Local Evacuation Shuffing and the Pieri Case

Kelsey Brown

May 15, 2023

Abstract

In this paper we provide a description of the evacuation shuffling algorithm on skew tableaux and a description of the simpler, faster local evacuation shuffling algorithm[1]. We will discuss the simplest type of tableaux to which these algorithms apply, and provide a proof sketch of the rule formed from that case which serves as the foundation for the proof of the local algorithm. We will also briefly discuss a conjecture relating to this same case.

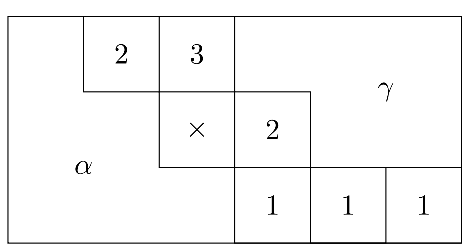
1 Introduction

In *Monodromy and K-Theory of Schubert Curves via Generalized Jeu de Taquin*, Maria Gillespie and Jake Levinson explore the relationship between complex Schubert curves and the orbits of a map ω on skew tableaux. This map ω consists of an operation called shuffling composed with one called evacuation shuffling, where shuffling uses Jeu de Taquin slides to move a box filled with $\times$ through our tableau, and evacuation shuffling is the rectification of a skew tableau, Jeu de Taquin slides on the resulting tableau, and the unrectification of this new tableau[1]. While the geometric meaning of ω is discussed in detail in the paper, our primary focus will be on the following result:

Theorem 1.1. *Local evacuation shuffling is equivalent to evacuation shuffling.*

1.1 Notation

We begin with a $k \times (n-k)$ rectangle (where k and n come from $Gr(k, \mathbb{C}^n)$) that is filled with the skew tableaux formed by four partitions, $\alpha, \square, \beta$, and γ, where $\square$ is a co-corner of α, and β has Littlewood-Richardson content, and α has unique Littlewood-Richardson content. For the purposes of this paper, we will not concern ourselves with γ past this point. We fill $\square$ with $\times$. For example,



is a valid rectangle. Note, applying ω here is really shuffling α past β and back again, so **Theorem 1.1** really provides us with a simpler method of moving tableaux past each other.

In order to appreciate the simplicity of the local evacuation shuffling algorithm, we first consider the traditional way of applying the evacuation shuffling. The procedure is as follows:

Definition 1.2. Evacuation shuffling[1] is performed according to the following steps:

1. Consider $\boxed{\times} \sqcup \beta$, and treat $\boxed{\times}$ as having value 0. Rectify this tableau. Track the order in which each box of the tableau is vacated.
2. Treat $\boxed{\times}$ as empty, and use Jeu de Taquin moves to slide the empty box out, replacing the single vacated spot with $\boxed{\times}$.
3. Treat $\boxed{\times}$ as having value ∞ and unrectify the tableau in the reverse order of the original rectification.

To demonstrate just how tedious this process is, we shall apply it to our previously described (relatively small) β.

Example 1.3. 1. First, we rectify, keeping track of our vacated boxes:

$$\begin{ytableau} 2 & 3 \\ \none & \times & 2 \\ \none & \none & 1 & 1 & 1 \end{ytableau} \longrightarrow \begin{ytableau} 2 & 3 \\ \times & 2 & \mathbf{1} \\ \none & \none & 1 & 1 & 1 \end{ytableau} \longrightarrow \begin{ytableau} 2 & 3 \\ \times & 2 & \mathbf{1} \\ \none & 1 & 1 & 1 & \mathbf{2} \end{ytableau} \longrightarrow \begin{ytableau} 3 & \mathbf{3} \\ 2 & 2 & \mathbf{1} \\ \times & 1 & 1 & 1 & \mathbf{2} \end{ytableau}$$

