Two Types of Bit String Generating Functions

Kyle Krueger

Advisor: Professor John Greene

TABLE OF CONTENTS

CHAPTER 1 – Introduction

Enumerative combinatorics is a branch of mathematics that deals with finding, enumerating, and counting patterns. One such type of pattern is called a permutation, which can be described as an ordering of a given set of elements. In classifying and describing permutations, it is often helpful to be able to quantify and measure "orderedness." Essentially, orderedness is the measure of how close a given permutation is to being in ascending order – that is, arranged so that no member is followed by any number smaller than it.

1.1 - The Inversion Number

One such measure in a sequence or bit string is the inversion number. This number counts the number of "swaps" of consecutive positions that need to happen to get a string back into ascending order. That is, for example, in the sequence 125436, there are three inversions. Swapping the 3 and the 5 would result in a properly ordered string, but three adjacent swaps are needed. The inversion number of this sequence is thus 3. In a bit string, since the entries are only either 0 or 1, an inversion is an occurrence of a 0 following (not just immediately following) a 1. The inversion number of a bit string then reflects the number of adjacent swaps it will take to get all zeroes to the left of all ones. One can calculate the inversion number of a bit string simply by counting the number of zeroes that follow each individual 1 and then summing that total. This is because the number of zeroes following a certain one indicates the number of times we must move that one to the right before it is no longer followed by any zeroes.

For an example, consider the bit string 01010011000. We'll start from the right side of the string. The three zeroes contribute nothing to the inversion number, so we move to the

rightmost one, in the eighth position in the string. Three zeroes follow this one, which means it would take three swaps to put the one in the correct position.

01010011**000.**

Moving from right to left, the next one occurs in the seventh position, again contributing three to the total inversion number:

01010011**000.**

Moving left, we reach the one in position four, which is followed by five zeroes:

0101**00**11**000.**

Finally, the leftmost one contributes six to the inversion number:

0**1**01**00**11**000.**

As such, the inversion number for this string is 17, which can be written as $Inv(01010011000) = 17$. This indicates that we have to make 17 swaps of adjacent numbers before this string is properly ordered as 00000001111.

1.2 – The Major Index

Another measure of ordering in a bit string is the major index. This is a count of the location of "descents" in the string. A descent occurs at the spot where a one is immediately followed by a zero. The major index is the sum of the locations of these descents. This means that while the inversion number measures the amount of work needed to properly order a string, the major index instead measures how "broken up" a given string is. While the string 11110000000 has a very high inversion number (28, indeed, the maximum inversion number from a string with four ones and seven zeroes), it only has a major index of 4. This is because the string, while not in ascending order, is still fairly "orderly." Another string with four ones and

seven zeroes is 00010101010. This has an inversion number of only 10, but a major index of 28. The reasoning is clear; while not much work needs to be done to properly order the string, it is very "broken up." That is, very few blocks of similar digits exist.

For another example, the major index of the bit string 01010011000 is calculated by first finding the initial descent (which is, again, a 1 followed immediately by a 0). Notice the location of this descent:

$$\underset{1}{0}\,\underset{\mathbf{2}}{\mathbf{1}}\,\underset{3}{0}\,\underset{4}{1}\,\underset{5}{0}\,\underset{6}{0}\,\underset{7}{1}\,\underset{8}{1}\,\underset{9}{0}\,\underset{10}{0}\,\underset{11}{0}\,.$$

Since the 1 in this descent is located at position 2, we add 2 to the major index.

$$\underset{1}{0}\,\underset{2}{1}\,\underset{3}{0}\,\underset{\mathbf{4}}{\mathbf{1}}\,\underset{5}{0}\,\underset{6}{0}\,\underset{7}{1}\,\underset{8}{1}\,\underset{9}{0}\,\underset{10}{0}\,\underset{11}{0}\,.$$

Notice that the second descent occurs at position 4, so we add 4 to the major index. The 1 located at position 7 is not involved in a descent, so we skip it and move on:

$$\underset{1}{0}\,\underset{2}{1}\,\underset{3}{0}\,\underset{4}{1}\,\underset{5}{0}\,\underset{6}{0}\,\underset{7}{1}\,\underset{\mathbf{8}}{\mathbf{1}}\,\underset{9}{0}\,\underset{10}{0}\,\underset{11}{0}\,.$$

Since this descent comes from the 1 located at position 8, we add 8 to the index. Note that there are no more descents in the bit string, so we can sum the counts, and find that the major index of this string is 14, which is written as $Maj(01010011000) = 14$.

