=2 with j $\geqslant$ 3 and then apply Monk with r=2, i=2 with j=3 the following permutation occurs from doing these steps: $(\_, A, C, \ldots, B, \ldots) \mapsto (\_, B, C, \ldots, A, \ldots) \mapsto (\_, C, B, \ldots, A, \ldots)$ where the values satisfy the inequality $A < B < C$. These steps are admissible because of Monk's rule. A similar pattern arises in step f. This pattern is $(_, A, C, \ldots, B, \ldots) \mapsto (_, C, A, \ldots, B, \ldots) \mapsto (_, C, B, \ldots, A, \ldots)$where the values satisfy the inequality $A < B < C$. The ending permutation is the same. Thus, we cancel this sequence in our algorithm.

(3) In case *ii* where we apply Monk r=2, i=2 with j $\geqslant$ 3 and then Monk r=2, i=1 with

j $\geqslant$ 3 we achieve the following permutations:

- $(A, C, \ldots B, \ldots, F, \ldots) \mapsto (A, F, \ldots, B, \ldots, C, \ldots) \mapsto (B, F, \ldots, A, \ldots, C, \ldots)$
- $(G, C, \ldots, H, \ldots, F, \ldots) \mapsto (G, F, \ldots, H, \ldots, C, \ldots) \mapsto (H, F, \ldots, G, \ldots, C, \ldots)$
- $(A, C, \ldots, F, \ldots, B, \ldots) \mapsto (A, F, \ldots, C, \ldots, B, \ldots) \mapsto (B, F, \ldots, C, \ldots, A, \ldots)$
- $(D, C, \ldots, F, \ldots, E, \ldots) \mapsto (D, F, \ldots, C, \ldots, E, \ldots) \mapsto (E, F, \ldots, C, \ldots, D, \ldots)$
- $(G, C, \ldots, F, \ldots, H, \ldots) \mapsto (G, F, \ldots, C, \ldots H, \ldots) \mapsto (H, F, \ldots, C, \ldots, G, \ldots)$

Each of these permutations correspond to a permutation in either b or d. So we discard these cases:

- $(A, C, \ldots, B, \ldots, F, \ldots) \mapsto (B, C, \ldots, A, \ldots, F, \ldots) \mapsto (B, F, \ldots, A, \ldots, C, \ldots)$
- $(G, C, \ldots, H, \ldots, F, \ldots) \mapsto (H, C, \ldots, G, \ldots, F, \ldots) \mapsto (H, F, \ldots, G, \ldots, C, \ldots)$
- $(A, C, \ldots, F, \ldots, B, \ldots) \mapsto (A, F, \ldots, C, \ldots, B, \ldots) \mapsto (B, F, \ldots, C, \ldots, A, \ldots)$
- $(D, C, \ldots, F, \ldots, E, \ldots) \mapsto (D, F, \ldots, C, \ldots, E, \ldots) \mapsto (E, F, \ldots, C, \ldots, D, \ldots)$
- $(G, C, \ldots, F, \ldots, H, \ldots) \mapsto (G, F, \ldots, C, \ldots, H, \ldots) \mapsto (H, F, \ldots, C, \ldots, G, \ldots)$

These cases can be written more simply as $(a, x, \ldots, b, \ldots y) \mapsto (b, x, \ldots, a, \ldots y) \mapsto (b, y, \ldots, a, \ldots x)$,
or $(a, x, \ldots, y, \ldots b) \mapsto (a, y, \ldots, x, \ldots b) \mapsto (b, y, \ldots, x, \ldots a)$ where the values $\{a, b, x, y\}$ can be in any numerical ordering. Therefore, we cancel these sequences in our algorithm for $\mathfrak{S}_{1342}$

$\square$

- $\mathfrak{S}_{1423} = x_0^2 + x_0x_1 + x_1^2$. This equals $(x_0 + x_1)(x_0 + x_1) - x_0x_1$. A procedure for evaluating the product $\mathfrak{S}_w\mathfrak{S}_{1423}$ is to evaluate $(\mathfrak{S}_{132}\mathfrak{S}_{132} - \mathfrak{S}_{231})\mathfrak{S}_w$.
Proposition. It is sufficient to perform the following algorithm for computing the product $\mathfrak{S}_{1423}$ $\mathfrak{S}_w$.
  1. Apply Monk's formula with r=2, i=1 or i=2 to Schubert polynomial $\mathfrak{S}_w$.
  2a. If in step 1 Monk r=2, i=1 was performed, apply Monk with r=2, i=1 to each term $\mathfrak{S}_{w'}$ of the output of the previous step.
  2b. If in step 1 Monk r=2, i=2 was performed, apply Monk with r=2, i=1 or i=2 to each term $\mathfrak{S}_{w'}$ of the output of the previous step.
  3. Discard from 2b each $\mathfrak{S}_{w''}$ such that the sequence of steps we just performed permutes three values according to the form $(A, B, \ldots, C, \ldots) \mapsto (A, C, \ldots, B, \ldots) \mapsto (B, C, \ldots, A, \ldots)$ where the values satisfy the inequality $A < B < C$.

Proof. We seek to evaluate the product $(\mathfrak{S}_{132}\mathfrak{S}_{132} - \mathfrak{S}_{231})\mathfrak{S}_w$. This can be conducted by the following algebraic procedure.
  (1) Evaluate the product $\mathfrak{S}_{132}$ $\mathfrak{S}_w$ by applying Monk's formula with r=2 to $\mathfrak{S}_w$. The output is a sum of Schubert polynomials; let $\mathfrak{S}_{w'}$ denote any arbitrary one of these terms. This application is without restriction: the possible values of the ordered pair (i, j) are any that satisfy that $i \in \{1, 2\}$ and $j \geq 3$.
  (2) Evaluate the product $\mathfrak{S}_{132}(\mathfrak{S}_{132}$ $\mathfrak{S}_w)$ by applying Monk with r=2 to each term $\mathfrak{S}_{w'}$ of the output of the previous step. The output is a sum of Schubert polynomials; let $\mathfrak{S}_{w''}$ denote any arbitrary one of these terms. This application is without restriction: the possible values of the ordered pair (i, j) are any that satisfy that $i \in \{1, 2\}$ and $j \geq 3$.
  (3) Evaluate $\mathfrak{S}_{231}$ by performing the following algorithm (found earlier in the thesis)
    (a) Apply Monk's formula with $r = 1$ to our given Schubert polynomial $\mathfrak{S}_w$.
    (b) Then apply Monk's formula, with $r = 2$ and restrict to the specific case $i = 2$, to each of the Schubert polynomials $\mathfrak{S}_{w'}$ of the output of the previous step.
    (c) From this final output, discard each Schubert polynomial $\mathfrak{S}_{w''}$ for which the sequence of steps we have just performed, $w \mapsto w' \mapsto w''$, permutes exactly three values according to the form $(A, C, \ldots, B, \ldots) \mapsto (C, A, \ldots, B, \ldots) \mapsto (C, B, \ldots, A, \ldots)$ where $A < B < C$. The first ellipsis represents any sequence of numbers not in the interval $[A, B]$, the second ellipsis represents any sequence of numbers, and each ellipsis sequence may be empty.
  (4) Subtract: The output of this Step 2 minus the output of this Step 3