2. Now we treat $\boxed{\times}$ as empty and perform Jeu de Taquin slides, placing $\boxed{\times}$ in the vacated spot:

$$\begin{ytableau} 3 & \mathbf{3} \\ 2 & 2 & \mathbf{1} \\ \none & 1 & 1 & 1 & \mathbf{2} \end{ytableau} \longrightarrow \begin{ytableau} 3 & \mathbf{3} \\ 2 & 2 & \mathbf{1} \\ 1 & 1 & 1 & \times & \mathbf{2} \end{ytableau}$$

3. Finally, we unrectify our tableau, treating $\boxed{\times}$ as ∞ and working backwards through our tracking numbers:

$$\begin{ytableau} 3 & \mathbf{3} \\ 2 & 2 & \mathbf{1} \\ 1 & 1 & 1 & \times & \mathbf{2} \end{ytableau} \longrightarrow \begin{ytableau} 2 & 3 \\ 1 & 2 & \mathbf{1} \\ \none & 1 & 1 & \times & \mathbf{2} \end{ytableau} \longrightarrow \begin{ytableau} 2 & 3 \\ 1 & 2 & \mathbf{1} \\ \none & \none & 1 & 1 & \times \end{ytableau} \longrightarrow \begin{ytableau} 2 & 3 \\ \none & 1 & 2 \\ \none & \none & 1 & 1 & \times \end{ytableau}$$

Even for a tableau as small as this one, the process took an unpleasant number of steps. For much larger tableaux, the process only gets more obnoxious and time-consuming, hence the need for a faster, local algorithm, such as the one proposed in Gillespie and Levinson's work.

Definition 1.4. Local evacuation shuffling[1] is performed in two phases according to the following steps:

Phase 1: Let $i = 1$.

1. If $\boxed{\times}$ precedes all the i's in reading order, proceed to Phase 2.
2. If $\boxed{\times}$ does not precede all the i's in reading order, swap $\boxed{\times}$ with the nearest i prior to in in reading order. Increment i to $i + 1$ and cycle through Phase 1 again.

Phase 2:

1. If the suffix of $\boxed{\times}$ contains the same number of i's and $i+1$'s, we are done.
2. If the suffix of $\boxed{\times}$ does not contain the same number of i's and $i+1$'s, switch $\boxed{\times}$ with the nearest i after $\boxed{\times}$ in reading order whose suffix does contain the same number of $i's$ and $i+1$'s. Increment i to $i+1$ and cycle through Phase 2 again.

Using the same β as in **Example 1.3**, we apply the local algorithm instead.

Example 1.5. We begin with Phase 1. $\boxed{\times}$ already precedes all the 1's in reading order, so we move directly to Phase 2. There are 3 1's following $\boxed{\times}$ but only 1 2, so we have no choice but to swap $\boxed{\times}$ with the last 1 in reading order, giving us

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

Notice how little time and space this algorithm took compared to the previous example, and it gave us the same result!

The local algorithm is even more clearly demonstrated on a more interesting, complex example, so we will consider one such example from the Gillespie and Levinson's paper.

Example 1.6. Let $\boxed{\times} \sqcup \beta$ be

$$\begin{array}{cccccccc} 2 & 3 & 5 & & & & & \\ & & 4 & 4 & & & & \\ & & & 3 & 3 & 4 & & \\ & & & 1 & 2 & 2 & 3 & \\ & & & \times & 1 & 1 & 2 & 2 \\ & & & & & 1 & 1 & 1 \end{array}$$

We begin with Phase 1 and notice that $\boxed{\times}$ does not precede all of the 1's in reading order, so we swap it with the nearest 1 prior to it, giving us

$$\begin{array}{cccccccc} 2 & 3 & 5 & & & & & \\ & & 4 & 4 & & & & \\ & & & 3 & 3 & 4 & & \\ & & & \times & 2 & 2 & 3 & \\ & & & 1 & 1 & 1 & 2 & 2 \\ & & & & & 1 & 1 & 1 \end{array}$$

Now $\boxed{\times}$ precedes all the 1's, so we increment i from 1 to 2. $\boxed{\times}$ does not precede all the 2's, so we swap it with the nearest 2 prior to it in reading order, resulting in