1.3 – Generating Functions for Multisets

In this paper, we will be looking exclusively at bit strings, which can be thought of as multiset permutations.

Definition 1.1 – Multiset

A multiset is a set in which repetition of members is allowed. For example, there is a unique set that has the members 1, 2, and 3. This set is {1,2,3}, which is indistinct from {3,1,2},

as the members are the same, and order does not matter in a set. There are, however, infinitely many examples of multisets that contain those members – some examples are {1,2,3}, {1,1,1,2,3}, {1,1,2,2,3,3}, and so on. These are all considered the same set, but they are distinct as multisets. The bit string from the previous section, 01010011000 is a permutation of the multiset {0,0,0,0,0,0,0,1,1,1,1}.

Both the inversion number and the major index are examples of a special kind of function known as a statistic.

Definition 1.2 – Statistic

Let M be a set comprised of all possible permutations of a given multiset. A statistic is a function that maps the members of M to $\boldsymbol{N}$, the set of all natural numbers. If we represent such a function by $stat$, and we let w represent a single permutation of a multiset, then $stat(w) = n$ where $n \epsilon \boldsymbol{N}$. In the case of the inversion number and major index, the set M need not be restricted to all possible permutations of some given multiset – it can, for example, be extended to include all bit strings.

These statistics allow for the calculation of other interesting measures. The one that we are most interested in for this paper is the calculation of the generating function. The way to find this is similar for both the inversion number and the major index. For our purposes, a generating function is a polynomial related to the orderedness of a set of bit strings.

Definition 1.3 – Generating functions

Given a set S of bit strings, with the strings being represented by w, let $f(w)$ be a statistic on S. Then, define a generating function associated with S by the following:

$$F(S, q) = \sum_{w \in S} q^{f(w)} .$$

This is a polynomial in the variable q. To differentiate between the inversion number and the major index, we can define the generating functions separately depending on the chosen statistic:

$$Inv(S,q) = \sum_{w \in S} q^{Inv(w)} \ , \tag{1}$$

and

$$Maj(S,q) = \sum_{w \in S} q^{Maj(w)} \ . \tag{2}$$

For an example, let $S = \{1010, 11010, 0100110100\}$. Since

$$Inv(1010) = 3 \ , \quad Inv(11010) = 5 \ , \text{and } Inv(0100110100) = 13 \ ,$$

the generating function for S using the inversion number is

$$Inv(S,q) = q^3 + q^5 + q^{13} \ .$$

Similarly,

$$Maj(1010) = 4 \ , \quad Maj(11010) = 6 \ , \text{ and } \quad Maj(0100110100) = 16 \ ,$$

so the generating function using the major index is

$$Maj(S,q) = q^4 + q^6 + q^{16} \ .$$

1.4 – A Special Set of Bit Strings

Consider the set of all strings of a certain length n with k ones and $n-k$ zeroes, denoted $S_{n,k}$. The generating functions of such sets have a few interesting properties. For an example, we could consider the set of strings that have length 9 with 4 ones. There are 126 such strings, populating the set of strings $S_{9,4}$. If we calculate the generating function of this set using the inversion number, we find:

$$Inv(S_{9,4}, q) = 1 + q + 2q^2 + 3q^3 + 5q^4 + 6q^5 + 8q^6 + 9q^7 + 11q^8 + 11q^9 + 12q^{10} + 11q^{11} + 11q^{12} + 9q^{13} + 8q^{14} + 6q^{15} + 5q^{16} + 3q^{17} + 2q^{18} + q^{19} + q^{20}.$$

Note that this also produces an interesting result:

$$Inv(S_{9,4}, 1) = 126 = |S|.$$

This is true for any set S - by substituting 1 in for q, the polynomial will yield the size of the set.