Observe the following properties of Step 3. In the case where Step 3a uses the ordered pair $(i, j) = (1, 2)$, the instructions of Steps 3b and 3c imply that in the final output a Schubert polynomial $\mathfrak{S}_{w''}$ will exist (and *not* get discarded) only if w'' and the original w differ only in the location of three entries, which satisfy that $w = (A, B, \ldots, C, \ldots)$ and $w'' = (B, C, \ldots, A, \ldots)$; no other orderings are admissible. And in the case where Step 3a uses an ordered pair (i, j) such that $j \geq 3$, the discarding rule in Step 3c will not apply.

So we observe that in the subtraction in Step 4 there is a set of terms that cancel trivially. That is, for each term that we obtain by applying to the original $\mathfrak{S}_w$, in Step 1 Monk's formula such that (i, j) satisfies $i = 1$ and $j \geq 3$, then in Step 2 Monk's formula such that (i, j) satisfies $i = 2$ and $j \geq 3$, we obtain an identical term by applying the same operations in Steps 3a and 3b.

And in the subtraction in Step 4 there is another set of terms that cancel. That is, for each term that we obtain by applying to the original $\mathfrak{S}_w$:
(1) in Step 3a, Monk's formula such that $(i, j) = (1, 2)$,
(2) then in Step 3b, Monk's formula such that (i, j) satisfies $i = 2$,
(3) and in 3c verifying that this permutation sequence just performed, $w \mapsto w' \mapsto w''$ has the form $(A, B, \ldots, C, \ldots) \mapsto (B, A, \ldots, C, \ldots) \mapsto (B, C, \ldots, A, \ldots)$;

there corresponds an identical term that we obtain by applying
(1) in Step 1, Monk's formula such that (i, j) satisfies $i = 2$ and $j \geq 3$,
(2) then in Step 2, Monk's formula such that (i, j) satisfies $i = 1$ and $j \geq 3$,
(3) where this sequence has form $(A, B, \ldots, C, \ldots) \mapsto (A, C, \ldots, B, \ldots) \mapsto (B, C, \ldots, A, \ldots)$.

This exhausts all of the cases where the subtraction in Step 4 applies. To conclude: if we take our "elementary" algorithm at the start of this proof, and remove the case whose terms cancel trivially (i.e. in Step 1 Monk's formula with $i = 1$, then in Step 2 Monk's formula with $i = 2$), and remove the cases whose terms cancel non trivially (i.e. in Step 1 Monk's formula with $i = 2$, then in Step 2 Monk's formula with $i = 1$, where $w = (A, B, \ldots, C, \ldots)$ and $w'' = (B, C, \ldots, A, \ldots)$), we obtain the algorithm for $\mathfrak{S}_{1423}$.

- $\mathfrak{S}_{2341} = x_0 x_1 x_2$. This equals $((x_0 + x_1 + x_2) - (x_0 + x_1))((x_0 + x_1) - x_0)x_0$. A procedure for evaluating the product $\mathfrak{S}_w\ \mathfrak{S}_{2341}$ is to evaluate $(\mathfrak{S}_{1243} - \mathfrak{S}_{1324})(\mathfrak{S}_{132} - \mathfrak{S}_{213})\mathfrak{S}_{213}\mathfrak{S}_w$.

Proposition. It is sufficient to perform the following algorithm
(1) Apply Monk's formula with $r = 1$ and specifying to $i = 1$ to Schubert polynomial $\mathfrak{S}_w$.
(2) Apply Monk's formula with $r = 2$ and specifying to $i = 2$ to each term $\mathfrak{S}_{w'}$ of the output of the previous step.
(3) Apply Monk's formula with $r = 3$, $i = 3$, to each term $\mathfrak{S}_{w''}$ of the output of the previous step.
(4) Discard each $\mathfrak{S}_{w^{(3)}}$ that is the final term of a sequence $w \mapsto w' \mapsto w'' \mapsto w^{(3)}$ which contains a subsequence $w^\alpha \mapsto w^\beta \mapsto w^\gamma$, possibly nonconsecutive, which has the form $(\_\ , A, C, \ldots, B, \ldots) \mapsto (\_\ , C, A, \ldots, B, \ldots) \mapsto (\_, C, B, \ldots, A, \ldots)$ or $(A, \_, C, \ldots, B, \ldots) \mapsto (C, \_, A, \ldots, B, \ldots) \mapsto (C, \_, B, \ldots, A, \ldots)$, where the values satisfy the inequality $A < B < C$. If there is a value before the first named letter, this value has no restriction. If there is a value in between the first two named letters, this value must not be in the interval $[A, C]$. The first ellipses represents any sequence of numbers not in the interval $[A, B]$, the second ellipses represents any sequence of numbers, and each ellipses sequence may be empty.

Proof. We seek to evaluate the multiplication of an arbitrary Schubert polynomial, $\mathfrak{S}_w$, by a product. This product is $((x_0 + x_1 + x_2) - (x_0 + x_1))((x_0 + x_1) - (x_0))x_0$. This equals $(\mathfrak{S}_{1243} - \mathfrak{S}_{1324})(\mathfrak{S}_{132} - \mathfrak{S}_{213})\mathfrak{S}_{213}\mathfrak{S}_w$. This can be done by the following algebraic procedure.
(1) Evaluate the product $\mathfrak{S}_{213}\ \mathfrak{S}_w$ by applying Monk's formula with $r = 1, i = 1$ to $\mathfrak{S}_w$. The output is a sum of Schubert polynomials; let $\mathfrak{S}_{w'}$ denote any arbitrary one of these terms.
(2) Evaluate $(\mathfrak{S}_{132} - \mathfrak{S}_{213})(\mathfrak{S}_{213}\mathfrak{S}_w)$ by applying the following.