$\times$	3	5						
		4	4					
			3	3	4			
			2	2	2	3		
			1	1	1	2	2	
						1	1	1

Now $\boxed{\times}$ precedes all the 2's, so we increment i from 2 to 3. $\boxed{\times}$ does precede all the 3's, so we move to Phase 2 with $i = 3$. $\boxed{\times}$ is followed by 4 3's and 3 4's, so we must swap $\boxed{\times}$ with the nearest 3 that will make it so these numbers are tied. We can swap it with the 3 immediately to its right to do so, giving us

3	$\times$	5						
		4	4					
			3	3	4			
			2	2	2	3		
			1	1	1	2	2	
						1	1	1

Now $\boxed{\times}$ is followed by 3 3's and 3 4's, so we increment i from 3 to 4. $\boxed{\times}$ is followed by 3 4's but only 1 5, so we must swap $\boxed{\times}$ with the nearest 4 that will fix this. Swapping it with even the closest 4 will leave $\boxed{\times}$ followed by 2 4's and no 5's, so we have no choice but to swap it with the last possible 4, leaving us with

3	4	5						
		4	4					
			3	3	$\times$			
			2	2	2	3		
			1	1	1	2	2	
						1	1	1

Now $\boxed{\times}$ is followed by 0 4's and 0 5's, so we increment i from 4 to 5. Our tableau contains no 6's, so $\boxed{\times}$ is followed by 0 5's and 0 6's, so we are done.

2 The Pieri Case

This local algorithm is certainly much faster (and computationally simpler) than standard evacuation shuffling, but it remains to be shown that this algorithm agrees with the usual evacuation shuffling algorithm. A large portion of the paper is devoted to this proof, so we will focus only on a small piece of it. Within ω, there is a special case in which β is a one-row partition. This case serves as the base case for the local evacuation shuffling algorithm, and the type of jump the $\boxed{\times}$ makes as described in this Pieri case corresponds to the possible jumps in Phase 1 of the local algorithm.

Theorem 2.1. *Let β be a one-row partition.*

1. *If $\boxed{\times}$ does not precede all the 1's in reading order, evacuation shuffling will exchange $\boxed{\times}$ with the nearest 1 prior to it in reading order.*
2. *If $\boxed{\times}$ precedes all the 1's in reading order, it will perform a special jump where it will exchange places with the last 1 in reading order.*

There are two possible cases of the Pieri case involving the location of $\boxed{\times}$ within the skew tableau, and we consider them before beginning our proof sketch.

Example 2.2. In Case 1, the tableau contains a vertical domino. In this case, ω will preserve the tableau, as evacuation shuffling will move the $\boxed{\times}$ down in the domino, and shuffling will move it back up. For example, we consider evacuation shuffling (in accordance with the Pieri case) performed on the following skew tableau:

$$\begin{array}{cccccccc} 1&1&&&&&& \\ &&1&1&1&&& \\ &&&&\times&1&1&1 \end{array} \longrightarrow \begin{array}{cccccccc} 1&1&&&&&& \\ &&1&1&\times&&& \\ &&&&1&1&1&1 \end{array}$$

When we then apply our shuffling steps, we get that

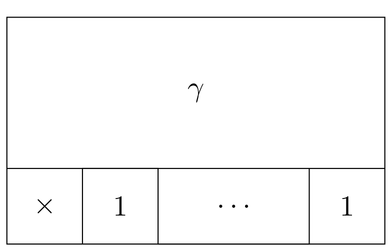
$$\begin{array}{cccccccc} 1&1&&&&&& \\ &&1&1&\times&&& \\ &&&&1&1&1&1 \end{array} \longrightarrow \begin{array}{cccccccc} 1&1&&&&&& \\ &&1&1&1&&& \\ &&&&\times&1&1&1 \end{array}$$