We can also see that the coefficient of each term counts the number of strings in $S_{9,4}$ with inversion numbers equal to the power of its associated term. That is, because the polynomial contains the term $6q^5$, there are 6 strings in the set that have an inversion number of 5. Now, if we calculate the major index generating function for this set, we get:

$$Maj(S_{9,4}, q) = 1 + q + 2q^2 + 3q^3 + 5q^4 + 6q^5 + 8q^6 + 9q^7 + 11q^8 + 11q^9 + 12q^{10} + 11q^{11} + 11q^{12} + 9q^{13} + 8q^{14} + 6q^{15} + 5q^{16} + 3q^{17} + 2q^{18} + q^{19} + q^{20}.$$

Note that $Maj(S_{9,4}, q) = Inv(S_{9,4}, q)$. In fact, if for any n and k, if $S = S_{n,k}$, then $Maj(S, q) = Inv(S, q)$. This will be proven for bit strings in Chapter 2. Because of this equivalence, these two statistics are referred to as "Mahonian statistics" [3, p.53] after Major Percy MacMahon.

Also because of this equivalence, the number of bit strings in S that have inversion number k is equal to the number of strings in S that have major index k. If we again consider $S_{9,4}$, we can see that the number of bit strings having major index 5 and inversion number 5 should be six each. The six strings with an inversion number of 5 are 100000111, 001010011, 000101110, 001001101, 010001011, and 000110101. The six strings having a major index of 5 are then 100100011, 101100001, 000010111, 000110011, 001110001, and 011110000.

1.5 – Generating Functions of Bit String Necklaces

A bit string necklace is a set made up of all possible rotations of a given bit string. For example, the necklace of the bit string 1011100 is generated by rotating the string until it returns to its original form. Such a necklace is made up of 1011100, 0101110, 0010111, 1001011, 1100101, 1110010, and 0111001. The next rotation yields a bit string identical to the original one, so we stop rotating at that point. The original string was of length 7, and there are seven strings in the necklace. The number of elements in a necklace is often the length of the strings making up that necklace. However, it can be shorter. When the necklace has fewer elements than the length of its bit strings, the strings are said to be periodic. The number of terms in the necklace is equal to the length of the period, which must be a divisor of the length of the strings. As an example, the necklace of 010011010011 is 010011010011, 101001101001, 110100110100, 011010011010, 001101001101, and 100110100110, since the next rotation yields the original string. The strings here each have length twelve, and the necklace is of size six.

In this paper, the generating function will be found for the necklace of a given string. For an example, we will calculate the generating function of the necklace generated by 101001010 first using the inversion number, and then using the major index. We calculate the inversion number of the original string:

$$101001010 \rightarrow 12\,.$$

Thus, this string adds the term q^{12} to the polynomial. Continuing forward, we rotate the string to the right so that the trailing 0 now becomes the leading element, and we find:

$$010100101 \rightarrow\ q^{8}\,.$$

We rotate again, calculate the new inversion number, and continue until the string returns to its original state:

$$101010010 \rightarrow q^{13},$$

$$010101001 \rightarrow q^{9},$$

$$101010100 \rightarrow q^{14},$$

$$010101010 \rightarrow q^{10},$$

$$001010101 \rightarrow q^{6},$$

$$100101010 \rightarrow q^{11},$$

and

$$010010101 \rightarrow q^{7}.$$

Notice that the next rotation, 101001010 is back to the original string, so we stop, and combine the q terms to form our generating function for this bit string necklace:

$$q^6 + q^7 + q^8 + q^9 + q^{10} + q^{11} + q^{12} + q^{13} + q^{14}.$$

The method is similar for the major index. We calculate the major index for each individual string in the necklace until we've returned to the first string.