(a) Evaluate $\mathfrak{S}_{213}(\mathfrak{S}_{213}\mathfrak{S}_w)$ by applying Monk's formula with $r = 1$ to each $\mathfrak{S}_{w'}$. We obtain, for each input $\mathfrak{S}_{w'}$, an output that is a sum of Schubert polynomials; let $\mathfrak{S}_{w''}$ denote any arbitrary one of these terms.
(b) Evaluate $\mathfrak{S}_{132}(\mathfrak{S}_{213}\mathfrak{S}_w)$ by applying Monk's formula with $r = 2$ to each $\mathfrak{S}_{w'}$. Again we obtain, for each $\mathfrak{S}_{w'}$, a sum of Schubert polynomials.
(c) Subtract the output of this Step 2a from the output of this Step 2b.

(3) Evaluate $(\mathfrak{S}_{1243} - \mathfrak{S}_{1324})(\mathfrak{S}_{132} - \mathfrak{S}_{213})(\mathfrak{S}_{213}\mathfrak{S}_w)$ by applying the following.
(a) Evaluate $\mathfrak{S}_{1324}(\mathfrak{S}_{132} - \mathfrak{S}_{213})(\mathfrak{S}_{213}\mathfrak{S}_w)$ by applying Monk's formula with $r = 2$ to each $\mathfrak{S}_{w''}$. We obtain, for each input $\mathfrak{S}_{w''}$, an output that is a sum of Schubert polynomials; let $\mathfrak{S}_{w^{(3)}}$ denote any arbitrary one of these terms.
(b) Evaluate $\mathfrak{S}_{1243}(\mathfrak{S}_{132} - \mathfrak{S}_{213})(\mathfrak{S}_{213}\mathfrak{S}_w)$ by applying Monk's formula with $r = 3$ to each $\mathfrak{S}_{w''}$. Again we obtain, for each $\mathfrak{S}_{w''}$, a sum of Schubert polynomials.
(c) Subtract the output of this Step 3a from the output of this Step 3b.

We have a sequence $w \mapsto w' \mapsto w'' \mapsto w^{(3)}$. Let's use this sequence for steps $2a, 3a$. We have a sequence that we will call $W \mapsto W' \mapsto W'' \mapsto W^{(3)}$, which we will use for steps $2b, 3b$ (the ones that we will "subtract"). In this elementary procedure, let us pause at each step to see more clearly the terms that we will subtract. Let us start with step 2. There are two cases:

(1) Case 1: Suppose that in step 2a we have used an ordered pair, call it (i, j) such that $i = 1$ and $j \geq 3$. We have a sequence $w \mapsto w' \mapsto w''$. In step 2b, this same ordered pair is admissible and, beginning with the same input, we obtain an identical sequence, $W \mapsto W' \mapsto W''$. These terms cancel in step 2c. Note: This is why in our algorithm, when we perform Monk's rule with $r = 2$, we restrict to $i = 2$.
(2) Case 2: Suppose that in step 2a, to exhaust all cases, we use the ordered pair (i, j), where $i = 1$ and $j = 2$. For this to be the admissible (i, j) pair in our first step, we must have used the ordered pair (i, j) such that $i = 1$ and $j \geq 3$. The sequence $w \mapsto w' \mapsto w''$ must have the form $(A, C, \dots B, \dots) \mapsto (B, C, \dots, A, \dots) \mapsto (C, B, \dots, A, \dots)$ for $A < B < C$. In step 2b a sequence arises, which begins with the same input and we are assuming cancellation, so it has the same output. The sequence that we obtain, $W \mapsto W' \mapsto W''$, has the form $(A, C, \dots, B, \dots) \mapsto (C, A, \dots, B, \dots) \mapsto (C, B, \dots, A, \dots)$, where $A < B < C$. These terms cancel in step 2c, so in our algorithm, we cancel this subsequence $(A, C, \dots, B, \dots) \mapsto (C, A, \dots, B, \dots) \mapsto (C, B, \dots, A, \dots)$. Note: In this proof, we assume that the contents of what is included in the dots is arbitrary. We will address this in a lemma and subsequent proof, following this proof.

In step 3, there are two cases to consider:

(1) Case 1: Suppose that in step 3a, we use an ordered pair, call it (i, j) such that $i = 1$ and $j \geq 4$ or $i = 2$ and $j \geq 4$. We have a sequence $w \mapsto w' \mapsto w'' \mapsto w^{(3)}$. In step 3b this same ordered pair is admissible and, beginning with the same input, we obtain an identical sequence, $W \mapsto W' \mapsto W'' \mapsto W^{(3)}$. These terms cancel in step 3c. Note: This is why in our algorithm, when we perform Monk's rule with $r = 3$, we restrict to $i = 3$.
(2) Case 2: Suppose that in step 3a, to exhaust all cases, we use the ordered pair (i, j), where $i = 1, j = 3$ or $i = 2, j = 3$. For this to be the admissible (i, j) pair we must have a subsequence, starting with step 2b that used the ordered pair (i, j) such that $i = 1$ and $j \geq 3$. The sequence $w' \mapsto w'' \mapsto w^{(3)}$ must have the following form $(A, \_, C, \dots B, \dots) \mapsto (B, \_, C, \dots, A, \dots) \mapsto (C, \_, B, \dots, A, \dots)$ or $(\_, A, C, \dots B, \dots) \mapsto (\_, B, C, \dots, A, \dots) \mapsto (\_, C, B, \dots, A, \dots)$, where $A <$

$B < C$ and each of the ellipses $(\dots)$ may vary in length. In step 3b a sequence arises, which begins with the an input that has the same starting form and assumes cancellation, $W' \mapsto W'' \mapsto W^{(3)}$. This sequence must have the form $(A,\_,C,\dots,B,\dots) \mapsto (C,\_,A,\dots,B,\dots) \mapsto (C,\_,B,\dots,A,\dots)$, or $(\_,A,C,\dots,B,\dots) \mapsto (\_,C,A,\dots,B,\dots) \mapsto (\_,C,B,\dots,A,\dots)$ where $A < B < C$ and each of the ellipses $(\dots)$ may vary in length. These terms cancel in step 3c, so in our algorithm, we cancel these subsequences $(\_,A,C,\dots,B,\dots) \mapsto (\_,C,A,\dots,B,\dots) \mapsto (\_,C,B,\dots,A,\dots)$ and $(A,\_,C,\dots,B,\dots) \mapsto (C,\_,A,\dots,B,\dots) \mapsto (C,\_,B,\dots,A,\dots)$.