Example 2.3. In Case 2, the tableau contains no vertical domino. In this case, ω will just cycle $\boxed{\times}$ through the rows of our tableau. For example, in the following tableau, if we apply our evacuation shuffling algorithm and then our shuffling algorithm repeatedly, we get:

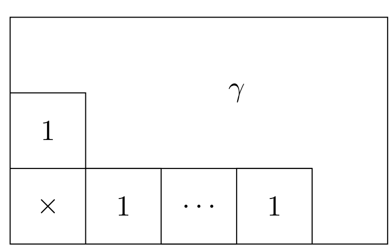
$$\begin{array}{cccc} 1&1&& \\ &&\times&1 \end{array} \longrightarrow \begin{array}{cccc} 1&\times&& \\ &&1&1 \end{array} \longrightarrow \begin{array}{cccc} \times&1&& \\ &&1&1 \end{array} \longrightarrow \begin{array}{cccc} 1&1&& \\ &&1&\times \end{array} \longrightarrow \begin{array}{cccc} 1&1&& \\ &&\times&1 \end{array}$$

The proof for the Pieri case involves inducting on the size of our partition α, and we provide a proof sketch using this method and a bit of proof by example.

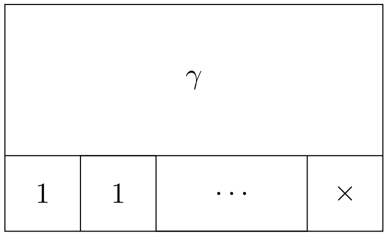
Proof. For an inductive proof, our base case is $\alpha = \emptyset$. This leaves us with two possibilities for $\boxed{\times} \sqcup \beta$:



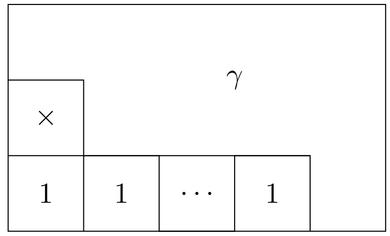
and



In the first case, when we apply our evacuation shuffling algorithm, there are no rectification steps. $\boxed{\times}$ will slide out past the 1s, and there are no unrectifiation steps. In the second case, there are again no rectification steps. $\boxed{\times}$ will slide up to the second row, and there are no unrectification steps. Thus, evacuation shuffling leaves us with

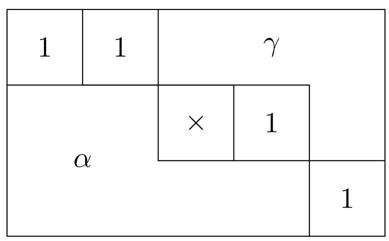


and

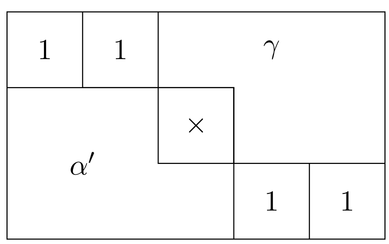


This is precisely what the Pieri case told us would happen, so our base case holds. We now assume the Pieri rule for all partitions α such that $|\alpha| \leq n$. Consider the partion α' where $|\alpha'| = n + 1$. We perform the very first rectification step of the evacuation shuffling algorithm. Doing so will reduce the size of α' by to n as either $\boxed{\times}$ will slide left or a 1 will slide down. But by our induction hypothesis, we know the Pieri rule holds for a partition of size n, so we will swap $\boxed{\times}$ as described in the Pieri case. Undoing the initial rectification step will return our partition to having size $n + 1$, and it the resulting β will be a tableau that was in accordance with the evacuation shuffling as described in the Pieri case.