$$101001010 \rightarrow 1 + 3 + 6 + 8 = 18 \rightarrow q^{18},$$

$$010100101 \rightarrow q^{13},$$

$$101010010 \rightarrow q^{17},$$

$$010101001 \rightarrow q^{12},$$

$$101010100 \rightarrow q^{16},$$

$$010101010 \rightarrow q^{20},$$

$$001010101 \rightarrow q^{15},$$

$$100101010 \rightarrow q^{19},$$

and

$$010010101 \rightarrow q^{14}.$$

We can thus form the generating function for this necklace of bit strings using the major index:

$$q^{12} + q^{13} + q^{14} + q^{15} + q^{16} + q^{17} + q^{18} + q^{19} + q^{20}.$$

It is interesting to note at this point that the generating function for a set of bit strings generally depends on whether you're using the inversion number or the major index. However, that does not mean that the two statistics are unrelated.

This project involves studying the properties of $Inv(S, q)$ and $Maj(S, q)$ in the case where S is a necklace of bit strings. However, before that, we utilize the q-binomial coefficient to relate our statistics together.

CHAPTER 2 – q-Series and Statistics on Words

2.1 – The q-Binomial Coefficient and q-Pascal Identities

In combinatorics, special numbers called binomial coefficients are commonly used. These are the integer coefficients that make up Pascal's Triangle, which is a visual representation of the coefficients in binomial expressions of different degrees.

Definition 2.1 – The Binomial Coefficient

The binomial coefficient, written $\binom{n}{k}$ and read as "n choose k," is defined as follows:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)(n-2)\cdots 2\cdot 1}{k(k-1)\cdots 2\cdot 1(n-k)(n-k-1)\cdots 2\cdot 1},$$

which can be simplified to:

$$\binom{n}{k} = \frac{n(n-1)(n-2)\cdots(k+2)(k+1)}{(n-k)(n-k-1)\cdots\cdot 2\cdot 1} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k(k-1)\cdots\cdot 2\cdot 1}. \quad \textbf{(3)}$$

This formula, and many more properties of binomial coefficients can be found in [4, p.23, 5, p.22, 6, p.153, 8, p.187]. It is interesting to note that $\binom{n}{k}$ counts the number of ways to choose k objects from a set of n objects [5, p.22]. This can be thought of as choosing the positions of the ones or the zeroes in a bit string of length n, and therefore, it can be seen that $\binom{n}{k} = |S_{n,k}|$.

Definition 2.2 – The q-Binomial Coefficient

The q-binomial coefficient, which extends from the regular binomial coefficient, is defined as:

$$\binom{n}{k}_q = \frac{(1-q^n)(1-q^{n-1})(1-q^{n-2})\cdots(1-q^2)(1-q)}{(1-q^k)(1-q^{k-1})\cdots(1-q)(1-q^{n-k})(1-q^{n-k-1})\cdots(1-q)}.$$

Once again, we simplify:

$$\binom{n}{k}_q = \frac{(1-q^n)(1-q^{n-1})(1-q^{n-2})\cdots(1-q^{n-k+2})(1-q^{n-k+1})}{(1-q^k)(1-q^{k-1})\cdots(1-q)}. \quad \textbf{(4)}$$

The q-binomial coefficient simplifies to a polynomial, although this is not immediately obvious. It follows from either Theorem 2.2 or Theorem 2.4, or by induction and the q-Pascal identities found in formulas (6) and (7) below. Also, $\lim_{q\to 1}\binom{n}{k}_q = \binom{n}{k}$. This can be seen by applying the fact that $\lim_{q\to 1}\frac{1-q^i}{1-q^j} = \frac{i}{j}$ to (4). As an example, consider $\binom{4}{2}_q$. We have

$$\binom{4}{2}_q = \frac{(1-q^4)(1-q^3)}{(1-q^2)(1-q)} = (1+q^2)(1+q+q^2) = 1+q+2q^2+q^3+q^4,$$

which is a polynomial. Also,

$$\lim_{q\to 1}\binom{4}{2}_q = \lim_{q\to 1}\frac{1-q^4}{1-q^2}\cdot\frac{1-q^3}{1-q} = \frac{4\cdot 3}{2\cdot 1} = \binom{4}{2}.$$

In this way, we see the relation between the original binomial coefficient and the q-binomial coefficient.

