Major Index for 01-Fillings of Moon Polyominoes

William Y.C. Chen[a,1], Svetlana Poznanović[b], Catherine H. Yan[a,b,2] and
Arthur L.B. Yang[a,1]

[a]Center for Combinatorics, LPMC-TJKLC
Nankai University, Tianjin 300071, P. R. China

[b]Department of Mathematics
Texas A&M University, College Station, TX 77843

Abstract

We propose a major index statistic on 01-fillings of moon polyominoes which, when specialized to certain shapes, reduces to the major index for permutations and set partitions. We consider the set $\mathbf{F}(\mathcal{M}, \mathbf{s}; A)$ of all 01-fillings of a moon polyomino $\mathcal{M}$ with given column sum $\mathbf{s}$ whose empty rows are A, and prove that this major index has the same distribution as the number of north-east chains, which are the natural extension of inversions (resp. crossings) for permutations (resp. set partitions). Hence our result generalizes the classical equidistribution results for the permutation statistics inv and maj. Two proofs are presented. The first is an algebraic one using generating functions, and the second is a bijection on 01-fillings of moon polyominoes in the spirit of Foata's second fundamental transformation on words and permutations.

1   Introduction

Given a multiset S of n positive integers, a word on S is a sequence $w = w_1 w_2 \ldots w_n$ that reorders the elements in S. When $S = [n] := \{1, \ldots, n\}$ the word is a permutation. A pair (w_i, w_j) is called an *inversion* of w if $i < j$ and $w_i > w_j$. One well-known statistic on words and permutations is $\text{inv}(w)$, defined as the number of inversions of w. The *descent set* and *descent statistic* of a word w are defined as

$$\text{Des}(w) = \{i : 1 \le i \le n-1, w_i > w_{i+1}\}, \qquad \text{des}(w) = \#\text{Des}(w).$$

In [15] MacMahon defined the *major index* statistic for a word w as

$$\text{maj}(w) = \sum_{i \in \text{Des}(w)} i,$$

and showed the remarkable result that its distribution over all words on S is equal to that of the inversion number over the same set. Precisely, for the set W_S of all words on S,

$$\sum_{w \in W_S} q^{\mathrm{maj}(w)} = \sum_{w \in W_S} q^{\mathrm{inv}(w)}. \tag{1.1}$$

The proof of (1.1) given by MacMahon relied on combinatorial analysis, while an elegant bijection $\Phi : W_S \to W_S$ such that $\mathrm{maj}(w) = \mathrm{inv}(\Phi(w))$ was constructed later by Foata [6].

A *set partition* of $[n]$ is a family of nonempty sets $B_1, \ldots, B_k$ which are pairwise disjoint and whose union is $[n]$. If all the blocks B_i have at most two elements, the set partition is called a *matching*. If the size of every block is exactly two, the matching is called a *perfect matching*. A set partition π can be represented as a simple graph with vertex set $[n]$ drawn on a horizontal line, where two vertices are joined by an arc if and only if they represent consecutive elements in the same block of π under the numerical order. Two statistics, the numbers of crossings and nestings, arise naturally from such graphical representations, and have been the central topic in many recent research articles, e.g., [1], [12], [13], [17], [20], [16], [18], [5], [22], to list a few. A *crossing* (resp. *nesting*) of a set partition is a pair of arcs (i_1, j_1), (i_2, j_2) such that $i_1 < i_2 < j_1 < j_2$ (resp. $i_1 < i_2 < j_2 < j_1$). As explained in [2, Section 2], a crossing can be viewed as a natural generalization of an inversion in a permutation. Hence it is natural to ask what would be the major index for matchings (set partitions) that extends MacMahon's equidistribution results (1.1). A definition of such a statistic, denoted by pmaj, was proposed by Chen et al. in [2]. For completeness, we present the definition for perfect matchings here. The general case for set partitions is very similar. Label the arcs of a perfect matching M of $[2n]$ by $1, \ldots, n$ in decreasing order with respect to their left endpoints from left to right. Suppose $r_1, \ldots, r_n$ are the right endpoints of arcs from right to left. Then to each right endpoint r_i of an arc associate a word $w^{(i)}$ on $[n]$ where the words $w^{(i)}$ are defined backwards recursively: let $w^{(1)} = a$, where a is the label of the arc that ends with r_1. In general, after defining $w^{(i)}$, assume that the left endpoints of the arcs labeled $a_1, \ldots, a_t$ lie between r_i and r_{i+1}. Then $w^{(i+1)}$ is obtained from $w^{(i)}$ by deleting entries $a_1, \ldots, a_t$ and adding the entry b at the very beginning, where b is the label of the arc that ends with r_{i+1}. Then

$$\mathrm{pmaj}(M) := \sum_{i=1}^{n} \mathrm{des}(w^{(i)}).$$

Chen et al. [2] computed the generating function for *pmaj* and showed that it is equally distributed with the number of crossings over all matchings with fixed left and right endpoints. A similar result holds for set partitions.

The aim of this paper is to extend the major index to certain 01-fillings of moon polyominoes which include words and set partitions. In Section 2 we introduce the necessary notations, and describe the definition of the major index for 01-fillings of moon polyominoes. We explain in Section 3 how the classical definition of major index on words and set partitions can be obtained by considering moon polyominoes of special shapes. In Section 4 we show that the maj statistic is equally distributed as ne, the number of north-east chains, by computing the corresponding generating functions. In the fillings of special shapes, ne corresponds to the number of inversions for words and crossings for set partitions. Therefore, our main result, Theorem 4.1, leads to a generalization of (1.1) and the analogous result for set partitions. In Section 5, we present a bijective proof for the equidistribution of maj and ne, which consists of three maps. The first is

from the fillings of left-aligned stack polyominoes to itself that sends maj to ne, constructed in the spirit of Foata's second fundamental transformation. The other two are maps that transform a moon polyomino to a left-aligned stack polyomino with the same set of columns, while preserving the statistics maj and ne, respectively. Composing these three maps yields the desired bijection.

2   Definition of major index for fillings of moon polyominoes

A *polyomino* is a finite subset of $\mathbb{Z}^2$, where we represent every element of $\mathbb{Z}^2$ by a square cell. The polyomino is *convex* if its intersection with any column or row is connected. It is *intersection-free* if every two columns are comparable, i.e., the row-coordinates of one column form a subset of those of the other column. Equivalently, it is intersection-free if every two rows are comparable. A *moon polyomino* is a convex intersection-free polyomino. If the rows (resp. columns) of the moon polyomino are left-aligned (resp. top-aligned), we will call it a *left-aligned stack polyomino* (resp. *top-aligned stack polyomino*). A *Ferrers diagram* is a left-aligned and top-aligned stack polyomino. See Figure 1 for an illustration. The term 'moon polyomino' was first used by Jonsson in [10] where he used fillings of such polyominoes to study generalized triangulations.
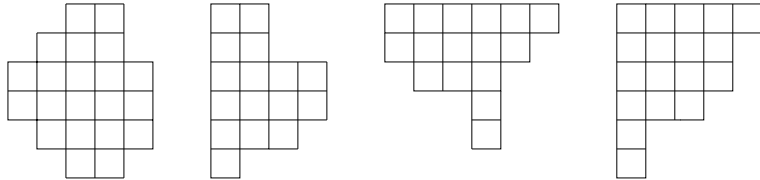

Figure 1: A moon polyomino, a left-aligned and a top-aligned stack polyomino, and a Ferrers diagram.

We are concerned with 01-fillings of moon polyominoes with restricted row sums. That is, given a moon polyomino $\mathcal{M}$, we assign a 0 or a 1 to each cell of $\mathcal{M}$ so that there is at most one 1 in each row. Throughout this paper we will simply use the term *filling* to denote such 01-filling. We say a cell is empty if it is assigned a 0, and it is a 1-cell otherwise. Given a moon polyomino $\mathcal{M}$ with m columns, we label them $c_1, \ldots, c_m$ from left to right. Let $\mathbf{s} = (s_1, \ldots, s_m)$ be an m-tuple of nonnegative integers and A be a subset of rows of $\mathcal{M}$. We denote by $\mathbf{F}(\mathcal{M}, \mathbf{s}; A)$ the set of fillings M of $\mathcal{M}$ such that the empty rows of M are exactly those in A and the column c_i has exactly s_i many 1's, $1 \leq i \leq m$. A *north-east (NE) chain* in a filling M of $\mathcal{M}$ is a set of two 1-cells such that one of them is strictly above and to the right of the other and the smallest rectangle containing them is contained in $\mathcal{M}$. The number of NE chains of M will be denoted $\mathrm{ne}(M)$. See Figure 2.

The set $\mathbf{F}(\mathcal{M}, \mathbf{s}; A)$ was first studied by Kasraoui [11], who showed that the numbers of northeast and south-east chains are equally distributed. The generating functions for these statistics are also presented in [11, Theorem 4.4].

To define the major index for fillings of moon polyominoes, we first state it for rectangular shapes, which is essentially the classical definition of major index for words, (c.f. Section 3.1).

Let R be a filling of a rectangle whose n *nonempty* rows $r_1, \ldots, r_n$ are numbered from top to bottom. We define the *descent statistic* for R as

$$\mathrm{des}(R) = |\{i \mid \text{ the 1-cells in rows } r_i \text{ and } r_{i+1} \text{ form an NE chain}\}|.$$
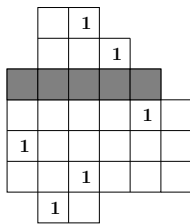
3

Figure 2: Filling M of a moon polyomino $\mathcal{M}$ for $A = \{3\}$ and $\mathbf{s} = (1, 1, 2, 1, 1, 0)$ with $\mathrm{ne}(M) = 5$.

That is, descents of a rectangular filling are NE chains in consecutive nonempty rows.

For each nonempty row r_i of R, let $R(r_i)$ denote the rectangle that contains the row r_i and all rows in R above the row r_i.

Definition 2.1. The major index for the rectangular filling R is defined to be

$$\mathrm{maj}(R) = \sum_{i=1}^{n} \mathrm{des}(R(r_i)). \tag{2.1}$$

Clearly, the empty rows of R do not play any role in $\mathrm{maj}(R)$. That is, if we delete them, the major index of the resulting filling remains the same.

Definition 2.2. Let M be a filling of a moon polyomino $\mathcal{M}$. Let $\mathcal{R}_1, \ldots, \mathcal{R}_r$ be the list of all the maximal rectangles contained in $\mathcal{M}$ ordered increasingly by height, and denote by R_i the filling M restricted on the rectangle $\mathcal{R}_i$. Then the major index of M is defined to be

$$\mathrm{maj}(M) = \sum_{i=1}^{r} \mathrm{maj}(R_i) - \sum_{i=1}^{r-1} \mathrm{maj}(R_i \cap R_{i+1}), \tag{2.2}$$

where $\mathrm{maj}(R_i)$ and $\mathrm{maj}(R_i \cap R_{i+1})$ are defined by (2.1).

In particular, $\mathrm{maj}(M) = \mathrm{maj}(R_1)$ when M is a rectangular shape. It is also clear that $\mathrm{maj}(M)$ is always nonnegative since $\mathrm{maj}(R_{i+1}) \geq \mathrm{maj}(R_i \cap R_{i+1})$, $i = 1, \ldots, r-1$.

Example 2.3. *Consider the filling from Figure 2. It has four maximal rectangles and* $\mathrm{maj}(M) = (2 + 2 + 1 + 1) - (2 + 0 + 0) = 4$.
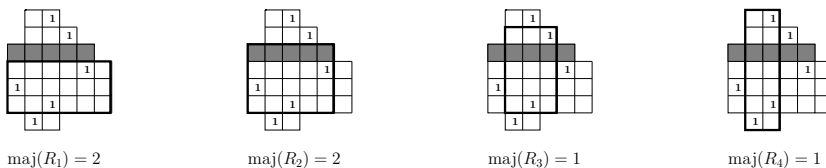

Figure 3: Calculation of $\mathrm{maj}(M)$ for a moon polyomino using Definition 2.2.

4

The major index can be equivalently defined in a slightly more complicated way. However, this way is useful in proofs, especially when one uses induction on the number of columns of the filling, (c.f. Theorem 4.1). We state this equivalent definition next. First we need some notation. Let $Left(\mathcal{M})$ be the set of columns of $\mathcal{M}$ which are to left of the columns of maximal length and let $Right(\mathcal{M})$ consist of the remaining columns of $\mathcal{M}$.