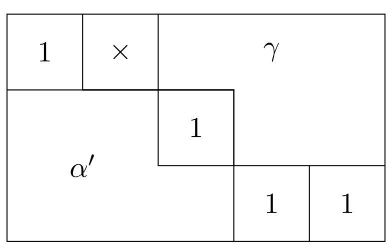
For example, consider the rectangle



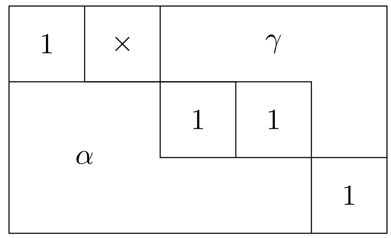
Performing the initial step in the rectification gives us



Following the Pieri rules swaps $\boxed{\times}$ with the 1 immediately to its left, resulting in



Undoing that initial rectification results in a tableau that satisfies the Pieri rule on our initial β :



$\square$

This Pieri rule, as previously mentioned, is an important step in the proof of the local algorithm. In Phase 1, when the $\boxed{\times}$ is swapped past all the i's and i is incremented until $\boxed{\times}$ need be swapped is making use of this Pieri rule, except in the case where the Pieri rule would call for $\boxed{\times}$ to make a special jump. Much of the proof relies on an understanding of the Pieri rule and its usefulness, making understanding its statement and proof a crucial first step in understanding the proof of the local algorithm!

3 Open Problems

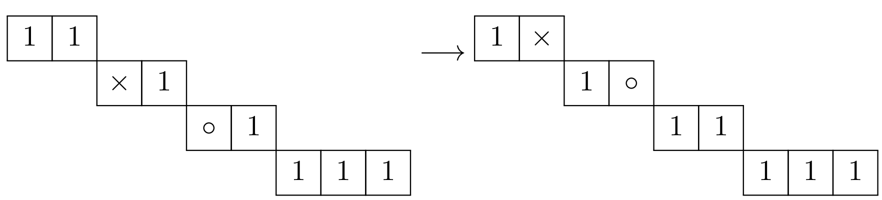
The paper considers partitions $\alpha, \square, \beta$, and γ, so it is only a natural next step to consider if a local algorithm can be found for the evacuation shuffling algorithm on a set of partitions that contains not just one box, but two. It seems likely that such an algorithm can be found, but we now must consider multiple cases, such as whether the two boxes form a horizontal or a vertical domino. Much like in the single box case, it is natural to begin by considering the Pieri case. In the horizontal domino case, our current conjecture handles two cases for this rule.

Conjecture 3.1. *Let β be a one-row partition, and let $\circ > \times$.*

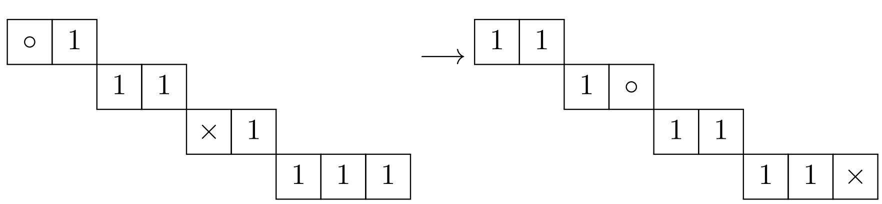
Case 1: *If $\boxed{\circ}$ is not in the top row of $\beta \sqcup \boxed{\times} \sqcup \boxed{\circ}$, then $\boxed{\times}$ and $\boxed{\circ}$ both swap with the 1 immediately to their left in the reading word.*

Case 2: *If $\boxed{\circ}$ is in the top row of $\beta \sqcup \boxed{\times} \sqcup \boxed{\circ}$, then $\boxed{\times}$ performs a special jump to the last position of the tableau, and $\boxed{\circ}$ swaps with the 1 that was immediately to the left of the $\boxed{\times}$ in the reading word.*

Example 3.2. For example, the following tableau falls under Case 1:



Example 3.3. We can also consider an example where our tableau falls under Case 2:



It is likely that the proof of this conjecture is similar to the proof of the Pieri case when we only have $\boxed{\times}$ to contend with, but there will be more cases to contend with when the first rectification step is performed.

References

[1] M. Gillespie, J. Levinson *Monodromy and K-Theory of Schubert Curves via Generalized Jeu de Taquin*, (2016).