$\square$

5. General Algorithms

In this section we generalize some of the specific algorithms, to S_n.

5.1. The Case Where 1 Goes to the Right. In this subsection, we present a general algorithm to evaluate the Schubert polynomial product $\mathfrak{S}_w \mathfrak{S}_\sigma$ in all cases where the permutation σ has the following form: $\sigma = [2,3,4,5,\dots,k,1]$, for any k ≥ 3. Here is the algorithm.

(1) Apply Monk's formula, with $r = 1$ and specifying to $i = 1$, to Schubert polynomial $\mathfrak{S}_w$.
(2) Apply Monk with $r = 2, i = 2$, to each term $\mathfrak{S}_{w'}$ of the output of the previous step.
(3) Apply Monk with $r = 3, i = 3$, to each term $\mathfrak{S}_{w''}$ of the output of the previous step.
$\dots$ $\dots$
(k-1) Apply Monk with $r = k-1, i = k-1$, to each term $\mathfrak{S}_{w^{(k-2)}}$ of the output of previous step.
(Lastly) Discard each $\mathfrak{S}_{w^{(k-1)}}$ that is the final term of a sequence $w \mapsto w' \mapsto w'' \mapsto \cdots \mapsto w^{(k-1)}$ which contains a subsequence $w^\alpha \mapsto w^\beta \mapsto w^\gamma$, possibly nonconsecutive, which has the form $(\dots,A,\dots,C,\dots,B,\dots) \mapsto (\dots,C,\dots,A,\dots,B,\dots) \mapsto (\dots,C,\dots,B,\dots,A,\dots)$, where the values satify the inequality $A < B < C$ and each of the ellipses $(\dots)$ may vary in length, and in these permutations $w^\alpha, w^\beta, w^\gamma$, the first two named letters are in positions that are to the left of position #k.

Proof. We seek to evaluate the multiplication of an arbitrary Schubert polynomial, $\mathfrak{S}_w$, by a product. This product is $((x_0+x_1+x_2+x_3+\dots+x_{k-2})-(x_0+x_1+x_2+\dots+x_{k-3}))\cdots((x_0+x_1+x_2+x_3)-(x_0+x_1+x_2))((x_0+x_1+x_2)-(x_0+x_1))((x_0+x_1)-(x_0))x_0$. This can be done by the following algebraic procedure.

(1) Evaluate the product $x_0\mathfrak{S}_w$ by applying Monk's formula with $r = 1, i = 1$ to $\mathfrak{S}_w$. The output is a sum of Schubert polynomials; let $\mathfrak{S}_{w'}$ denote any arbitrary one of these terms.
(2) Evaluate $((x_0+x_1)-(x_0))x_0\mathfrak{S}_w)$ by applying the following.
  (a) Evaluate $x_0(x_0\mathfrak{S}_w)$ by applying Monk's formula with $r = 1$ to each $\mathfrak{S}_{w'}$. We obtain, for each input $\mathfrak{S}_{w'}$, an output that is a sum of Schubert polynomials; let $\mathfrak{S}_{w''}$ denote any arbitrary one of these terms.
  (b) Evaluate $(x_0+x_1)(x_0\mathfrak{S}_w)$ by applying Monk's formula with $r = 2$ to each $\mathfrak{S}_{w'}$. Again we obtain, for each $\mathfrak{S}_{w'}$, a sum of Schubert polynomials.
  (c) Subtract the output of this Step 2a from the output of this Step 2b.
(3) Evaluate $((x_0+x_1+x_2)-(x_0+x_1))((x_0+x_1)-(x_0))x_0\mathfrak{S}_w)$ by applying the following.
  (a) Evaluate $(x_0+x_1)((x_0+x_1)-(x_0))x_0\mathfrak{S}_w$ by applying Monk's formula with $r = 2$ to each $\mathfrak{S}_{w''}$. We obtain, for each input $\mathfrak{S}_{w''}$, an output that is a sum of Schubert polynomials; let $\mathfrak{S}_{w^{(3)}}$ denote any arbitrary one of these terms.
  (b) Evaluate $(x_0+x_1+x_2)((x_0+x_1)-(x_0))x_0\mathfrak{S}_w)$ by applying Monk's formula with $r = 3$ to each $\mathfrak{S}_{w''}$. Again we obtain, for each $\mathfrak{S}_{w''}$, a sum of Schubert polynomials.

(c) Subtract the output of this Step 3a from the output of this Step 3b.

(4) Evaluate $((x_0+x_1+x_2+x_3)-(x_0+x_1+x_2))((x_0+x_1+x_2)-(x_0+x_1))((x_0+x_1)-(x_0))x_0\mathfrak{S}_w)$ by applying the following.

(a) Evaluate $(x_0+x_1+x_2)((x_0+x_1+x_2)-(x_0+x_1))((x_0+x_1)-(x_0))x_0\mathfrak{S}_w)$ by applying Monk's formula with $r=3$ to each $\mathfrak{S}_{w^{(3)}}$. We obtain, for each input $\mathfrak{S}_{w^{(3)}}$, an output that is a sum of Schubert polynomials; let $\mathfrak{S}_{w^{(4)}}$ denote any arbitrary one of these terms.

(b) Evaluate $(x_0+x_1+x_2+x_3)((x_0+x_1+x_2)-(x_0+x_1))((x_0+x_1)-(x_0))x_0\mathfrak{S}_w)$ by applying Monk's formula with $r=4$ to each $\mathfrak{S}_{w^{(3)}}$. Again we obtain, for each $\mathfrak{S}_{w^{(3)}}$, a sum of Schubert polynomials.