We order the columns of $\mathcal{M}$, $c_1, \ldots, c_m$, by a total order $\prec$ as follows: $c_i \prec c_j$ if and only if

- $|c_i| < |c_j|$ or

- $|c_i| = |c_j|$, $c_i \in Left(\mathcal{M})$ and $c_j \in Right(\mathcal{M})$, or

- $|c_i| = |c_j|$, $c_i, c_j \in Left(\mathcal{M})$ and c_i is to the left of c_j, or

- $|c_i| = |c_j|$, $c_i, c_j \in Right(\mathcal{M})$ and c_i is to the right of c_j,

where $|c|$ denotes the length of the column c. A similar ordering of rows was used in [11].

For every column $c_i \in Left(\mathcal{M})$ define the rectangle $\mathcal{M}(c_i)$ to be the largest rectangle that contains c_i as the leftmost column. For $c_i \in Right(\mathcal{M})$, the rectangle $\mathcal{M}(c_i)$ is taken to be the largest rectangle that contains c_i as the rightmost column and does not contain any columns from $Left(\mathcal{M})$ of same length as c_i.

Definition 2.2'. Let $c_{i_1} \prec c_{i_2} \prec \cdots \prec c_{i_m}$ be the ordering of the columns of $\mathcal{M}$ and let M_j be the restriction of M on the rectangle $\mathcal{M}(c_{i_j})$, $1 \leq j \leq m$. Then

$$\mathrm{maj}(M) = \sum_{j=1}^{m} \mathrm{maj}(M_j) - \sum_{j=1}^{m-1} \mathrm{maj}(M_j \cap M_{j+1}), \tag{2.3}$$

where $\mathrm{maj}(M_j)$ and $\mathrm{maj}(M_j \cap M_{j+1})$ are defined by (2.1).

Example 2.4. *Consider the filling M from Figure 2. The order $\prec$ on the columns of $\mathcal{M}$ is $c_6 \prec c_1 \prec c_5 \prec c_4 \prec c_3 \prec c_2$. So, $\mathcal{M}_1 = \mathcal{M}(c_6)$, $\mathcal{M}_2 = \mathcal{M}(c_1)$, $\mathcal{M}_3 = \mathcal{M}(c_5)$, $\mathcal{M}_4 = \mathcal{M}(c_4)$, $\mathcal{M}_5 = \mathcal{M}(c_3)$, and $\mathcal{M}_6 = \mathcal{M}(c_2)$, as illustrated in Figure 4. By Definition 2.2', $\mathrm{maj}(M) = (2 + 2 + 1 + 1 + 1 + 0) - (2 + 1 + 0 + 0 + 0) = 4$.*

Proposition 2.5. *Definitions 2.2 and 2.2' are equivalent.*

Proof. Some of the rectangular fillings M_j in Definition 2.2' may contain M_{j+1}, in which case $\mathrm{maj}(M_{j+1}) - \mathrm{maj}(M_j \cap M_{j+1}) = 0$ and formula (2.3) can be simplified. More precisely, there are uniquely determined indices $1 = j_1 < j_2 < \cdots < j_r \leq m$ such that $M_1 = M_{j_1} \supseteq M_2 \supseteq \cdots \supseteq M_{j_2-1} \not\supseteq M_{j_2} \supseteq \cdots \supseteq M_{j_r-1} \not\supseteq M_{j_r} \supseteq \cdots \supseteq M_m$. That is, if the rectangles M_i are ordered by containment, then $M_{j_1}, M_{j_2}, \ldots, M_{j_r}$ are the maximal elements. Following the notation in Definition 2.2, the filling M_{j_k} is the filling R_k of the k-th maximal rectangle contained in $\mathcal{M}$. Then, after cancellation of some terms, the right-hand side of (2.3) becomes the right-hand side of (2.2). For instance, in the previous example, the rectangles M_3 and M_6 are not maximal and therefore $\mathrm{maj}(M_3) - \mathrm{maj}(M_2 \cap M_3) = 0$ and $\mathrm{maj}(M_6) - \mathrm{maj}(M_5 \cap M_6) = 0$. $\qquad\square$
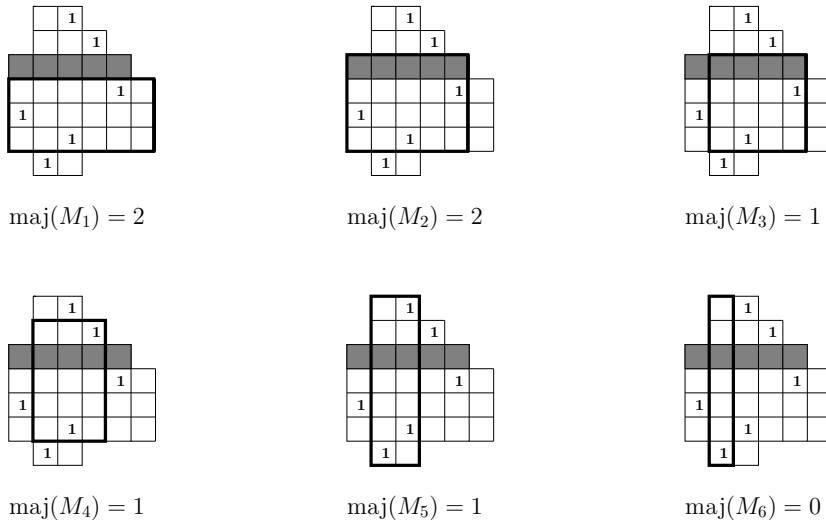
Figure 4: Calculation of $\mathrm{maj}(M)$ for a moon polyomino using Definition 2.2'.

3 $\mathrm{maj}(M)$ for special shapes $\mathcal{M}$

It is well-known that permutations and set-partitions can be represented as fillings of Ferrers diagrams (e.g. [14]). Here we will describe such presentations to show how to get the classical major index for words and set partitions from Definition 2.2.

3.1 When $\mathcal{M}$ is a rectangle: words and permutations

For any rectangle, label the rows from top to bottom, and columns from left to right. Fillings of rectangles are in bijection with words. More precisely, let $w = w_1 \ldots w_n$ be a word with letters in the set $[m]$. The word w can be represented as a filling M of an $n \times m$ rectangle $\mathcal{M}$ in which the cell in row $n - i + 1$ and column $m - j + 1$ is assigned the integer 1 if $w_i = j$, and is empty otherwise. Conversely, each filling of the rectangle $\mathcal{M}$ corresponds to a word w. It is easy to check that $\mathrm{ne}(M) = \mathrm{inv}(w)$, $\mathrm{des}(M) = \mathrm{des}(w)$, and $\mathrm{maj}(M) = \mathrm{maj}(w)$. The latter follows from the fact that $\mathrm{maj}(w) = \sum_{i=1}^{n} \mathrm{des}(w_i w_{i+1} \ldots w_n)$.

3.2 When $\mathcal{M}$ is a Ferrers diagram: matchings and set partitions

As explained in [4], general fillings of Ferrers diagrams correspond to multigraphs. Here we briefly describe the correspondence when restricted to 01-fillings with row sum at most 1. The Ferrers diagram is bounded by a vertical line from the left, a horizontal line from above, and a path consisting of east and north steps. Following this path starting from the bottom left end, we label the steps of the path by $1, 2, \ldots, n$. This gives a labeling of the columns and rows of the diagram. Draw n vertices on a horizontal line and draw an edge connecting vertices i and j if and only if there is a 1 in the cell of the column labeled i and the row labeled j (see Figure 5). The resulting graph has no loops or multiple edges and every vertex is a right endpoint of at most one edge. The NE chains correspond to crossings of edges in the graph. If the column sums are at most one,

the graph is a matching. If, moreover, in the labeling of the steps we allow to have a vertical step followed by a horizontal one with a same label, then the corresponding graph will have vertices which are both right and left endpoints of edges. Such graphs represent set partitions. In either case, the major index of the filling is equal to the major index of the corresponding matching or set partition as defined in [2]. Restricting to the triangular Ferrers diagram, the 01-fillings considered in this paper also include *linked partitions*, a combinatorial structure that was studied in [3, 11].
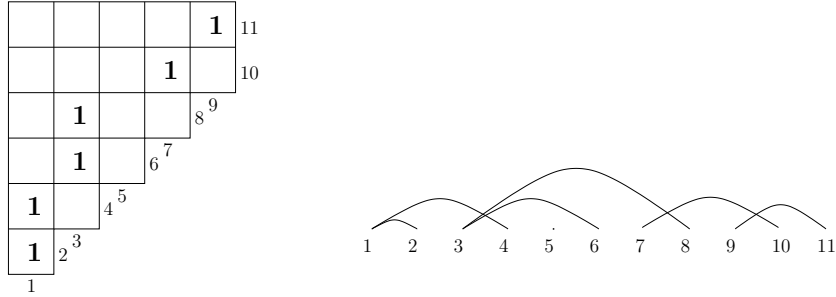

Figure 5: A filling of a Ferrers diagram and the corresponding graph.

3.3 When $\mathcal{M}$ is a top-aligned stack polyomino

In the case when $\mathcal{M}$ is a top-aligned stack polyomino, $\mathrm{maj}(M)$ has a simpler form in terms of the des statistic, just as in words and permutations. Explicitly, if the nonempty rows of M are $r_1, \ldots, r_n$, denote by $M(r_i)$ the filling M restricted to the largest rectangle contained in $\mathcal{M}$ whose bottom row is r_i.

Proposition 3.1. *Let M be a filling of a top-aligned stack polyomino and $M(r_i)$ be as defined above. Then*

$$\mathrm{maj}(M) = \sum_{i=1}^{n} \mathrm{des}(M(r_i)). \tag{3.1}$$

Proof. Let $r_{k+1}, \ldots, r_n$ be all the nonempty rows of M that are of same length as the last row r_n. Denote by $M' = M \backslash \{r_{k+1}, \ldots, r_n\}$ and by R the filling of the largest rectangle of $\mathcal{M}$ containing r_n. We proceed by induction on the number n of rows of M. The claim is trivial when M has only one row or is a rectangle. Otherwise,

$$\mathrm{maj}(M) = \mathrm{maj}(M') + \mathrm{maj}(R) - \mathrm{maj}(M' \cap R) \qquad \text{(Definition 2.2)}$$

$$= \sum_{i=1}^{k} \mathrm{des}(M(r_i)) + \sum_{i=k+1}^{n} \mathrm{des}(M(r_i)), \qquad \text{(Induction hypothesis and Definition 2.1)}$$

which completes the proof. $\qquad \square$

Example 3.2. *Consider the filling M of a top-aligned stack polyomino in Figure 6 with $A = \{3\}$ and $\boldsymbol{s} = (0, 1, 2, 1, 1)$. M has five nonempty rows and, by Proposition 3.1, $\mathrm{maj}(M) = \sum_{i=1}^{5} \mathrm{des}(M(r_i)) = 0 + 1 + 0 + 1 + 1 = 3$.*

Proposition 3.1 will be used in the proof of Lemma 5.5.

$\mathrm{des}(M(r_1)) = 0 \quad \mathrm{des}(M(r_2)) = 1 \quad \mathrm{des}(M(r_3)) = 0 \quad \mathrm{des}(M(r_4)) = 1 \quad \mathrm{des}(M(r_5)) = 1$
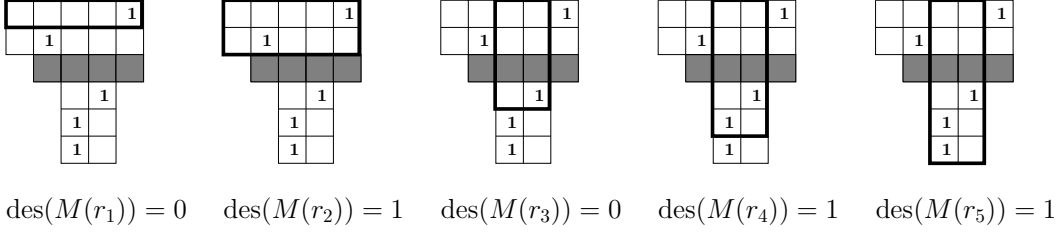
Figure 6: Calculation of $\mathrm{maj}(M)$ using Proposition 3.1.