Definition 2.3 – Pascal's Identity and the q-Pascal Identities

Pascal's Identity for binomial coefficients states:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}. \tag{5}$$

Extending the analogy of the binomial coefficient to the q-binomial coefficient, we discover two similarly analogous q-Pascal Identities [1, p.35]:

$$\binom{n}{k}_q = \binom{n-1}{k-1}_q + q^k\binom{n-1}{k}_q, \tag{6}$$

and

$$\binom{n}{k}_q = q^{n-k}\binom{n-1}{k-1}_q + \binom{n-1}{k}. \tag{7}$$

Formulas (6) and (7) will be proved in Lemma 2.1.

2.2 – q-Pascal and Bit String Order

One must now wonder what the q-Pascal identities have in common with the inversion number and major index of bit strings. To demonstrate this relationship, I'll first rewrite the Pascal identities a bit to fit the notation that I will be using for the upcoming proofs. Let m represent the number of ones in a given bit string, and let n represent the number of zeroes. That means that any such bit string is of length $m+n$, and we can then write

$$\binom{m+n}{m}_q = \frac{(1-q^{m+n})(1-q^{m+n-1})(1-q^{m+n-2})\cdots(1-q^{n+2})(1-q^{n+1})}{(1-q^m)(1-q^{m-1})\cdots(1-q)}. \qquad \textbf{(8)}$$

In this form, the q-Pascal identities become

$$\binom{m+n}{m}_q = q^m\binom{m+n-1}{m}_q + \binom{m+n-1}{m-1}_q \qquad \textbf{(9)}$$

and

$$\binom{m+n}{m}_q = \binom{m+n-1}{m}_q + q^n\binom{m+n-1}{m-1}_q. \qquad \textbf{(10)}$$

Again, these identities will be proven in Lemma 2.1. Now, to draw the analogy to bit strings, recall that $\binom{m+n}{m}$ counts the set of all bit strings of length $m+n$ with m ones. This can be described as placing $m+n$ positions and then choosing m of them to be ones, with the remaining digits being zeroes. Also, we see that $\binom{m+n}{m} = \binom{m+n}{n}$ [8, p.187].

Pascal's identity is closely related to bit strings, and that relationship can be shown with a combinatorial proof. There are $\binom{m+n}{m}$ bit strings of length $m+n$ with m ones and n zeroes. If the last bit in a given string is a one, deleting it leaves a string with $m-1$ ones and n zeroes, counted by $\binom{m+n-1}{m-1}$. If the trailing bit is a zero, deleting it gives a string with m ones and $n-1$ zeroes, counted by $\binom{m+n-1}{m}$. This yields $\binom{m+n}{m} = \binom{m+n-1}{m} + \binom{m+n-1}{m-1}$, which is a minor twist on equation (5).

For example, $\binom{5}{2} = 10$, so there are 10 bit strings of length 5 that contain 2 ones. If we write these down based on whether the last bit is a zero or a one, we have the 10:

$$11000, 10100, 10010, 01100, 01010, 00110$$

and

$$00011, 00101, 01001, 10001 .$$

When the last bit is deleted from each of these, we get the 6 elements of $S_{4,2}$ from the first list, and the 4 elements of $S_{4,1}$ from the second list.

It has been established that $\binom{n}{k} = |S_{n,k}|$, and thus, $\binom{m+n}{m} = |S_{m+n,m}|$. In the next few sections, we show that $\binom{m+n}{m}_q$ is the generating function for $S_{m+n,m}$ using either the inversion number or the major index.