(c) Subtract the output of this Step 4a from the output of this Step 4b.

$\dots\ \dots$

(k-1) Evaluate $((x_0+x_1+x_2+x_3+\ldots+x_{k-2})-(x_0+x_1+x_2+\ldots+x_{k-3})\cdots((x_0+x_1+x_2+x_3)-(x_0+x_1+x_2))((x_0+x_1+x_2)-(x_0+x_1))((x_0+x_1)-(x_0))x_0\mathfrak{S}_w$ by applying the following.

(a) Evaluate $(x_0+x_1+x_2+\ldots+x_{k-3})\cdots((x_0+x_1+x_2+x_3)-(x_0+x_1+x_2))((x_0+x_1+x_2)-(x_0+x_1))((x_0+x_1)-(x_0))x_0\mathfrak{S}_w$ by applying Monk's formula with $r=k-2$ to each $\mathfrak{S}_{w^{(k-2)}}$. We obtain for each input $\mathfrak{S}_{w^{(k-2)}}$, an output that is a sum of Schubert polynomials; let $\mathfrak{S}_{w^{(k-1)}}$ denote any arbitrary one of these.

(b) Evaluate $(x_0+x_1+x_2+x_3+\ldots+x_{k-2})\cdots((x_0+x_1+x_2+x_3)-(x_0+x_1+x_2))((x_0+x_1+x_2)-(x_0+x_1))((x_0+x_1)-(x_0))x_0\mathfrak{S}_w$ by applying Monk's formula with $r=k-1$ to each $\mathfrak{S}_{w^{(k-2)}}$. Again we obtain, for each $\mathfrak{S}_{w^{(k-2)}}$, a sum of Schubert polynomials.

(c) Subtract that output of this step $k-1$a for the output of this step $k-1$b.

We have a sequence $w \mapsto w' \mapsto w'' \mapsto w^{(3)} \mapsto w^{(4)} \mapsto \ldots \mapsto w^{(k-1)}$. Let's use this sequence for steps $2a, 3a, \ldots$. We have a sequence that we will call $W \mapsto W' \mapsto W'' \mapsto W^{(3)} \mapsto W^{(4)} \mapsto \ldots \mapsto W^{(k-1)}$, which we will use for steps $2b, 3b, \ldots$ (the ones that we will "subtract"). In this elementary procedure, let us pause at each step to see more clearly the terms that we will subtract. Let us start with step 2. There are two cases:

(1) Case 1: Suppose that in step 2a we have used an ordered pair, call it (i, j) such that $i = 1$ and $j \geq 3$. We have a sequence $w \mapsto w' \mapsto w''$. In step 2b, this same ordered pair is admissible and, beginning with the same input, we obtain an identical sequence, $W \mapsto W' \mapsto W''$. These terms cancel in step 2c. Note: This is why in our algorithm, when we perform Monk's rule with $r = 2$, we restrict to $i = 2$.

(2) Case 2: Suppose that in step 2a, to exhaust all cases, we use the ordered pair (i, j), where $i = 1$ and $j = 2$. For this to be the admissible (i, j) pair in our first step, we must have used the ordered pair (i, j) such that $i = 1$ and $j \geq 3$. The sequence $w \mapsto w' \mapsto w''$ must have the form $(A, C, \ldots B, \ldots) \mapsto (B, C, \ldots, A, \ldots) \mapsto (C, B, \ldots, A, \ldots)$ for $A < B < C$. In step 2b a sequence arises, which begins with the same input and we are assuming cancellation, so it has the same output. The sequence that we obtain, $W \mapsto W' \mapsto W''$, has the form $(A, C, \ldots, B, \ldots) \mapsto (C, A, \ldots, B, \ldots) \mapsto (C, B, \ldots, A, \ldots)$, where $A < B < C$. These terms cancel in step 2c, so in our algorithm, we cancel this subsequence $(A, C, \ldots, B, \ldots) \mapsto (C, A, \ldots, B, \ldots) \mapsto (C, B, \ldots, A, \ldots)$. Note: In this proof, we assume that the contents of what is included in the dots is arbitrary. We will address this in a lemma and subsequent proof, following this proof.

In step 3, there are two cases to consider:

(1) Case 1: Suppose that in step 3a, we use an ordered pair, call it (i, j) such that $i = 1$ and $j \geq 4$ or $i = 2$ and $j \geq 4$. We have a sequence $w \mapsto w' \mapsto w'' \mapsto w^{(3)}$. In step 3b this same ordered pair is admissible and, beginning with the same input, we obtain an identical sequence, $W \mapsto W' \mapsto W'' \mapsto W^{(3)}$. These terms cancel in step

3c. Note: This is why in our algorithm, when we perform Monk's rule with $r = 3$, we restrict to $i = 3$.

(2) Case 2: Suppose that in step 3a, to exhaust all cases, we use the ordered pair (i, j), where $i = 1, j = 3$ or $i = 2, j = 3$. For this to be the admissible (i, j) pair we must have a subsequence, starting with step 2b that used the ordered pair (i, j) such that $i = 1$ and $j \geq 3$. The sequence $w' \mapsto w'' \mapsto w^{(3)}$ must have the following form $(A, \_, C, \ldots B, \ldots) \mapsto (B, \_, C, \ldots, A, \ldots) \mapsto (C, \_, B, \ldots, A, \ldots)$ or $(\_, A, C, \ldots B, \ldots) \mapsto (\_, B, C, \ldots, A, \ldots) \mapsto (\_, C, B, \ldots, A, \ldots)$, where $A < B < C$ and each of the ellipses $(\ldots)$ may vary in length. In step 3b a sequence arises, which begins with the an input that has the same starting form and assumes cancellation, $W' \mapsto W'' \mapsto W^{(3)}$. This sequence must have the form $(A, \_, C, \ldots, B, \ldots) \mapsto (C, \_, A, \ldots, B, \ldots) \mapsto (C, \_, B, \ldots, A, \ldots)$, or $(\_, A, C, \ldots, B, \ldots) \mapsto (\_, C, A, \ldots, B, \ldots) \mapsto (\_, C, B, \ldots, A, \ldots)$ where $A < B < C$ and each of the ellipses $(\ldots)$ may vary in length. These terms cancel in step 3c, so in our algorithm, we cancel these subsequences $(\_, A, C, \ldots, B, \ldots) \mapsto (\_, C, A, \ldots, B, \ldots) \mapsto (\_, C, B, \ldots, A, \ldots)$ and $(A, \_, C, \ldots, B, \ldots) \mapsto (C, \_, A, \ldots, B, \ldots) \mapsto (C, \_, B, \ldots, A, \ldots)$.