4 The generating function for the major index

In this section we state and prove the main theorem which gives the generating function for maj over $\mathbf{F}(\mathcal{M}, \mathbf{s}; A)$, the set of all fillings of the moon polyomino $\mathcal{M}$ with column sums given by the integer sequence $\mathbf{s}$ and empty rows given by A. Let $|c_1|, \ldots, |c_m|$ be the column lengths of M and let a_i the number of rows in A that intersect the column c_i. Suppose that, under the ordering defined in Section 2, $c_{i_1} \prec c_{i_2} \prec \cdots \prec c_{i_m}$. Then for $j = 1, \ldots, m$, define

$$h_{i_j} = |c_{i_j}| - a_{i_j} - (s_{i_1} + s_{i_2} + \cdots + s_{i_{j-1}}) \tag{4.1}$$

The numbers h_i have the following meaning: if one fills in the columns of M from smallest to largest according to the order $\prec$, then h_{i_j} is the number of available cells in the j-th column to be filled.

Theorem 4.1. *For a moon polyomino $\mathcal{M}$ with m columns,*

$$\sum_{M \in \mathbf{F}(\mathcal{M}, \mathbf{s}; A)} q^{\mathrm{maj}(M)} = \prod_{i=1}^{m} \begin{bmatrix} h_i \\ s_i \end{bmatrix}_q \tag{4.2}$$

where h_i is defined by (4.1).

We postpone the proof of Theorem 4.1 until the end of this section.

Example 4.2. *Suppose $\mathcal{M}$ is the first moon polyomino in Figure 1, $A = \{5\}$, and $\mathbf{s} = (1, 0, 2, 1, 1)$. The $\prec$ order on the columns of $\mathcal{M}$ is: $c_1 \prec c_5 \prec c_2 \prec c_4 \prec c_3$. The fifth row intersects all columns except c_1, so, $a_1 = 0$ and $a_2 = a_3 = a_4 = a_5 = 1$. Therefore,*

$$
\begin{aligned}
h_1 &= h_{i_1} = |c_1| - a_1 = 2 - 0 = 2 \\
h_5 &= h_{i_2} = |c_5| - a_5 - s_1 = 3 - 1 - 1 = 1 \\
h_2 &= h_{i_3} = |c_2| - a_2 - (s_1 + s_5) = 4 - 1 - (1 + 1) = 1 \\
h_4 &= h_{i_4} = |c_4| - a_4 - (s_1 + s_5 + s_2) = 6 - 1 - (1 + 1 + 0) = 3 \\
h_3 &= h_{i_5} = |c_3| - a_3 - (s_1 + s_5 + s_2 + s_4) = 6 - 1 - (1 + 1 + 0 + 1) = 2
\end{aligned}
$$

By Theorem 4.1, $\sum_{M \in \mathbf{F}(\mathcal{M}, \mathbf{s}; A)} q^{\mathrm{maj}(M)} = \prod_{i=1}^{m} \begin{bmatrix} h_i \\ s_i \end{bmatrix}_q = \begin{bmatrix} 2 \\ 1 \end{bmatrix}_q \begin{bmatrix} 1 \\ 0 \end{bmatrix}_q \begin{bmatrix} 2 \\ 2 \end{bmatrix}_q \begin{bmatrix} 3 \\ 1 \end{bmatrix}_q \begin{bmatrix} 1 \\ 1 \end{bmatrix}_q = (1 + q)(1 + q + q^2) = 1 + 2q + 2q^2 + q^3$. The fillings in $\mathbf{F}(\mathcal{M}, \mathbf{s}; A)$ are listed in Figure 7.
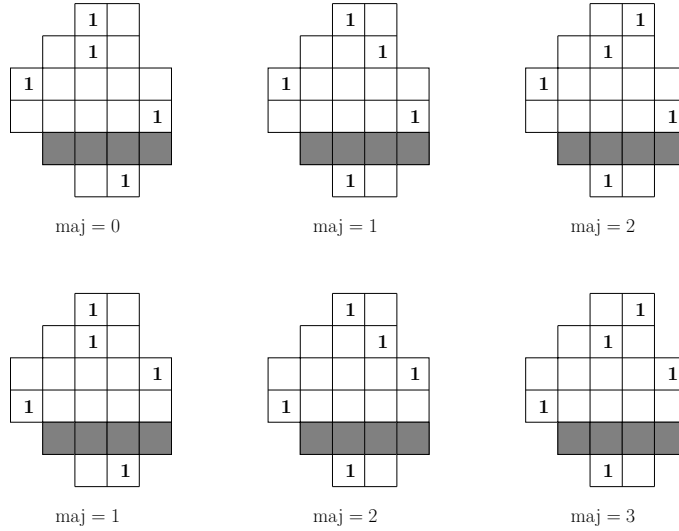
Figure 7: Illustration of Theorem 4.1.

Corollary 4.3. *Let σ be a permutation of $[m]$ and let $\mathcal{M}$ be a moon polyomino with columns $c_1, \ldots, c_m$. Suppose the shape $\mathcal{N}$ with columns $c_i' = c_{\sigma(i)}$ is also a moon polyomino and $\boldsymbol{s}' = (s_1', \ldots, s_m')$ with $s_i' = s_{\sigma(i)}$. Then*

$$\sum_{M \in \boldsymbol{F}(\mathcal{M}, \boldsymbol{s}; A)} q^{\mathrm{maj}(M)} = \sum_{N \in \boldsymbol{F}(\mathcal{N}, \boldsymbol{s}'; A)} q^{\mathrm{maj}(N)}. \tag{4.3}$$

That is, the generating function $\sum_{M \in \boldsymbol{F}(\mathcal{M}, \boldsymbol{s}; A)} q^{\mathrm{maj}(M)}$ does not depend on the order of the columns of $\mathcal{M}$.

Proof. The formula (4.2) seems to depend on the order $\prec$ of the columns of the moon polyomino. However, note that $c_i \prec c_j$ if $|c_i| < |c_j|$; and columns of same length are consecutive in the order $\prec$. Hence it suffices to compare the terms in the right-hand side of (4.2) that come from columns of fixed length. If $c_{i_{j+1}} \prec \cdots \prec c_{i_{j+k}}$ are all the columns of $\mathcal{M}$ of fixed length, then

$$h_{i_{j+r}} = h_{i_{j+1}} - (s_{i_{j+1}} + \cdots + s_{i_{j+r-1}}), \ 2 \le r \le k. \tag{4.4}$$

Thus the contribution of these columns to the right-hand side of (4.2) is

$$\prod_{r=1}^{k} \begin{bmatrix} h_{i_{j+r}} \\ s_{i_{j+r}} \end{bmatrix}_q = \begin{bmatrix} h_{i_{j+1}} \\ s_{i_{j+1}}, \ldots, s_{i_{j+k}}, h_{i_{j+1}} - (s_{i_{j+1}} + \cdots + s_{i_{j+k}}) \end{bmatrix}_q. \tag{4.5}$$

The last q-multinomial coefficient is invariant under permutations of $s_{i_{j+1}}, \ldots, s_{i_{j+k}}$, and the claim follows.

$\square$

The right-hand side of formula (4.2) also appeared in the generating function of NE chains, as shown in [11]:

9

Theorem 4.4 (Kasraoui).
$$\sum_{M \in \boldsymbol{F}(\mathcal{M}, \boldsymbol{s}; A)} q^{\mathrm{ne}(M)} = \prod_{i=1}^{m} \begin{bmatrix} h_i \\ s_i \end{bmatrix}_q.$$

Combining Theorems 4.1 and 4.4, we have

Theorem 4.5.
$$\sum_{M \in \boldsymbol{F}(\mathcal{M}, \boldsymbol{s}; A)} q^{\mathrm{maj}(M)} = \sum_{M \in \boldsymbol{F}(\mathcal{M}, \boldsymbol{s}; A)} q^{\mathrm{ne}(M)}.$$

That is, the maj statistic has the same distribution as the ne statistic over the set $\mathbf{F}(\mathcal{M}, \mathbf{s}, A)$. Since $\mathrm{ne}(M)$ is the natural analogue of inv for words and permutations, Theorem 4.5 generalizes MacMahon's equidistribution result. Unfortunately, as noted in [2], the refinement of MacMahon's theorem which asserts that the joint distribution of maj and inv over permutations is symmetric does not hold even for matchings.

We shall prove a lemma about the major index for words and then use it to prove Theorem 4.1.

Lemma 4.6. *Let $w = w_1 \ldots w_k$ be a word such that $w_i < n$ for all i. Consider the set $S(w)$ of all the words w' that can be obtained by inserting m many n's between the letters of w. Then the difference $\mathrm{maj}(w') - \mathrm{maj}(w)$ ranges over the multiset $\{i_1 + i_2 + \cdots + i_m \mid 0 \le i_1 \le i_2 \le \cdots \le i_m \le k\}$. It follows that*

$$\sum_{w' \in S(w)} q^{\mathrm{maj}(w')} = q^{\mathrm{maj}(w)} \sum_{0 \le i_1 \le i_2 \le \cdots \le i_m \le k} q^{i_1 + i_2 + \cdots + i_m} = q^{\mathrm{maj}(w)} \begin{bmatrix} k + m \\ m \end{bmatrix}_q. \qquad (4.6)$$

The same statement holds if $w = w_1 \ldots w_k$ is a word such that $w_i > 1$ for all i and $S(w)$ is the set of all the words w' obtained by inserting m many 1's between the letters of w.

Proof. We give an elementary proof for the case of inserting n's. The case of inserting 1's is dealt with similarly. For a word $w = w_1 \ldots w_k$ we define the *descent sequence* $\mathrm{desseq}(w) = a_1 \ldots a_{k+1}$ by letting $a_i = \mathrm{maj}(w^{(i)}) - \mathrm{maj}(w)$, where $w^{(i)}$ is obtained by inserting one n in the i-th gap of w, i.e., $w^{(i)} = w_1 \ldots w_{i-1} n w_i \ldots w_k$. Note that

$$\mathrm{maj}(w^{(i)}) = \mathrm{maj}(w) + \begin{cases} 1 + \mathrm{des}(w_i \ldots w_k), & \text{if } w_{i-1} > w_i,\ 2 \le i \le k \text{ or } i = 1 \\ i + \mathrm{des}(w_i \ldots w_k), & \text{if } w_{i-1} \le w_i,\ 2 \le i \le k \\ 0, & \text{if } i = k + 1. \end{cases}$$

Hence, if $t = \mathrm{des}(w)$ then $\mathrm{desseq}(w)$ is a shuffle of the sequences $t+1, t, \ldots, 0$ and $t+2, t+3, \ldots, k$ which begins with $t+1$ and ends with 0. Moreover, every such sequence is $\mathrm{desseq}(w)$ for some word w. By definition, the multiset $\{\mathrm{maj}(w') - \mathrm{maj}(w) \mid w' \in S(w)\}$ depends only on $\mathrm{desseq}(w)$ and not on the letters of w.

First we consider the case when $\mathrm{desseq}(w) = k(k-1) \ldots 0$, i.e., $\mathrm{Des}(w) = [k-1]$. Inserting b n's between w_{i-1} and w_i increases the major index by $b(k - i + 1)$. Hence, if one inserts b_i many n's between w_{i-1} and w_i for $1 \le i \le k + 1$, the major index is increased by

$$\sum_{i=1}^{k+1} b_i(k - i + 1) = \underbrace{k + \cdots + k}_{b_1} + \underbrace{(k-1) + \cdots + (k-1)}_{b_2} + \cdots + \underbrace{0 + \cdots + 0}_{b_{k+1}}.$$

10

Letting $(b_1, \ldots, b_{k+1})$ range over all $(k+1)$-tuples with $\sum_{i=1}^{k+1} b_i = m$, the increased amount ranges over the multiset $\{i_1 + i_2 + \cdots + i_m \mid 0 \leq i_1 \leq i_2 \leq \cdots \leq i_m \leq k\}$.