2.3 – q-Pascal and Inversion Numbers

Lemma 2.1

$$\binom{m+n}{m}_q = \binom{m+n-1}{m-1}_q + q^m \binom{m+n-1}{m}_q ,$$

and

$$\binom{m+n}{m}_q = \binom{m+n-1}{m}_q + q^n \binom{m+n-1}{m-1}_q .$$

Proof

These can be found in [1, p.35]. Let us first show that the first q-Pascal identity works for the q-binomial coefficients. We have:

$$\binom{m+n}{m}_q = \frac{(1-q^{m+n})(1-q^{m+n-1})\cdots(1-q^{n+1})}{(1-q)(1-q^2)\cdots(1-q^m)},$$

$$q^m \binom{m+n-1}{m}_q = q^m \left(\frac{(1-q^{m+n-1})(1-q^{m+n-2}) \cdots (1-q^n)}{(1-q)(1-q^2) \cdots (1-q^m)} \right),$$

$$\binom{m+n-1}{m-1}_q = \frac{(1-q^{m+n-1})(1-q^{m+n-2}) \cdots (1-q^{n+1})}{(1-q)(1-q^2) \cdots (1-q^{m-1})}.$$

We now can find a common denominator when the second and third terms are added:

$$q^m \binom{m+n-1}{m}_q + \binom{m+n-1}{m-1}_q = \frac{q^m(1-q^{m+n-1})(1-q^{m+n-2}) \cdots (1-q^n)}{(1-q)(1-q^2) \cdots (1-q^m)}$$

$$+ \frac{(1-q^{m+n-1})(1-q^{m+n-2}) \cdots (1-q^{n+1})}{(1-q)(1-q^2) \cdots (1-q^{m-1})} \cdot \frac{1-q^m}{1-q^m}$$

$$= \frac{q^m(1-q^{m+n-1})(1-q^{m+n-2}) \cdots (1-q^n) + (1-q^{m+n-1})(1-q^{m+n-2}) \cdots (1-q^{n+1})(1-q^m)}{(1-q)(1-q^2) \cdots (1-q^m)}$$

$$= \frac{(q^m - q^{m+n} + 1 - q^m)(1-q^{m+n-1})(1-q^{m+n-2}) \cdots (1-q^{n+1})}{(1-q)(1-q^2) \cdots (1-q^m)}$$

$$= \frac{(1-q^{m+n})(1-q^{m+n-1})(1-q^{m+n-2}) \cdots (1-q^{n+1})}{(1-q)(1-q^2) \cdots (1-q^m)}$$

$$= \binom{m+n}{m}_q.$$

Thus, the first q-Pascal identity holds for any q-binomial coefficient.

The method used for the second identity is similar. Simply replace m in the bottom of the q-binomial coefficient, and the relation follows from there, following the same steps as the first identity.

Theorem 2.2

The q-binomial coefficient $\binom{m+n}{m}_q$ is the inversion number generating function for all bit strings of length $m+n$ with m ones.

Proof

The proof will follow by induction using the q-Pascal identities.

Base Case:

The result is easy for small *m* and *n.* Consider a row of Pascal's triangle – we'll look at the second row. The coefficients are 1, 2, and 1 – corresponding to the binomial coefficients $\binom{2}{0}$, $\binom{2}{1}$, and $\binom{2}{2}$. The first one, $\binom{2}{0}$, can be seen to count all bit strings of length 2 with no ones – clearly, the only bit string that satisfies this is "00". The inversion number of "00" is 0, meaning that the generating function of the bit string of length 2 with 2 zeroes is 1, and $\binom{2}{0}_q = \frac{1-q}{1-q} = 1$. Next, $\binom{2}{1}$ counts all bit strings of length 2 with 1 one, which are "10" and "01." The inversion numbers of these strings are 1 and 0 respectively, giving a generating function of $1 + q$, and $\binom{2}{1}_q = \frac{1-q^2}{1-q}$, which simplifies to $1 + q$. Finally, $\binom{2}{2}$ counts all bit strings of length 2 with 2 ones, and the only member of this set is "11". The inversion number of this is 0, meaning that this generating function is 1, and $\binom{2}{2}_q = \frac{(1-q^2)(1-q)}{(1-q)(1-q^2)} = 1$. Thus, for these cases, the generating function by way of inversion is equal to the matching q-binomial coefficient.