In step 4, there are 2 cases to consider

(1) Case 1: Suppose that in step 4a we use an ordered pair, call it (i, j) such that $i = 1$ and $j \geq 5$ or $i = 2$ and $j \geq 5$ or $i = 3$ and $j \geq 5$. We have a sequence $w \mapsto w' \mapsto w'' \mapsto w^{(3)} \mapsto w^{(4)}$. In step 4b this same ordered pair is admissible and ,beginning with the same input, we obtain an identical sequence, $W \mapsto W' \mapsto W'' \mapsto W^{(3)} \mapsto W^{(4)}$. These terms cancel in step 4c. Note: This is why in our algorithm, when we perform Monk's rule with $r = 4$, we restrict to $i = 4$.

(2) Case 2: Suppose that in step 4a, to exhaust all cases, we use the ordered pair (i, j), where $i = 1, j = 4$ or $i = 2, j = 4$or $i = 3, j = 4$. For this to be the admissible (i, j) pair we must have a subsequence, starting with step 3b that used the ordered pair (i, j) such that $i = 1$ and $j \geq 4$. The sequence $w'' \mapsto w^{(3)} \mapsto w^{(4)}$ must have the following form $(A, \_, \_, C, \ldots, B, \ldots) \mapsto (B, \_, \_, C, \ldots, A, \ldots) \mapsto (C, \_, \_, B, \ldots, A, \ldots)$ or $(\_, A, \_, C, \ldots, B, \ldots) \mapsto (\_, B, \_, C, \ldots, A, \ldots) \mapsto (\_, C, \_, B, \ldots, A, \ldots)$ or $(\_, \_, A, C, \ldots, B, \ldots) \mapsto (\_, \_, B, C, \ldots, A, \ldots) \mapsto (\_, \_, C, B, \ldots, A, \ldots)$, where $A < B < C$ and each of the ellipses $(\ldots)$ may vary in length. In step 4b a sequences arises which begins with the same starting form and assumes cancellation, $W'' \mapsto W^{(3)} \mapsto W^{(4)}$. This sequence has the form $(A, \_, \_, C, \ldots, B, \ldots) \mapsto (C, \_, \_, A, \ldots, B, \ldots) \mapsto (C, \_, \_, B, \ldots, A, \ldots)$ or $(\_, A, \_, C, \ldots, B, \ldots) \mapsto (\_, C, \_, A, \ldots, B, \ldots) \mapsto (\_, C, \_, B, \ldots, A, \ldots)$ or $(\_, \_, A, C, \ldots, B, \ldots) \mapsto (\_, \_, C, A, \ldots, B, \ldots) \mapsto (\_, \_, C, B, \ldots, A, \ldots)$ where $A < B < C$ and each of the ellipses $(\ldots)$ may vary in length. These terms cancel in step 4c, so in our algorithm, we cancel these subsequences $(A, \_, \_, C, \ldots, B, \ldots) \mapsto (C, \_, \_, A, \ldots, B, \ldots) \mapsto (C, \_, \_, B, \ldots, A, \ldots)$ or $(\_, A, \_, C, \ldots, B, \ldots) \mapsto (\_, C, \_, A, \ldots, B, \ldots) \mapsto (\_, C, \_, B, \ldots, A, \ldots)$ or $(\_, \_, A, C, \ldots, B, \ldots) \mapsto (\_, \_, C, A, \ldots, B, \ldots) \mapsto (\_, \_, C, B, \ldots, A, \ldots)$.

□

$\ldots \ldots$ In step $k - 1$, there are two cases to consider

(1) Case 1: Suppose that in step $(k - 1)$a, we use an ordered pair, call it (i, j) such that $i = 1$ and $j \geq k$ or $i = 2$ and $j \geq k$, up to $i = k - 2$ and $j \geq k$. We have a sequence $w \mapsto w' \mapsto w'' \mapsto w''' \mapsto \ldots \mapsto w^{k-1}$. In step $(k - 1)$b this same ordered pair is admissible and we obtain a sequence $W \mapsto W' \mapsto W'' \mapsto W''' \mapsto \ldots \mapsto W^{k-1}$, where $w^{k-1} = W^{k-1}$. These terms cancel in step $(k - 1)$c. Note: This is why in our algorithm, when we perform Monk's rule with $r = (k - 1)$, we restrict to $i = (k - 1)$.

(2) Case 2: Suppose that in step $(k - 1)$a, we use the ordered pair (i, j), where $i = 1, j = (k - 1)$ or $i = 2, j = (k - 1)$ up to $i = (k - 2), j = (k - 1)$. For this to be the admissible (i, j) pair we must have a subsequence, starting with step $(k - 2)$a that used the ordered pair (i, j) such that $i = 1$ and $j \geq (k - 1)$. The

subsequence $w^{k-3} \mapsto w^{k-2} \mapsto w^{k-1}$ must have the form $(\ldots A, \ldots, C, \ldots, B, \ldots) \mapsto (\ldots, B, \ldots, C, \ldots, A, \ldots) \mapsto (\ldots, C, \ldots, B, \ldots, A, \ldots)$ where $A < B < C$ and each of the ellipses $(\ldots)$ may vary in length and the first two named letters are to the left of position k. In step $(k-1)$b a similar subsequence arises. $W^{k-3} \mapsto W^{k-2} \mapsto W^{k-1}$, where $w^{k-1} = W^{k-1}$, which has the form $(\ldots A, \ldots, C, \ldots, B, \ldots) \mapsto (\ldots, C, \ldots, B, \ldots, A, \ldots) \mapsto (\ldots, C, \ldots, B, \ldots, A, \ldots)$ where $A < B < C$ and each of the ellipses $(\ldots)$ may vary in length and the first two named letters are to the left of position k. These terms cancel in step $(k-1)$c, so in our algorithm, we cancel these subsequences.

Lemma .... The choice of values of entries in the dots in our proof of the case 1 goes to the right do not alter the truth of the proof.