To deal with the other possible descent sequences, we note that any descent sequence $\mathrm{desseq}(w)$ can be obtained by applying a series of adjacent transpositions to $k(k-1)\ldots 0$, where we only transpose adjacent elements i, j which correspond to ascent and descent positions, respectively. Suppose $\mathrm{desseq}(w) = a_1 \ldots a_{i-1} a_{i+1} a_i a_{i+2} \ldots a_{k+1}$ is obtained by transposing the i-th and $(i+1)$-st element in $\mathrm{desseq}(v) = a_1 \ldots a_{i-1} a_i a_{i+1} a_{i+2} \ldots a_{k+1}$, where a_i and a_{i+1} correspond to an ascent and a descent in v, respectively. Consequently, $w_{i-1} > w_i \leq w_{i+1}$ and $v_{i-1} \leq v_i > v_{i+1}$, while $w_j \leq w_{j+1}$ if and only if $v_j \leq v_{j+1}$ for $j \neq i-1, i$. It suffices to show that the multisets $\{\mathrm{maj}(w') - \mathrm{maj}(w) \mid w' \in S(w)\}$ and $\{\mathrm{maj}(v') - \mathrm{maj}(v) \mid v' \in S(v)\}$ are equal. Suppose $w' \in S(w)$ is obtained by inserting b_j many n's in the j-th gap of w, $1 \leq j \leq k+1$. Let $v' \in S(v)$ be the word obtained by inserting c_j n's in the j-th gap of v where c_j are defined as follows.
If $b_{i+1} = 0$ then

$$c_j = \begin{cases} 0, & \text{if } j = i \\ b_i, & \text{if } j = i+1 \\ b_j, & \text{if } j \neq i, i+1. \end{cases} \tag{4.7}$$

If $b_{i+1} > 0$ then

$$c_j = \begin{cases} b_i + 1, & \text{if } j = i \\ b_{i+1} - 1, & \text{if } j = i+1 \\ b_j, & \text{if } j \neq i, i+1. \end{cases} \tag{4.8}$$

This defines a map between sequences (b_j) with $b_{i+1} = 0$ (resp. $b_{i+1} > 0$), and (c_j) with $c_i = 0$ (resp. $c_i > 0$), which is a bijection. The n's inserted in the j-th gap of w for $j \neq i, i+1$ contribute to $\mathrm{maj}(w') - \mathrm{maj}(w)$ the same amount as do the n's inserted in the same gaps in v to $\mathrm{maj}(v') - \mathrm{maj}(v)$. The major index contributed to $\mathrm{maj}(w')$ by the segment

$$w_{i-1} \underbrace{n \ldots n}_{b_i} w_i \underbrace{n \ldots n}_{b_{i+1}}$$

is the sum of $i - 1$ (which equals the contribution of $w_{i-1} w_i$ to $\mathrm{maj}(w)$) and

$$\begin{cases} (b_1 + \cdots + b_i) + (b_1 + \cdots + b_{i+1} + i), & \text{if } b_{i+1} > 0 \\ (b_1 + \cdots + b_i), & \text{if } b_{i+1} = 0. \end{cases}$$

Similarly, the major index contributed to $\mathrm{maj}(v')$ by the segment

$$v_{i-1} \underbrace{n \ldots n}_{c_i} v_i \underbrace{n \ldots n}_{c_{i+1}}$$

is the sum of i (which equals the contribution of $v_{i-1} v_i$ to $\mathrm{maj}(v)$) and

$$\begin{cases} (c_1 + \cdots + c_i + i - 1) + (c_1 + \cdots + c_{i+1}), & \text{if } c_i > 0 \\ (c_1 + \cdots + c_{i+1}), & \text{if } c_i = 0. \end{cases}$$

Now, one readily checks that $\mathrm{maj}(w') - \mathrm{maj}(w) = \mathrm{maj}(v') - \mathrm{maj}(v)$. This completes the proof.

$\square$

Remark 4.7. Lemma 4.6 can be proved in many ways, for example, in [8, Theorem 1.2] Foata and Han constructed a transformation on the shuffle class $\text{Sh}(0^m v)$, where v is a nonempty word whose letters are positive integers. Alternatively, one can use the theory of P-partitions [9], which leads to a stronger result, namely, Equation (4.6) holds if $w = w_1 \ldots w_k$ with $w_i \neq n_0$, $i = 1, \ldots, k$ and $S(w)$ is the set of all words w' obtained by inserting m copies of n_0 between the letters of w. To show this, it is enough to consider the case when w is a permutation of length k on k distinct numbers between 1 and $k + m$ such that the integers $N_1 < N_2 < \cdots < N_m$ are missing from it and w' varies over the set $S(w)$ of all permutations obtained by inserting $N_1, N_2, \ldots, N_m$ in w from left to right. We outline the proof in this general case assuming that the reader is familiar with the theory of P-partitions. We adopt the notation and definitions from Section 4.5 in [21], which is also a good reference for basic results on P-partitions.

Let P_1 be the antichain with k elements labeled with the letters of w. Let P_2 be a disjoint union of P_1 and a chain of m elements labeled $N_1 < N_2 < \cdots < N_m$. We denote by $F_w(q_1, \ldots, q_k)$ the generating function of all w-compatible functions $f : [k] \to \mathbb{N}$ (see [21] for the definitions). Lemma 4.5.2 in [21] asserts that

$$F_w(q, \ldots, q) = \frac{q^{\text{maj}(w)}}{(1 - q) \cdots (1 - q^k)}. \tag{4.9}$$

All P_2-partitions compatible with some $w' \in S(w)$ are obtained by combining a P_1-partition compatible with w and a integer partition with at most m parts. Therefore,

$$\sum_{w' \in S(w)} F_{w'}(q, \ldots, q) = \frac{F_w(q, \ldots, q)}{(1 - q) \cdots (1 - q^m)}. \tag{4.10}$$

Applying Lemma 4.5.2 in [21] to all $w' \in S(w)$, we have

$$\sum_{w' \in S(w)} F_{w'}(q, \ldots, q) = \sum_{w' \in S(w)} \frac{q^{\text{maj}(w')}}{(1 - q) \cdots (1 - q^{k+m})}. \tag{4.11}$$

Combining equations (4.9), (4.10), and (4.11) gives (4.6).

Proof of Theorem 4.1. Let c be the smallest column of $\mathcal{M}$ in the order $\prec$ and $\mathcal{M}' = \mathcal{M} \backslash c$. Note that c is the leftmost or the rightmost column of $\mathcal{M}$. Let M be a filling of $\mathcal{M}$ with s nonempty cells in c and let M' be its restriction on $\mathcal{M}'$. From Definition 2.2', we derive that

$$\text{maj}(M) = \text{maj}(M') + \text{maj}(R_c) - \text{maj}(R_c \cap M'),$$

where R_c is the filling of the rectangle $\mathcal{M}(c)$ determined by the column c. R_c is obtained by adding a column with s many 1-cells to the rectangular filling $R_c \cap M'$. Using the bijection between fillings of rectangles and words described in Section 3.1, one sees that if c is the leftmost (resp. rightmost) column in M, this corresponds to inserting s maximal (resp. minimal) elements in the word determined by the filling $R_c \cap M'$. When the column c varies over all possible $\binom{h}{s}$ fillings, by Lemma 4.6, the value $\text{maj}(R_c) - \text{maj}(R_c \cap M')$ varies over the multiset $\{i_1 + i_2 + \cdots + i_s \mid 0 \le i_1 \le i_2 \le \cdots \le i_s \le (h - s)\}$ with generating function $\sum_{0 \le i_1 \le \cdots \le i_s \le (h-s)} q^{i_1 + \cdots + i_s} = \begin{bmatrix} h \\ s \end{bmatrix}_q$, where h is the value h_1 (resp. h_m) if c is the leftmost (resp. the rightmost) column of M, as defined in (4.1). Therefore,

$$\sum_{M \in \mathbf{F}(\mathcal{M}, \mathbf{s}; A)} q^{\text{maj}(M)} = \begin{bmatrix} h \\ s \end{bmatrix}_q \sum_{M' \in \mathbf{F}(\mathcal{M}', \mathbf{s}'; A)} q^{\text{maj}(M')}$$

where $\mathbf{s}'$ is obtained from $\mathbf{s}$ by removing the first (resp. last) component if c is the leftmost (resp. rightmost) column of M. Equation (4.2) now follows by induction on the number of columns of $\mathcal{M}$. $\qquad\square$

5 The Foata-type bijection for moon polyominoes

The objective of this section is to give a bijective proof for Theorem 4.5. For the set W_S of all words of a multiset S, the equidistribution of maj and inv was first proved by MacMahon by combinatorial analysis. This raised the question of constructing a canonical bijection $\Phi : W_S \to W_S$ such that $\mathrm{maj}(w) = \mathrm{inv}(\Phi(w))$. Foata answered the question by constructing an elegant map Φ [6], which is referred to as the *second fundamental transformation* [7]. We begin this section by reviewing Foata's map $\Phi : W_S \to W_S$.

Let $w = w_1 w_2 \cdots w_n$ be a word on $\mathbb{N}$ and let a be an integer. If $w_n \leq a$, the a-factorization of w is $w = v_1 b_1 \cdots v_p b_p$, where each b_i is a letter less than or equal to a, and each v_i is a word (possibly empty), all of whose letters are greater than a. Similarly, if $w_n > a$, the a-factorization of w is $w = v_1 b_1 \cdots v_p b_p$, where each b_i is a letter greater than a, and each v_i is a word (possibly empty), all of whose letters are less than or equal to a. In both cases one defines

$$\gamma_a(w) = b_1 v_1 \cdots b_p v_p.$$

With the above notation, let $a = w_n$ and let $w' = w_1 \cdots w_{n-1}$. The second fundamental transformation Φ is defined recursively by $\Phi(w) = w$ if w has length 1, and

$$\Phi(w) = \gamma_a(\Phi(w'))a,$$

if w has length $n > 1$. The map Φ has the property that it preserves the last letter of the word, and $\mathrm{inv}(\Phi(w)) = \mathrm{maj}(w)$.

Foata's map Φ is constructed recursively with certain "local operations" to eliminate the difference caused by adding the last letter in the words. Inspired by this idea, we construct a Foata-type bijection $\phi : \mathbf{F}(\mathcal{M}, \mathbf{s}; A) \to \mathbf{F}(\mathcal{M}, \mathbf{s}; A)$ with the property $\mathrm{maj}(M) = \mathrm{ne}(\phi(M))$. The map ϕ can be defined directly for left-aligned stack polyominoes. But we describe it first for Ferrers diagrams in Section 5.1, because this case contains all the essential steps and is easy to understand. This map is a revision of the Foata-type bijection presented in the preprint of [2] for set partitions, which correspond to fillings with row and column sums at most 1. Then in Section 5.2 we extend the construction to left-aligned stack polyominoes and prove that $\mathrm{maj}(M) = \mathrm{ne}(\phi(M))$. In Section 5.3, we construct two bijections, f and g, that transform a filling of a moon polyomino to a filling of a left-aligned stack polyomino and preserve the statistics maj and ne, respectively. Composing these maps with ϕ defined on left-aligned stack polyominoes yields a bijection on $\mathbf{F}(\mathcal{M}, \mathbf{s}; A)$ that sends maj to ne.

5.1 The bijection ϕ for Ferrers diagrams

The empty rows in the filling $M \in \mathbf{F}(\mathcal{M}, \mathbf{s}; A)$ do not play any role in the definitions of $\mathrm{maj}(M)$ or $\mathrm{ne}(M)$. Therefore, in what follows we assume that $A = \emptyset$ and describe ϕ for fillings without empty rows. For $A \neq \emptyset$, one can first delete the empty rows of the filling, apply the map ϕ, and reinsert the empty rows back.

Let $\mathcal{F}$ be a Ferrers diagram and F a filling of $\mathcal{F}$. The bijection ϕ is defined inductively on the number of rows of F. If F has only one row, then $\phi(F) = F$. Otherwise, we denote by F_1 the filling obtained by deleting the top row r of F. Let $F_1' = \phi(F_1)$ and let F_2 be the filling obtained by performing the algorithm γ_r described below. Then $F' = \phi(F)$ is obtained from F_2 by adding the top row r. So, by definition, ϕ preserves the first row of F.

The algorithm γ_r

If C is the 1-cell of r, then denote by $\mathcal{R}$ the set of all rows of F_1' that intersect the column of C. The 1-cells in $\mathcal{R}$ that are strictly to the left of C are called left and the 1-cells in $\mathcal{R}$ that are weakly to the right of C are called right. The cell C is neither left nor right.

Let CLC (critical left cell) be the topmost left 1-cell, and CRC (critical right cell) be the leftmost right 1-cell that is above CLC. If there is more than one such cell the CRC is defined to be the lowest one. Note that CLC and CRC need not exist. Denote by $\mathcal{R}_1$ the set of all rows weakly below the row of CRC that intersect the column of CRC and $\mathcal{R}_2$ is the set of all rows in $\mathcal{R}$ that do not intersect the column of CRC. If CLC does not exist then both $\mathcal{R}_1$ and $\mathcal{R}_2$ are empty. If CLC exists but CRC does not, then $\mathcal{R}_1$ is empty and $\mathcal{R}_2$ contains all the rows in $\mathcal{R}$ weakly below CLC. See Figure 8 for an illustration.
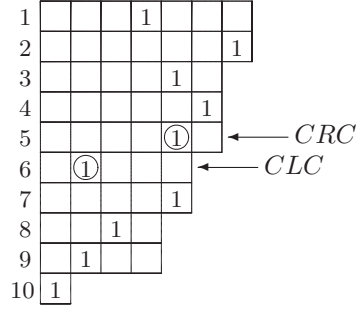

Figure 8: The CRC and CLC are in rows 5 and 6, respectively. $\mathcal{R}_1$ contains rows 5, 6, and 7, while $\mathcal{R}_2$ contains rows 8 and 9.