Induction Case:

Consider again the combinatorial proof of Pascal's identity given on page 12, and consider what happens when we take the inversion numbers of the bit strings involved into account. Assume that the q-binomial coefficient equals the inversion generating function for all bit strings of length $m + n - 1$ with any number of ones. We can, for the purposes of this argument, represent the number of ones in some bit string by the letter k. To write this assumption symbolically, we assume

$$\binom{m+n-1}{k}_q = \sum_w q^{Inv(w)},$$

where, the sum is over all bit strings w with $m+n-1$ digits and k ones.

If we consider all strings of length $m+n$ that have k ones, as before, the string must end in either a one or a zero. If it ends in a one, then the string is of the form $w1$, where $w \epsilon S_{m+n-1,k-1}$, and $Inv(w1) = Inv(w)$. This also implies the following:

$$q^{Inv(w1)} = q^{Inv(w)},$$

and thus,

$$\sum_w q^{Inv(w1)} = \sum_w q^{Inv(w)}.$$

If the string ends in a zero, then the string is of the form $w0$, where $w \epsilon S_{m+n-1,k}$, and $Inv(w0) = Inv(w) + k$. This implies

$$q^{Inv(w0)} = q^{Inv(w)+k} = q^{Inv(w)}q^k,$$

and so,

$$\sum_w q^{Inv(w0)} = \sum_w q^{Inv(w)+k} = \sum_w q^{Inv(w)}q^k = q^k \sum_w q^{Inv(w)}.$$

Indeed, every element of $S_{m+n,k}$ will be of one of these two forms.

By the induction hypothesis,

$$\sum_w q^{Inv(w)} = \sum_{w_1} q^{Inv(w_1 0)} + \sum_{w_2} q^{Inv(w_2 1)}$$

$$= q^k \sum_{w_1} q^{Inv(w_1)} + \sum_{w_2} q^{Inv(w_2)}$$

$$= q^k \binom{m+n-1}{k}_q + \binom{m+n-1}{k-1}_q = \binom{m+n}{k}_q$$

by Lemma 2.1. Setting $k = m$ gives the result. ■

To see this in action, we return to the example from the end of Section 2.2. Notice that the four strings from the second list, 10001, 01001, 00101, and 00011, all have the same inversion number as the strings with the trailing 1 deleted. Also note that the strings from the first list, 11000, 10100, 10010, 01100, 01010, and 00110, all have inversion numbers that are 2 larger than the strings with the trailing 0 deleted. Thus, $Inv(S_{5,2}, q) = Inv(S_{4,1}, q) + q^2 Inv(S_{4,2}) = \binom{4}{1}_q + q^2\binom{4}{2}_q = \binom{5}{2}_q$.

2.4 – q-Pascal and Major Indices

This relationship doesn't stop at the inversion number. As it turns out, the major index generating function also equals the q-binomial coefficient. For the proof of this, a few special tricks are needed to address the latent intricacies of the major index. We therefore begin with a lemma that will be necessary for this proof.

Lemma 2.3

The standard binomial coefficient identity,

$$\binom{k}{k} + \binom{k+1}{k} + \binom{k+2}{k} + \cdots + \binom{n}{k} = \binom{n+1}{k+1}, \tag{11}$$

found in [8, p.226] extends to q-binomial coefficients in the following way:

$$\binom{k}{k}_q + q\binom{k+1}{k}_q + q^2\binom{k+2}{k}_q + \cdots + q^{n-k}\binom{n}{k}_q = \binom{n+1}{k+1}_q, \tag{12}$$

which can be found in [1, p.37].

Proof

Base Case:

When $n = k$, this identity states

$$\binom{k}{k}_q = \binom{k+1}{k+1}_q,$$

which is true, since both are equal to 1.