Proof. In this proof, we will examine the dots that are in our cancelable sequences in the proof of our general algorithm for the case of 1 going to the right. Let us first consider, the case of $k = 3$, and then we will move on to $k = 4$ and $k = 5$.

Here is the case, $k = 3$.
In step 2a there arises the sequence, $w \mapsto w' \mapsto w''$, which is $(A, C, \ldots, B, \ldots) \mapsto (B, C, \ldots, A, \ldots) \mapsto (C, B, \ldots, A, \ldots)$. In step 2b, the sequence that cancels with this arises, $W \mapsto W' \mapsto W''$, which is $(A, C, \ldots, B, \ldots) \mapsto (C, A, \ldots, B, \ldots) \mapsto (C, B, \ldots, A, \ldots)$. Look at the first sequence. Look at w, and suppose we have a value in between C and B, call it a. A can swap with B only if $a \notin [A, B]$, according to Monk's rule. In w', we now have $(B, C, \ldots, A, \ldots)$. B needs to swap with C, to obtain w''. Any value will work, according to Monk's rule. Now, move on to the sequence that arises in step 2b. To go from W to W', A needs to swap with C. This is admissible always, according to Monk's rule. Now suppose that in W', there is a value, call it, b, that is between A and B. A needs to swap with B and can only do so if $b \notin [A, B]$. Since $a \notin [A, B]$ and $b \notin [A, B]$, we can suppose that they are the same. Since, no switches occur after the third named letter, any value can be in the dots.

Here is the case $k = 4$.
In step 3a, there arises the sequences, $(\_A, C, \ldots, B, \ldots) \mapsto (\_B, C, \ldots, A, \ldots) \mapsto (\_C, B, \ldots, A, \ldots)$ or $(A, \_C, \ldots, B, \ldots) \mapsto (B, \_C, \ldots, A, \ldots) \mapsto (C, \_B, \ldots, A, \ldots)$. In step 3b, the sequences that cancel arise. They are $(\_A, C, \ldots, B, \ldots) \mapsto (\_C, A, \ldots, B, \ldots) \mapsto (\_C, B, \ldots, A, \ldots)$ or $(A, \_C, \ldots, B, \ldots) \mapsto (C, \_A, \ldots, B, \ldots) \mapsto (C, \_B, \ldots, A, \ldots)$. Let us consider the sequences in step 3a. In these sequences, assume that there is a value, call it x_2 in between C and B in the starting permutation. A can only swap with B, if $x_2 \notin [A, B]$, according to Monk's rule. If there is a value in between the first two named letters in the sequence, call this value, x_1. A can only swap with B in the first step, if $x_1 \notin [A, B]$ and B and C can only swap with each other, in the second step, if $x_1 \notin [B, C]$. Therefore $x_1 \notin [A, C]$.
In step 3b, consider the sequences that arise. If there is a value in between A and C in the starting permutation, call it y_1. For A to swap with C, $y_1 \notin [A, C]$. Now, if there is a value in between C and B in the starting permutation, call this value y_2. For A to swap with B in the second step, $y_2 \notin [A, B]$.
Since $x_1 \notin [A, C]$ and $y_1 \notin [A, C]$, we can suppose that they are the same. Since $x_2 \notin [A, B]$ and $y_2 \notin [A, B]$, we can suppose that they are the same. Note: If there are values before our three named letters $(A, B, \text{or} C)$ or after, these have no interactions with the switches in this sequence. Therefore, any value can be in these spots.

Here is the case $k = 5$.
In step 4a, there arises the sequences $(\_\_A, C, \ldots, B, \ldots) \mapsto (\_\_B, C, \ldots, A, \ldots) \mapsto (\_\_C, B, \ldots, A, \ldots)$ or $(\_A, \_C, \ldots, B, \ldots) \mapsto (\_B, \_C, \ldots, A, \ldots) \mapsto (\_C, \_B, \ldots, A, \ldots)$ or $(A, \_\_C, \ldots, B, \ldots) \mapsto (B, \_\_C, \ldots, A, \ldots) \mapsto (C, \_\_B, \ldots, A, \ldots)$. In step 4b, the sequences, that cancel arise. They are $(\_\_A, C, \ldots, B, \ldots) \mapsto (\_\_C, A, \ldots, B, \ldots) \mapsto (\_\_C, B, \ldots, A, \ldots)$ or $(\_A, \_C, \ldots, B, \ldots) \mapsto$

$(\_C, \_A, \dots, B, \dots) \mapsto (\_C, \_B, \dots, A, \dots)$ or $(A, \_\_C, \dots, B, \dots) \mapsto (C, \_\_A, \dots, B, \dots) \mapsto (C, \_\_B, \dots, A, \dots)$. Let us first consider the sequences in step 4a. If there is a value in between A and C, in our starting permutation, call it z_1. A can only swap with B, in the first step, if $z_1 \notin [A, B]$, according to Monk's rule. In the second step B can only swap with C, if $z_1 \notin [B, C]$, according to Monk's rule. Therefore, $z_1 \notin [A, C]$. Now, if there is a value in between C and B in the starting permutation, call it z_2, A can only swap with B, in the first step, if $z_2 \notin [A, B]$, according to Monk's rule.
Now, let us look at 4b. If there is a value in between A and C in the starting permutation, call it t_1. A can only swap with C, in the first step, if $t_1 \notin [A, C]$, according the Monk's rule. If there is a value in between C and B, in our starting permutation, call it t_2. For A to swap with B in the second step, $t_2 \notin [A, B]$, according to Monk's rule.
Since $z_1 \notin [A, C]$ and $t_1 \notin [A, C]$, we can suppose that they are the same. Since $z_2 \notin [A, B]$ and $t_2 \notin [A, B]$, we can suppose that they are the same. Note: If there are values before our three named letters (A, B, or, C) or after, these have no interactions with the switches in this sequence. Therefore, any value can be in these spots.

As the spaces in between our first two named letters increases throughout the general algorithm, we only introduce more values with the same restrictions. Any value before or after our named letters has no restrictions regardless of the quantity. Any value in between the first two named letters must not be in the interval $[A, C]$, according to Monk's rule. Any value between the second two named letters must not be in the interval $[A, B]$, according to Monk's rule.