Definition 5.1. Let C_1 and C_2 be two 1-cells with coordinates (i_1, j_1) and (i_2, j_2), respectively. We *swap the cells C_1 and C_2* by deleting the 1's from these two cells and write 1's in the cells with coordinates (i_1, j_2) and (i_2, j_1).

For a cell C, denote by $col(C)$ the column of C, and $|col(C)|$ the length of $col(C)$.

Algorithm γ_r^1 on $\mathcal{R}_1$

Let ptr_1 and ptr_2 be two pointers.

(A) Set ptr_1 on the highest row of $\mathcal{R}_1$ and ptr_2 on the next row in $\mathcal{R}_1$ below ptr_1.

(B) If ptr_2 is null, then go to (D). Otherwise, the pointers ptr_1 and ptr_2 point at 1-cells C_1 and C_2, respectively.

(B1) If C_2 is a left cell then swap the cells C_1 and C_2 and move ptr_1 to the row of ptr_2.

(B2) If C_2 is a right cell, then

> **(B2.1)** If $|col(C_1)| = |col(C_2)|$ then move ptr_1 to the row of ptr_2.
>
> **(B2.2)** If C_1 is to the left of C_2 and $|col(C_1)| > |col(C_2)|$ then do nothing.
>
> **(B2.3)** If C_1 is to the right of C_2 and $|col(C_1)| < |col(C_2)|$, then find the lowest left 1-cell L that is above C_1. Suppose that the row-column coordinates of the cells L, C_1, and C_2 are (i_1, j_1), (i_2, j_2), and (i_3, j_3), respectively. Delete the 1's from these three cells and write them in the cells with coordinates (i_1, j_3), (i_2, j_1), and (i_3, j_2). Move ptr_1 to the row of ptr_2.†

(C) Move ptr_2 to the next row in $\mathcal{R}_1$. Go to (B).

(D) Stop.

See Figure 9 for illustration of the steps.

†**Note:** This step is well-defined since it cannot occur before ptr_2 reaches CLC (that would contradict the definition of CRC) and, after that, ptr_1 is always below CLC. Therefore, the 1-cell L always exists. The fact that the square (i_3, j_2) belongs to $\mathcal{F}$ follows from the definition of $\mathcal{R}_1$ and Lemma 5.2 (c).

When the algorithm γ_r^1 stops, we continue processing the rows of $\mathcal{R}_2$ by algorithm γ_r^2.

The algorithm γ_r^2 on $\mathcal{R}_2$

(A′) Set ptr_1 on the highest row of $\mathcal{R}_2$ and ptr_2 on the next row in $\mathcal{R}_2$ below ptr_1.

(B′) (***Borrowing***) If ptr_1 points to a right 1-cell then find the lowest left 1-cell above it and swap them. Now ptr_1 points to a left 1-cell.

(C′) If ptr_2 is null, then go to (E′). Otherwise, the pointers ptr_1 and ptr_2 point at 1-cells C_1 and C_2 respectively.

> **(C′1)** If C_2 is a right cell then swap C_1 and C_2.
>
> **(C′2)** If C_2 is a left cell then do nothing.

(D′) Move ptr_1 to the row of ptr_2 and ptr_2 to the next row in $\mathcal{R}_2$ below. Go to (C′).

(E′) Stop.

See Figure 10 for illustration of the steps in γ_r^2.

Let us note the following easy but useful properties of the algorithm γ_r.

1. The pointers ptr_1 and ptr_2 process the rows of $\mathcal{R}_1$ and $\mathcal{R}_2$ from top to bottom. However, while ptr_2 always moves from one row to the next one below it, ptr_1 sometimes stays on the same row (cf. step (B2.2)) and sometimes "jumps" several rows below (cf. step (B1)).

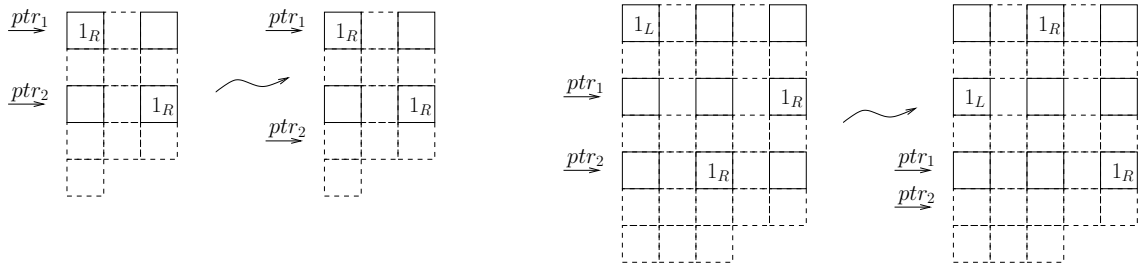2. Pointer ptr_1 always points to a right 1-cell in γ_r^1 and to a left 1-cell in γ_r^2.

Further we show some nontrivial properties which will be used to show that the map ϕ has the desired property $\mathrm{maj}(F) = \mathrm{ne}(\phi(F))$.

15
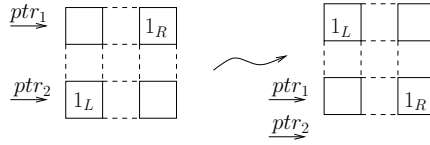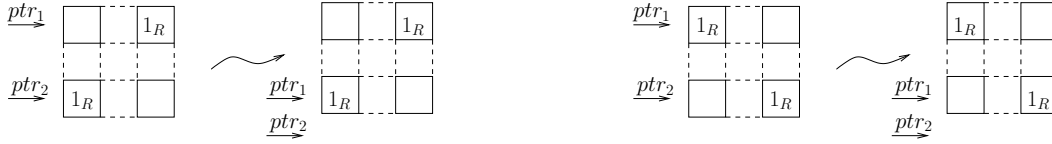
Step (B1).

Step (B2.1).

Step (B2.2).   **Step (B2.3).**

Figure 9: The algorithm γ_r^1 on $\mathcal{R}_1$.

Lemma 5.2. *Suppose in γ_r^1 ptr_1 is pointing to C_1 and ptr_2 is pointing to C_2. Then*

(a) *In the rows between C_1 and C_2 there are no 1's weakly to the left of C_1.*

(b) *If L is the lowest left 1-cell above C_1, then in the rows between L and C_1 there are no 1's weakly to the left of C_1.*

(c) *The 1-cell C_1 is in a column with same length as the column of the critical right cell CRC. Moreover, when γ_r^1 stops ptr_1 points to the last row in $\mathcal{R}_1$.*

Proof. The first two parts are proved by induction on the number of steps performed in the algorithm.

(a) The only step which leaves the gap between ptr_1 and ptr_2 nonempty is (B2.2). But after this step, the number of 1's in the rows between ptr_1 and ptr_2 weakly to the left of C_1 does not change.

(b) The case when the last step is (B2.2) follows from the induction hypothesis. If the last step is (B1) or (B2.3) the claim follows from part (a). The case when the last step was (B2.1)
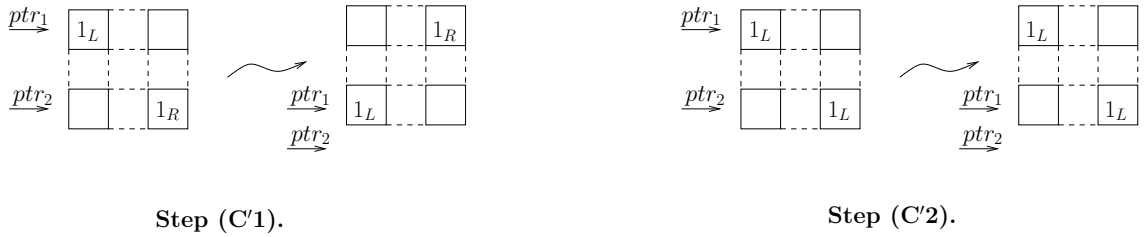
16

Figure 10: Algorithm γ_r^2 on $\mathcal{R}_2$.

and the cells C_1 and C_2 formed an NE chain the claim also follows from part (a) and the induction hypothesis. If in (B2.1), the cells C_1 and C_2 do not form an NE chain, in addition to the induction hypothesis, one uses the fact that there are no 1's in the rows between these two cells which are weakly to the left of C_2.

(c) The first part follows immediately from the definition of the steps. If the second part is not true, then the last step must be (B2.2). and the column of C_1 is longer than the column of C_2. Since this is the last step of γ_r^1, it follows that the column of C_1 intersects the top row of $\mathcal{R}_2$. However, the column of CRC does not intersect the top row of $\mathcal{R}_2$ by definition of $\mathcal{R}_2$. This contradicts the fact that C_1 and CRC are in columns of same length.

$\square$

Proposition 5.3. *During algorithm γ_r^1 the number of NE chains decreases by the total number of left 1-cells in $\mathcal{R}_1$, whereas during algorithm γ_r^2 the number of NE chains increases by the total number of right 1-cells in $\mathcal{R}_2$.*

Proof. To prove the first part note that pointer ptr_2 points to each left 1-cell in $\mathcal{R}_1$ exactly once in the algorithm γ_r^1. When that happens, step (B1) is performed during which, by Lemma 5.2(a), the number of NE chains decreases by one. In the other steps, the number of NE chains remains unchanged. This is trivial for steps (B2.1) and (B2.2), whereas for step (B2.3) it follows from Lemma 5.2 (a) and (b).

For the second part, note that pointer ptr_2 points to each right 1-cell in $\mathcal{R}_2$ exactly once in the algorithm γ_r^2 with possible exception being the top 1-cell in $\mathcal{R}_2$ to which it never points. In those steps the number of NE chains is increased by one, while otherwise it remains the same. When the top 1-cell is right, then Borrowing occurs. It follows from parts (b) and (c) of Lemma 5.2 that the number of NE chains is also increased by one. $\square$

Theorem 5.4. *The map $\phi : \boldsymbol{F}(\mathcal{F}, \boldsymbol{s}; A) \to \boldsymbol{F}(\mathcal{F}, \boldsymbol{s}; A)$ is a bijection.*

The proof is quite technical and is given in the appendix. Here we only present an example of the bijection ϕ in Figure 11 and explain how it includes Foata's second fundamental transformation. If the moon polyomino is rectangular, then the algorithm γ_r has a simpler description because some of the cases can never arise. Namely, if the cell in the second row forms an NE chain with the top cell (a descent) then $\mathcal{R}_1 = \emptyset$, and only the steps (C'1) and (C'2) are performed. Otherwise, $\mathcal{R}_2 = \emptyset$ and, since all columns are of equal height, only the steps (B1) and (B2.1) are performed. In both
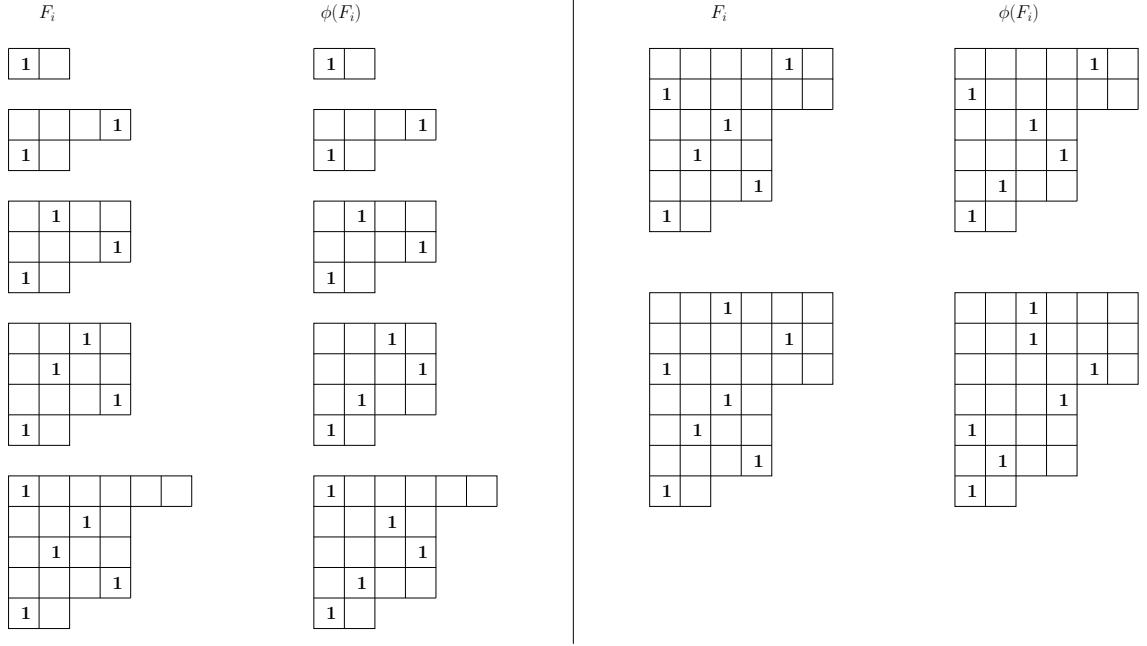
17

Figure 11: Example of the map ϕ applied inductively on $F = F_7$. The fillings F_i are restrictions of F on the last i rows.

these cases, via the correspondence between words and rectangular fillings (c.f. Section 3.1) the algorithms γ_r^2 and γ_r^1 for rectangles are equivalent to Foata's transformation for words [6].

5.2 The bijection ϕ for left-aligned stack polyominoes

Next we extend the bijection ϕ to fillings of left-aligned stack polyominoes which sends maj to ne. Let $\mathcal{M}$ be a left-aligned stack polyomino. As in the case of Ferrers diagrams, we can assume that $A = \emptyset$ and only consider fillings without empty rows. Suppose M is a filling of $\mathcal{M}$ with top row r. The map, which is again denoted by ϕ, is defined inductively on the number of rows of M. If M has only one row, then $\phi(M) = M$. Otherwise, let $\mathcal{F}$ be the maximal Ferrers diagram in $\mathcal{M}$ that contains the top row and $\mathcal{F}_1$ be $\mathcal{F}$ without the top row. Let $F = M \cap \mathcal{F}$ and $F_1 = M \cap \mathcal{F}_1$. To obtain $\phi(M)$ we perform the following steps. (See Figure 12).

1. Delete the top row of M and get $M_1 = M \backslash r$.

2. Apply ϕ to M_1 and get $M_1' = \phi(M_1)$.

3. Apply the algorithm γ_r to the filling $F_1' = M_1' \cap \mathcal{F}_1$ and leave the cells in M_1' outside of F_1' unchanged. Denote the resulting filling by M_2.

4. $\phi(M)$ is obtained by adding row r back to M_2.

It is a bijection from $\mathbf{F}(\mathcal{M}, \mathbf{s}; A)$ to itself since every step is invertible. That the Step 3 is invertible follows from the proof of Theorem 5.4.
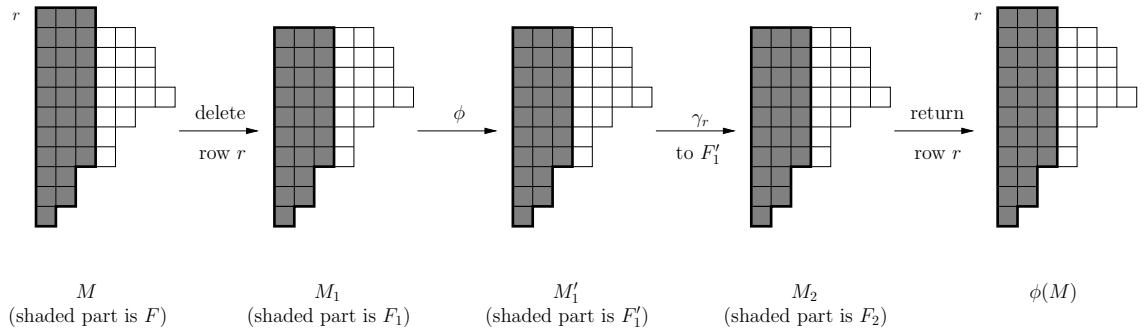
18

Figure 12: Illustration of ϕ for left-aligned stack polyominoes.

Lemma 5.5. *Using the notation introduced above,*

$$\text{maj}(M) - \text{maj}(M_1) = \text{maj}(F) - \text{maj}(F_1) = \#\mathcal{R}_2(F_1) \tag{5.1}$$

where $\mathcal{R}_2(F_1)$ is defined with respect to the top row r of F as in Section 5.1.

Proof. Let S be the filling M restricted to the rows which are not completely contained in $\mathcal{F}$. From the definition of maj, we have

$$\text{maj}(M) = \text{maj}(F) + \text{maj}(S) - \text{maj}(F \cap S) \tag{5.2}$$
$$\text{maj}(M_1) = \text{maj}(F_1) + \text{maj}(S) - \text{maj}(F_1 \cap S) \tag{5.3}$$

Note that $F \cap S = F_1 \cap S$, hence (5.2) and (5.3) give the first equality in (5.1). The second one is readily checked using Proposition 3.1 for major index of stack polyominoes. □

The rest of this subsection is devoted to proving that the map ϕ has the desired property $\text{maj}(M) = \text{ne}(\phi(M))$. We shall need one more lemma.

Let C be a 1-cell in a filling M of a left-aligned stack polyomino. We say that C is *maximal* if there is no 1 in M which is strictly above and weakly to the left of C. Clearly, a maximal cell must be the highest 1-cell in its column. For a maximal cell C in M, we use $t(C, M)$ to denote the index of the leftmost column (the columns are numbered from left to right) to the right of C that contains a 1-cell which together with C forms an NE chain. If such a column does not exist set $t(C, M) = \infty$.

Lemma 5.6. *A 1-cell C in M is maximal if and only if the highest 1-cell C' in the same column in $M' = \phi(M)$ exists and is maximal. Moreover, $t(C, M) = t(C', M')$ when they are both maximal.*

Proof. We proceed by induction on the number of rows of M and keep the notation as in Figure 11. The case when M has one row is trivial.

Suppose that the top row r of M has a 1-cell C^* in the column i^*. The cell C^* is maximal in both M and M' and $t(C^*, M) = \infty = t(C^*, M')$. Note that a 1-cell $C \neq C^*$ in M (resp. M') is maximal if and only if C is in column $i < i^*$ and is maximal in M_1 (resp. M_2).

Denote by C_1, C_1', and C_2 the highest 1-cells in column $i < i^*$ in the fillings M_1, $M_1' = \phi(M_1)$, and M_2, respectively. By the induction hypothesis, C_1 is maximal in M_1 if and only if C_1' is maximal

19

in M_1'. M_2 is obtained from M_1' by applying the algorithm γ_r with respect to the row r to F_1'. C_1' is a left cell in this algorithm, and since γ_r does not change the relative position of the left 1-cells, C_2 is maximal in M_2.

It only remains to prove that $t(C, M) = t(C', M')$. Suppose $t(C_1, M_1) = t(C_1', M_1') = a$ (they are equal by the inductive hypothesis).

If $a < i^*$ then $t(C, M) = a$. In M_1' it corresponds to a 1-cell D' in a column left of C^* such that C_1', D' form an NE chain. Since the algorithm γ_r preserves the column sums and the relative positions of left 1-cells, we have $t(C_2, M_2) = a$, and hence $t(C', M') = a$.

If $a \geq i^*$ then $t(C, M) = i^*$. The fact that $t(C_1', M_1') = a \geq i^*$ means that there is no left 1-cell in M_1' above and to the right of C_1' and, since γ_r does not change the relative positions of the left 1-cells, there will be no left 1-cell above and to the right of C_2 in M_2. So, in this case, we conclude $t(C', M') = i^* = t(C, M)$. $\square$

Theorem 5.7. *If M is a filling of a left-aligned stack polyomino $\mathcal{M}$, then $\mathrm{maj}(M) = \mathrm{ne}(\phi(M))$.*

Proof. Again we proceed by induction on the number of rows of M. The case when M has one row is trivial. We use the notation as in Figure 11, and let $F_2 = M_2 \cap \mathcal{F}_1$.

By Proposition 5.3, the algorithm γ_r on M_1' decreases ne by one for each left 1-cell in $\mathcal{R}_1(M_1')$ and increases it by one for each right 1-cell in $\mathcal{R}_2(M_1')$. Therefore,

$$\begin{aligned}
\mathrm{ne}(\phi(M)) &= \mathrm{ne}(M_2) + \#\{\text{left 1-cells in } \mathcal{R}(F_2)\} \\
&= \mathrm{ne}(M_1') - \#\{\text{left 1-cells in } \mathcal{R}_1(F_1')\} + \#\{\text{right 1-cells in } \mathcal{R}_2(F_1')\} \\
&\quad + \#\{\text{left 1-cells in } \mathcal{R}(F_2)\}
\end{aligned}$$

Since γ_r preserves the column sums,

$$\#\{\text{left 1-cells in } \mathcal{R}(F_2)\} = \#\{\text{left 1-cells in } \mathcal{R}(F_1')\}$$

hence

$$\mathrm{ne}(\phi(M)) = \mathrm{ne}(M_1') + \#\mathcal{R}_2(F_1').$$

One the other hand, from Lemma 5.5 and the induction hypothesis, one gets

$$\mathrm{maj}(M) = \mathrm{maj}(M_1) + \#\mathcal{R}_2(F_1) = \mathrm{ne}(M_1') + \#\mathcal{R}_2(F_1).$$

So, it suffices to show that

$$\#\mathcal{R}_2(F_1) = \#\mathcal{R}_2(F_1'). \tag{5.4}$$

To show equation (5.4), note that for a filling of a Ferrers diagram F the number of rows in $\mathcal{R}_2(F)$ is determined by the column indices of CLC (critical left cell) and CRC (critical right cell). The CLC is the topmost left 1-cell which, if exists, is the topmost *maximal* cell besides the 1-cell in the first row. By Lemma 5.6, CLC in $F_1 \subseteq M_1$ is in the same column as CLC' in $F_1' \subseteq M_1'$, or neither of them exists. The latter case is trivial, as $\mathcal{R}_2(F)$ is empty. In the first case, let $|r|$ denote the length of row r. Then CRC, the critical right cell of F_1, is in the column $t(CLC, M_1)$ if $t(CLC, M_1) \leq |r|$ and does not exist otherwise. Similarly, the critical right cell of F_1', CRC', is in the column $t(CLC', M_1')$ if $t(CLC, M_1') \leq |r|$, and does not exist otherwise. By Lemma 5.6, $t(CLC, M_1) = t(CLC', M_1')$, and this implies (5.4). $\square$

5.3 The case of a general moon polyomino

Now we consider the case when $\mathcal{M}$ is a general moon polyomino. We label the rows of $\mathcal{M}$ by $r_1, \ldots, r_n$ from top to bottom. Let $\mathcal{N}$ be the unique left-aligned stack polyomino whose sequence of row lengths is equal to $|r_1|, \ldots, |r_n|$ from top to bottom. In other words, $\mathcal{N}$ is the left-aligned polyomino obtained by rearranging the columns of $\mathcal{M}$ by length in weakly decreasing order from left to right. For the definitions that follow, we use an idea of Rubey [19] to describe an algorithm that rearranges the columns of $\mathcal{M}$ to obtain $\mathcal{N}$ (Figure 13).

Algorithm α for rearranging $\mathcal{M}$:

1. Set $\mathcal{M}' = \mathcal{M}$.
2. If $\mathcal{M}'$ is left aligned go to (4).
3. If $\mathcal{M}'$ is not left-aligned consider the largest rectangle $\mathcal{R}$ completely contained in $\mathcal{M}'$ that contains c_1, the leftmost column of $\mathcal{M}'$. Update $\mathcal{M}'$ by letting $\mathcal{M}'$ be the polyomino obtained by moving the leftmost column of $\mathcal{R}$ to the right end. Go to (2).
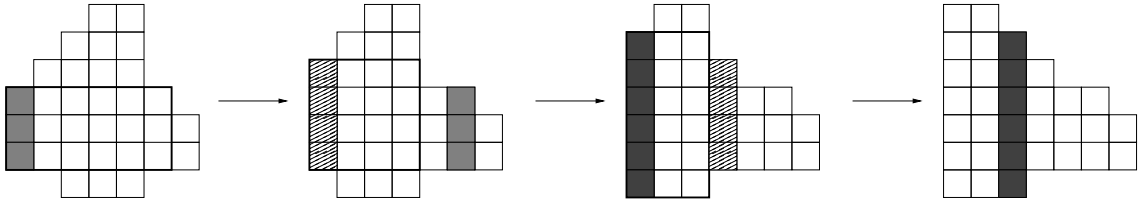4. Set $\mathcal{N} = \mathcal{M}'$.