□

5.2. **The Case Where 2 Goes to the Right.** In this subsection we present a general algorithm to evaluate the Schubert polynomial product $\mathfrak{S}_w \mathfrak{S}_\sigma$ in all cases where the permutation σ has the following form: $\sigma = [1, 3, 4, 5, \dots, k, 2]$, for any k ≥ 4.
Claim: It is sufficient to perform the following algorithm for $\mathfrak{S}_{[1345\dots k2]} \mathfrak{S}_w$.

(1) Apply Monk's formula, with r=2 to Schubert polynomial $\mathfrak{S}_w$.
(2) Apply Monk with r=3 to each term $\mathfrak{S}_{w'}$ of the output of the previous step.
(3) Apply Monk with r=4 to each term $\mathfrak{S}_{w''}$ of the output of the previous step.
... ...
(k-2) Apply Monk r=k-1 to each term $\mathfrak{S}_{w^{(k-3)}}$ of the output of the previous step.
(Lastly) Discard each $\mathfrak{S}_{w^{(k-2)}}$ that is the final term of a sequence $w \mapsto w' \mapsto \dots \mapsto w^{(k-2)}$ such that one or more of the following conditions are satisfied.
- An i-value, in Monk's rule, is used more than once.
- The sequence contains a subsequence $w^\alpha \mapsto w^\beta \mapsto w^\gamma$, possibly nonconsecutive, with form $(\dots, A, \dots, C, \dots, B, \dots) \mapsto (\dots, C, \dots, A, \dots, B, \dots) \mapsto (\dots, C, \dots, B, \dots, A, \dots)$, where the values satify the inequality $A < B < C$ and each of the ellipses $(\dots)$ may vary in length, and in these permutations $w^\alpha, w^\beta, w^\gamma$, the first two named letters are in positions that are to the left of position $\#k$.
- The sequence contains a subsequence, possibly nonconsecutive, which has the form $(\dots a, \dots x, \dots, b, \dots y) \mapsto (\dots b, \dots x, \dots, a, \dots y) \mapsto (\dots b, \dots y, \dots, a, \dots x)$, or $(\dots a, \dots x, \dots, y, \dots b) \mapsto (\dots a, \dots y, \dots, x, \dots b) \mapsto (\dots b, \dots y, \dots, x, \dots a)$, where the values $\{a, b, x, y\}$ are arbitrary, each of the ellipses may vary in length, and in these permutations the first two named letters are in positions that are to the left of position $\#k - 1$.

Although, at this point in time, there is no formal proof to this conjecture, we have verified it computationally up to the case of $k = 6$. Also, we have investigated the case of any number going to the right; however, this is still a work in progress.

6. References

(1) Assaf, Sami, Nantel Bergeron, and Frank Sottile. *A combinatorial proof that Schubert vs. Schur coefficients are nonnegative*. arXiv preprint arXiv:1405.2603 (2014).

(2) Sami Assaf, *Combinatorial models for Schubert polynomials*, arXiv:1703.00088. (2017)

(3) Assaf, Sami; Searles, Dominic. *Schubert polynomials, slide polynomials, Stanley symmetric functions and quasi-Yamanouchi pipe dreams*. Adv. Math. 306 (2017), 89–122.

(4) Assaf, Sami, and Anne Schilling. *A Demazure crystal construction for Schubert polynomials*. arXiv preprint arXiv:1705.09649 (2017).

(5) Billey, Sara C; Jockusch, William; Stanley, Richard P. *Some combinatorial properties of Schubert polynomials*. J. Algebraic Combin. 2 (1993), no. 4, 345-374.

(6) Nantel Bergeron and Sara Billey, *RC-graphs and Schubert polynomials*, Experiment. Math. 2 (1993), no. 4, 257269.

(7) Bergeron, Nantel; Sottile, Frank. *Schubert polynomials, the Bruhat order, and the geometry of flag manifolds*. Duke Math. J. 95 (1998), no. 2, 373–423.

(8) Fomin, Sergey; Stanley, Richard P. *Schubert polynomials and the nil-Coxeter algebra*. Adv. Math. 103 (1994), no. 2, 196–207.

(9) Fomin, Sergey; Kirillov, Anatol N. *The Yang-Baxter equation, symmetric functions, and Schubert polynomials*. Proceedings of the 5th Conference on Formal Power Series and Algebraic Combinatorics (Florence, 1993). Discrete Math. 153 (1996), no. 1-3, 123–143.

(10) Knutson, Allen; Miller, Ezra. *Gröbner geometry of Schubert polynomials*. Ann. of Math. (2) 161 (2005), no. 3, 1245–1318.

(11) Alain Lascoux and Marcel-Paul Schüzenberger, *Pôlynomes de Schubert*, C. R. Acad. Sci. Paris Sr. I Math. 294 (1982), no. 13, 447450.

(12) Alain Lascoux and Marcel-Paul Schüzenberger, *Schubert polynomials and the Littlewood-Richardson rule*, Lett. Math. Phys. 10 (1985), 111124.

(13) Ian G. Macdonald, *Symmetric functions and Hall polynomials*, second ed., Oxford Mathematical Monographs, The Clarendon Press Oxford University Press, New York, 1995.

(14) Ian G. Macdonald, *Notes on Schubert polynomials*, Laboratoire de combinatoire et dinformatique mathematique (LACIM), Univ. de Quebeca Montreal, Montreal, 1991.

(15) Miller, Ezra. *Mitosis recursion for coefficients of Schubert polynomials*. J. Combin. Theory Ser. A 103 (2003), no. 2, 223–235.

(16) C. Monical, N. Tokcan, and A. Yong. *Newton Polytopes in Algebraic Combinatorics*. arXiv:1703.02583, March 2017.

(17) Monk, D. *The geometry of flag manifolds*. Proc. London Math. Soc. (3) 9 1959 253–286.

(18) Sottile, Frank. *Pieri's formula for flag manifolds and Schubert polynomials*. Ann. Inst. Fourier (Grenoble) 46 (1996), no. 1, 89–110.

(19) Winkel, Rudolf. *On the multiplication of Schubert polynomials*. Adv. in Appl. Math. 20 (1998), no. 1, 73–97.

(20) Winkel, Rudolf. *Diagram rules for the generation of Schubert polynomials*. J. Combin. Theory Ser. A 86 (1999), no. 1, 14–48.