Figure 13: The algorithm α.

In this section we give two bijections $f, g : \mathbf{F}(\mathcal{M}, \mathbf{s}; A) \to \mathbf{F}(\mathcal{N}, \mathbf{s}'; A)$ which preserve the major index and the number of NE chains, respectively. The sequence $\mathbf{s}'$ is obtained by rearranging the sequence $\mathbf{s}$ in the same way $\mathcal{N}$ is obtained by rearranging the columns of $\mathcal{M}$.

5.3.1 Bijection $f : \mathbf{F}(\mathcal{M}, \mathbf{s}; A) \to \mathbf{F}(\mathcal{N}, \mathbf{s}'; A)$ such that $\mathrm{maj}(M) = \mathrm{maj}(f(M))$

Let R be a filling of a rectangle $\mathcal{R}$ with column sums $s_1, s_2, \ldots, s_m$. First, we describe a transformation τ which gives a filling of $\mathcal{R}$ with column sums $s_2, s_3, \ldots, s_m, s_1$ and preserves the major index. Recall that a descent in R is a pair of 1's in consecutive nonempty rows that form an NE chain. Define an *ascent* to be a pair of 1's in two consecutive nonempty rows that does not form an NE chain. Suppose the first column of R has k many 1-cells: $C_1, \ldots, C_k$ from top to bottom.

Transformation τ on rectangles:

1. Let R^* be the rectangle obtained by adding one empty column to R from right. Process the 1-cells $C_1, \ldots, C_k$ from bottom to top (see Figure 14):

2. For $r = k, k-1, \ldots, 1$ do the following:

    Let D_a be the lowest 1-cell above C_r in R^* and D_b be the highest 1-cell below C_r in R^*.
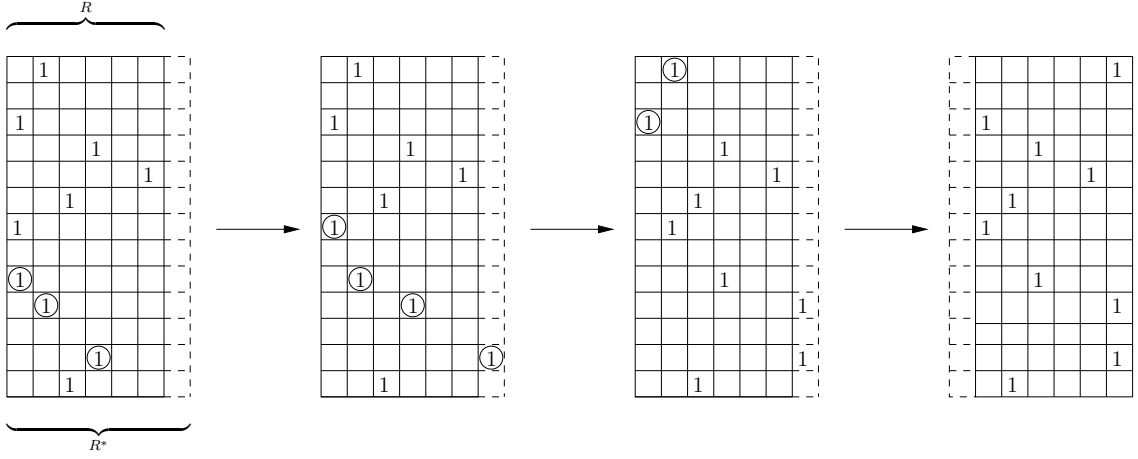
Figure 14: Illustration of the transformation τ for rectangles.

(a) If D_a does not exist or $D_a = C_{r-1}$, and D_b does not exist, just move C_r horizontally to the last column of R^*;

(b) If (i) D_a does not exist or $D_a = C_{r-1}$, but D_b exists, or (ii) both D_a and D_b exist, and D_a is strictly to the right of D_b:

Let $D_1 = C_r, D_2 = D_b, D_3, \ldots, D_p$ be the maximal chain of consecutive ascents in R^*. Move D_i horizontally to the column of D_{i+1}, for $i = 1, \ldots, p-1$, and move D_p horizontally to the last column of R^*. (If D_p is in the last column of R^*, it remains there.)

(c) If $D_a \neq C_{r-1}$, and (D_a is weakly to the left of D_b or D_b does not exist):

Let $D_1 = C_r, D_2 = D_a, D_3, \ldots, D_p$ be the maximal chain of consecutive descents in R^*. Move D_i horizontally to the column of D_{i+1}, for $i = 1, \ldots, p-1$, and move D_p horizontally to the last column of R^*.

3. Delete the first column of R^*.

Note that each iteration in Step 2 decreases the number of 1's in the first column of R^* by one and increases the number of 1's in the last column by one, while the other column sums remain unchanged.

Lemma 5.8. *The transformation τ is invertible. Moreover, it preserves the descents and hence* $\mathrm{maj}(R) = \mathrm{maj}(\tau(R))$.

Proof. The map τ is invertible since τ^{-1} can be obtained by taking $\rho \circ \tau \circ \rho$, where ρ is the rotation of rectangles by $180°$. Moreover, each iteration of Step 2 of τ preserves the positions of descents, hence the second part of the claim holds. $\square$

Now we are ready to define the map f. Suppose $M \in \mathbf{F}(\mathcal{M}, \mathbf{s}; A)$. We perform the algorithm α on M to transform the shape $\mathcal{M}$ to $\mathcal{N}$. While we are in Step 3, instead of just moving the first column of $\mathcal{R}$ to the right end, we perform the algorithm τ on the filling R of $\mathcal{R}$. $f(M)$ is defined to be the resulting filling of $\mathcal{N}$.

Proposition 5.9. *The map $f : \boldsymbol{F}(\mathcal{M}, \boldsymbol{s}; A) \to \boldsymbol{F}(\mathcal{N}, \boldsymbol{s}'; A)$ is a bijection and* $\mathrm{maj}(M) = \mathrm{maj}(f(M))$.

Proof. The first part follows from Lemma 5.8 and the fact that α is invertible. The filling $f(M)$ is obtained after several iterations of Step 3 in the algorithm α combined with application of τ. For the second part of the claim, it suffices to show that maj is preserved after one such iteration. Let M' be the filling of the shape $\mathcal{M}'$ obtained from M after one step of f in which the rectangle R is transformed into $\tau(R)$. We shall use Definition 2.2 to compare $\mathrm{maj}(M)$ and $\mathrm{maj}(M')$. Let $R_1, \ldots, R_k$ (resp. $R_1', \ldots, R_k'$) be the maximal rectangles of M (resp. M'), ordered by height. In particular, R_i and R_i' are of same size. Suppose $R = R_s$, then $R_s' = \tau(R)$. We partition the set of rectangles into three subsets according to $i < s$, $i = s$, and $i > s$, and compare the contribution of R_i's in each subset.

1. For $1 \le i \le s - 1$, the rectangles R_i and $R_i \cap R_{i+1}$ are of smaller height than R and contain several consecutive rows of R. Using Lemma 5.8 we know that the descents in $R_i' \cap \tau(R)$ (resp. $R_i' \cap R_{i+1}' \cap \tau(R)$) $1 \le i \le s-1$, are in the same rows as in $R_i \cap R$ (resp. $R_i \cap R_{i+1} \cap R$). And descents formed by one cell in R and one cell on the right of R are preserved since τ preserves the set of rows that contain a non-zero entry. Consequently,

$$\mathrm{maj}(R_i) = \mathrm{maj}(R_i'), \ \mathrm{maj}(R_i \cap R_{i+1}) = \mathrm{maj}(R_i' \cap R_{i+1}'), \ 1 \le i \le s-1. \tag{5.5}$$

2. For $i = s$, by Lemma 5.8, we have

$$\mathrm{maj}(R_s) = \mathrm{maj}(R_s'). \tag{5.6}$$

3. For $i > s$, the height of the rectangle R_i or $R_{i-1} \cap R_i$ is greater than or equal to that of R. Each of these rectangles contains several consecutive columns of R excluding the first one. The filling of each subrectangle of R consisting of the columns $c_j, \ldots, c_{j+m}$ $(j > 1)$ is equal, up to a rearrangement of the empty rows, to the filling of the subrectangle of $\tau(R)$ consisting of the columns $c_{j-1}, \ldots, c_{j+m-1}$. To see this, note that the shifting of the 1-cells $D_1, \cdots, D_p$ horizontally in the algorithm τ can be viewed as moving the 1-cell D_i vertically to the row of D_{i-1} and moving D_1 to the last column of R^* in the row of D_p. Thus, the descent positions in the fillings $R_i' \cap \tau(R)$ (resp. $R_{i-1}' \cap R_i' \cap \tau(R)$), $i > s$ are the same as in $R_i \cap R$ (resp. $R_{i-1} \cap R_i \cap R$). Finally, one checks that the descents formed by one cell in R and one cell outside R are also preserved. Therefore,

$$\mathrm{maj}(R_i) = \mathrm{maj}(R_i'), \ \mathrm{maj}(R_{i-1} \cap R_i) = \mathrm{maj}(R_{i-1}' \cap R_i'), \ i > s. \tag{5.7}$$

Combining (5.5), (5.6), and (5.7) gives $\mathrm{maj}(M) = \mathrm{maj}(M')$. $\qquad\square$

5.3.2   Bijection $g : \mathbf{F}(\mathcal{M}, \mathbf{s}; A) \to \mathbf{F}(\mathcal{N}, \mathbf{s}'; A)$ such that $\mathrm{ne}(M) = \mathrm{ne}(g(M))$

Suppose $M \in \mathbf{F}(\mathcal{M}, \mathbf{s}; A)$. To obtain $g(M)$, we perform the algorithm α to transform the shape $\mathcal{M}$ to $\mathcal{N}$ and change the filling when we move columns in Step 3 so that the number of 1's in each row and column is preserved.

Let R be the rectangular filling in Step 3 of α that contains the column c_1 of the current filling. Suppose c_1 contains k many 1-cells $C_1, \ldots, C_k$ from top to bottom. Shade the empty rows of R and the cells in R to the right of $C_1, \ldots, C_k$. Let l_i denote the number of empty white cells in R above C_i. If $\mathcal{R}'$ is the rectangle obtained by moving the column c_1 from first to last place, fill it to obtain a filling R' as follows.

1. The rows that were empty remain empty. Shade these rows.

2. Write k many 1's in the last column from bottom to top so that there are l_i white empty cells below the i-th 1, $1 \le i \le k$.

3. Shade the cells to the left of the nonempty cells in the last column.

4. Fill in the rest of the rectangle by writing 1's in the unshaded rows of $\mathcal{R}'$ so that the unshaded part of R to the right of the first column is the same as the unshaded part of R' to the left of the last column.
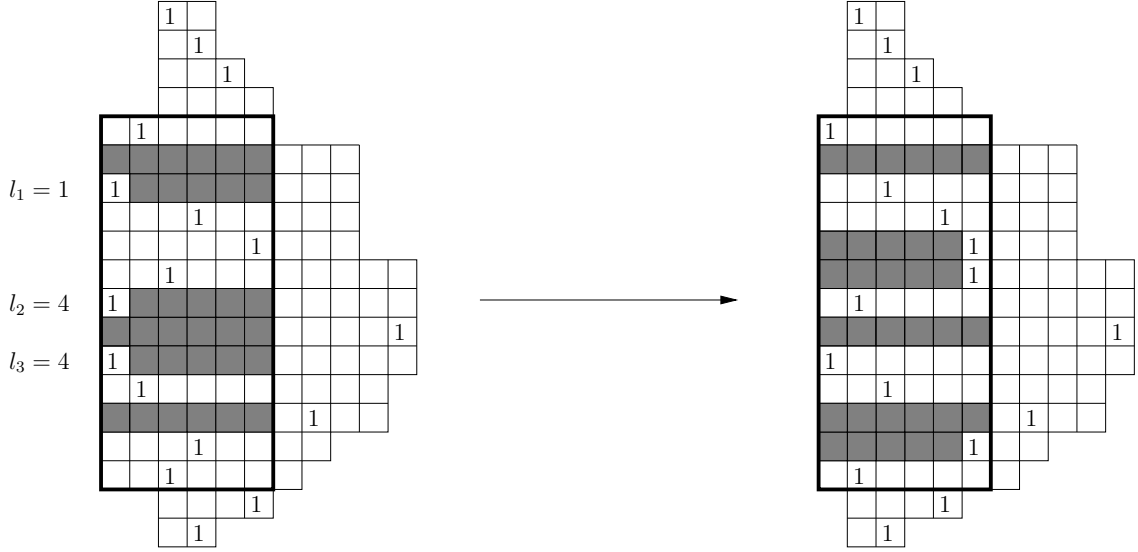
See Figure 15 for an example.

Figure 15: One step of the map g.

Proposition 5.10. *The map* $g : \boldsymbol{F}(\mathcal{M}, \boldsymbol{s}; A) \to \boldsymbol{F}(\mathcal{N}, \boldsymbol{s}'; A)$ *is a bijection and* $\mathrm{ne}(M) = \mathrm{ne}(g(M))$.

Proof. Clearly, every step of g is invertible. To see that ne is preserved, it suffices to show that it is preserved in each step of g, when one column is moved from left to right. First note that $\mathrm{ne}(R) = \mathrm{ne}(R')$. This follows from the fact that the 1's in the first column of R form $l_1, \ldots, l_k$ NE chains from top to bottom, while the 1's in the last column of R' form $l_1, \ldots, l_k$ NE chains from bottom to top and the remaining parts of R and R' are essentially equal. The numbers of NE chains in M and M' respectively made of at least one 1 which is outside of R and R' respectively are equal because each step of g preserves the row and column sums. $\qquad\square$

Theorem 5.11. *Let* ϕ *denote the Foata-type bijection described in Section 5.2. For a general moon polyomino* M *the map* $\psi = g^{-1} \circ \phi \circ f : \boldsymbol{F}(\mathcal{M}, \boldsymbol{s}; A) \to \boldsymbol{F}(\mathcal{M}, \boldsymbol{s}; A)$ *is a bijection with the property*

$$\mathrm{maj}(M) = \mathrm{ne}(\psi(M)) \tag{5.8}$$

24

Proof. The maps f and g permute the columns of M together with the corresponding number of 1's, while ϕ preserves the column sums. Moreover, all three of them preserve the empty rows. Therefore, ψ is indeed a map from $\mathbf{F}(\mathcal{M}, \mathbf{s}; A)$ to itself. Finally, (5.8) follows from Proposition 5.9, Proposition 5.10, and Theorem 5.7. $\qquad\square$

Acknowledgment

Appendix

To show that the map $\phi : \mathbf{F}(\mathcal{F}, \mathbf{s}; A) \to \mathbf{F}(\mathcal{F}, \mathbf{s}; A)$ is a bijection, it suffices to describe the inverse of the algorithm γ_r. First, we need to determine what $\mathcal{R}_1$ and $\mathcal{R}_2$ were. For that we use the following properties of γ_r.

1. When γ_r^1 stops, the pointer ptr_1 is at the lowest row of $\mathcal{R}_1$ pointing at a right 1-cell C_1. The algorithm γ_r^2 does not change the cell C_1 and when the whole algorithm γ_r terminates, there is no 1-cell below C_1 that together with it forms an NE chain.

2. After performing γ_r^2, all the right 1-cells in $\mathcal{R}_2$ have at least one 1-cell below them with which they form an NE chain.

3. If $\mathcal{R}_2$ was nonempty then, after performing γ_r, the lowest row of $\mathcal{R}_2$ contains a left 1-cell.

4. If any borrowing occurred and the right cell $C_1 = (i_1, j_1)$ and the left cell $C_2 = (i_2, j_2)$ were swapped, then right after this step the cell (i_1, j_2) is a right 1-cell in $\mathcal{R}_1$ which forms an NE chain with a left 1-cell from $\mathcal{R}_2$ and this is the lowest 1-cell in $\mathcal{R}_1$ with this property.

Therefore, if $\mathcal{R}$ has no left 1-cells then $\mathcal{R}_2 = \mathcal{R}_1 = \emptyset$. Otherwise, find the lowest right 1-cell C^* in $\mathcal{R}$ such that there is no 1-cell in $\mathcal{R}$ below it that together with C^* forms an NE chain. Then, by Properties 1 and 2 all the nonempty rows in $\mathcal{R}$ below C^* are in $\mathcal{R}_2$. If there is no 1-cell C^* with that property, then $\mathcal{R}_2 = \mathcal{R}$.

Algorithm δ_r^2: the inverse of γ_r^2

(IA$'$) Initially, set ptr_1 to the lowest row of $\mathcal{R}_2$ and ptr_2 to the next row in $\mathcal{R}_2$ above ptr_1.

(IB$'$) If ptr_2 is null then go to step (ID$'$). Otherwise, suppose ptr_1 and ptr_2 are pointing at 1-cells C_1 and C_2.

    (IB$'$1) If C_2 is a left cell do nothing.

    (IB$'$2) If C_2 is a right cell then swap the cells C_1 and C_2.

(IC$'$) Move ptr_1 to the row of ptr_2 and ptr_2 to the next row in $\mathcal{R}_2$ above it. Go to (IB$'$).

(ID′) (*Inverse borrowing*) Note that ptr_1 always points to a left 1-cell. When it reaches the highest row of $\mathcal{R}_2$ and points to a 1-cell C_1 first we need to determine whether there was any borrowing. For that purpose find the lowest right 1-cell C_2 above C_1 such that the two cells form an NE chain. Using Property 4 we conclude:

  (ID′1) If there is no such a cell C_2, then there was no borrowing so do nothing.

  (ID′2) If there is such a cell C_2, and there is a left 1-cell in the rows between C_1 and C_2, then there was no borrowing so do nothing.

  (ID′3) If there is such a cell C_2, and there is no left 1-cell between C_1 and C_2, then swap C_1 and C_2.

(IE′) Stop.

When the algorithm δ_r^2 stops, continue by applying the algorithm δ_r^1 on $\mathcal{R}\backslash\mathcal{R}_2$.

Algorithm δ_r^1: the inverse of γ_r^1

(IA) Position ptr_1 on the lowest row in $\mathcal{R}\backslash\mathcal{R}_2$ and ptr_2 on the lowest nonempty row in $\mathcal{R}\backslash\mathcal{R}_2$ above it.

(IB) If ptr_2 is null then go to step (ID). Otherwise, suppose that ptr_1 and ptr_2 point at the 1-cells C_1 and C_2 respectively.

  (IB1) If C_2 is a left cell then check whether there exists a right 1-cell R above C_2 in a column longer than the column of C_1 such that there are no left 1-cells between R and C_2.

    (IB1.1) If such a cell R exists, then suppose that the row-column coordinates of C_1, C_2, and R are (i_1, j_1), (i_2, j_2), and (i_3, j_3), respectively. Delete the 1's from these three cells and write them in the cells with coordinates (i_1, j_3), (i_2, j_1), and (i_3, j_2). Move ptr_1 to the row of ptr_2.

    (IB1.1) If there is no such a cell R, then swap C_1 and C_2 and move ptr_1 to the row of ptr_2.

  (IB2) If C_2 is a right cell, then

    (IB2.1) If $|col(C_1)| = |col(C_2)|$ then move ptr_1 to the row of ptr_2.

    (IB2.2) If $|col(C_1)| \neq |col(C_2)|$ then do nothing.

(IC) Move ptr_2 to the next row in $\mathcal{R}\backslash\mathcal{R}_2$ above it. Go to (IB).

(ID) Stop.

Suppose M' is a filling obtained by applying the algorithm γ_r to the filling $M \in \mathbf{F}(\mathcal{M}, \mathbf{s}; A)$. Next we show that by applying δ_r^2 and δ_r^1 to M' one obtains the filling M.

Steps (IB′1) and (IB′2) clearly invert the steps (C′2) and (C′1), respectively. Immediately after the borrowing step in γ_r^2, by Lemma 5.2, (b) and (c), and Property 4, we can conclude that borrowing has been performed if and only if there is a right 1-cell in $\mathcal{R}\backslash\mathcal{R}_2$ which forms an NE

chain with a left 1-cell from $\mathcal{R}_2$ and does not form an NE chain with any left cells in $\mathcal{R}_1$. It is clear that step (ID$'$) detects whether there was any borrowing and inverts it if it occurred. So, the first algorithm δ_r^2 is indeed the inverse of γ_r^2.

In δ_r^1, when ptr_2 points to a left 1-cell we need to invert either (B1) or (B2.3). Using Lemma 5.2 (c) one sees that step (IB1) detects exactly which of (B1) and (B2.3) occurred and inverts it. If ptr_2 points to a right cell, it is clear that (IB2.1) and (IB2.2) invert the steps (B2.1) and (B2.2) of γ_r^1. Moreover, if both ptr_1 and ptr_2 point to right 1-cells, and there is no left 1-cell above them, then clearly the algorithm δ_r^1 leaves them unchanged. That is, the algorithm δ_r^1 preserves all rows that are above the rows in $\mathcal{R}_1$. Therefore, we have

$$M \xrightarrow{\gamma_r} M' \xrightarrow{\delta_r^2 + \delta_r^1} M.$$

This implies that $\gamma_r : \mathbf{F}(\mathcal{M}, \mathbf{s}; A) \to \mathbf{F}(\mathcal{M}, \mathbf{s}; A)$ is injective. Since $\mathbf{F}(\mathcal{M}, \mathbf{s}; A)$ is finite, it follows that γ_r is bijective.

References

[1] W.Y.C. Chen, E.Y.P. Deng, R.R.X. Du, R.P. Stanley, C.H. Yan, Crossings and nestings of matchings and partitions, Trans. Amer. Math. Soc. 359 (2007) 1555–1575.

[2] W.Y.C. Chen, I. Gessel, C.H. Yan, A.L.B. Yang, A Major Index for Matchings and Set Partitions, J. Combin. Theory Ser. A 115 (2008) 1069–1076. A longer version containing a bijective proof is on arXiv:math/0701096.

[3] W.Y.C. Chen, S.Y.J. Wu, C.H. Yan, Linked partitions and linked cycles, European J. Combin. 29 (2008) 1408–1426.

[4] A. de Mier, k-noncrossing and k-nonnesting graphs and fillings of Ferrers diagrams, Combinatorica 27 (2007) 699–720.

[5] M. de Sainte-Catherine, Couplages et Pfaffiens en Combinatoire, Physique et Informatique, Ph.D. thesis, University of Bordeaux I, Talence, France, 1993.

[6] D. Foata, On the Netto inversion number of a sequence, Proc. Amer. Math. Soc. 19 (1968) 236–240.

[7] D. Foata, Rearrangements of words, in M. Lothaire, Combinatorics on Words, Cambridge University Press, Cambridge, 1997.

[8] D. Foata, G-N. Han, Fix-Mahonian calculus, I: Two transformations, European J. Combin. 29 (7) (2008) 1721–1732.

[9] I. Gessel, Applications of quasi-symmetric functions and noncommutative symmetric functions in permutation enumeration, talk presented at the Canadian Mathematical Society Winter Meeting 2008, and private communication.

[10] J. Jonsson, Generalized triangulations and diagonal-free subsets of stack polyominoes, J.Combin. Theory Ser. A 112 (2005), no. 1, 117142.

[11] A. Kasraoui, On the symmetry of ascents and descents over 01-fillings of moon polyominoes, arxiv:0807.0914.

[12] A. Kasraoui, J. Zeng, Distribution of crossings, nestings and alignments of two edges in matchings and partitions, Electronic J. of Combin. 13 (2006) #R33.

[13] M. Klazar, On identities concerning the numbers of crossings and nestings of two edges in matchings, SIAM J. Discrete Math. 20 (2006) 960–976.

[14] C. Krattenthaler, Growth diagrams, and increasing and decreasing chains in fillings of Ferrers shapes, Adv. in Appl. Math. 37 (2006) 404–431.

[15] P.A. MacMahon, Two applications of general theorems in combinatory analysis, Proc. London Math. Soc. 15 (1916) 314–321.

[16] S. Poznanović, A bijection between partially directed paths in the symmetric wedge and matchings, preprint, arXiv:0803.4233.

[17] S. Poznanović, C. Yan, Crossings and nestings of two edges in set partitions, SIAM J. Discrete Math. 23 (2009), no. 2, 787-804.

[18] J. Riordan, The distribution of crossings of chords joining pairs of $2n$ points on a circle, Math. Comput.29 (1975) 215–222.

[19] M. Rubey, Increasing and decreasing sequences in fillings of moon polyominoes, arXiv:math.CO/0604140.

[20] M. Rubey, Nestings of Matchings and Permutations and North Steps in PDSAWs , arXiv:0712.2804.

[21] R. Stanley, Enumerative Combinatorics, vol 1, Cambridge University Press, Cambridge, 1997.

[22] J. Touchard, Sur un probléme de configurations et sur les fractions continues, Canadian J. Math. 4 (1952) 2–25.