Induction Case:

Assume that equation (12) is true for some *n* and *k*. Adding $q^{n-k+1}\binom{n+1}{k}_q$ to both sides of the equation, we have

$$\binom{k}{k}_q + q\binom{k+1}{k}_q + q^2\binom{k+2}{k}_q + \cdots + q^{n-k}\binom{n}{k}_q + q^{n-k+1}\binom{n+1}{k}_q$$

$$= \binom{n+1}{k+1}_q + q^{n-k+1}\binom{n+1}{k}_q.$$

By Lemma 2.1,

$$\binom{n+1}{k+1}_q + q^{n-k+1}\binom{n+1}{k}_q = \binom{n+2}{k+1}_q.$$

Thus, if we assume that equation (12) is true for *n*, then the following is also true:

$$\binom{k}{k}_q + q\binom{k+1}{k}_q + q^2\binom{k+2}{k}_q + \cdots + q^{n-k}\binom{n}{k}_q + q^{n-k+1}\binom{n+1}{k}_q = \binom{n+2}{k+1}_q.$$

Thus, equation (12) is also true for $n + 1$. ■

Theorem 2.4

The q-binomial coefficient $\binom{m+n}{m}_q$ is equal to the major index generating function for all bit strings of length $m + n$ with m ones.

To prove Theorem 2.4, we mimic a proof of formula (11). This formula has the following combinatorial explanation. Every bit string in $S_{m+n,m}$ has its rightmost one in some specific

position. If it falls in the final position, then it follows a string from the set $S_{m+n-1,m-1}$. The size of this set is $\binom{m+n-1}{m-1}$. If the final one falls in the second to last position, it is followed by a zero, and the string that precedes it is a member of the set $S_{m+n-2,m-1}$, which is of size $\binom{m+n-2}{m-1}$. This pattern continues until we consider the earliest possible position of the final one, which is position *m*. This means that there are *n* zeroes following the one, and the string preceding it is from $S_{m-1,m-1}$, a set of size 1. This counts the set of all strings of length $m + n$ having *m* ones.

For an example of this proof, consider the case where $m = n = 3$. There are 20 such bit strings in this set. We can group them by where the rightmost one occurs:

$$\{000111, 001011, 010011, 100011, 001101, 010101, 100101, 011001, 101001, 110001\},$$

$$\{001110, 010110, 100110, 011010, 101010, 110010\},$$

$$\{011100, 101100, 110100\},$$

and

$$\{111000\}.$$

Call these sets $S_1, S_2, S_3,$ and S_4, respectively. If we remove the bits in boldface from these sets, we are left with:

$$\{00011, 00101, 01001, 10001, 00110, 01010, 10010, 01100, 10100, 11000\},$$

$$\{0011, 0101, 1001, 0110, 1010, 1100\},$$

$$\{011, 101, 110\},$$

and

$$\{11\}.$$

Call these sets $S_1', S_2', S_3',$ and S_4', respectively. Note that $S_1' = S_{5,2}$, $S_2' = S_{4,2}$, $S_3' = S_{3,2}$, and $S_4' = S_{2,2}$. This shows that

$$\binom{6}{3} = \binom{5}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2}.$$

We now turn our attention to the generating function side of things. Each string in S_1 has the rightmost one in the final position. If we remove this one from each string, resulting in the set S_1', the major index does not change, meaning that the major index generating function of S_1 is the same as that of S_1'. Each string in S_2 has the rightmost one in the fifth position. Removing the final two bits from these strings lowers each major index by 5, meaning that the generating function of S_2 is the same as that of S_2' with an extra q^5 term at the end. This continues similarly for S_3 and S_4. The generating function for S_3 is the same as the one for S_3' with q^4 added to the end, and the generating function for S_4 is the same as the one for S_4' with q^3 added to the end.

From this, we see that

$$Maj(S_{6,3}, q) = Maj(S_{5,2}, q) + q^5 Maj(S_{4,2}, q) + q^4 Maj(S_{3,2}, q) + q^3 Maj(S_{2,2}, q).$$

If we know that $Maj(S_{n,k}, q) = \binom{n}{k}_q$ when $n \leq 5$, then formulas (10) and (12) show that

$$Maj(S_{6,3}, q) = \binom{6}{3}_q.$$

Proceeding as in this example, we now give a proof of Theorem 2.4.

Proof

Base Case:

We begin by considering the entire second row of Pascal's triangle, which is generated by $\binom{2}{0}$, $\binom{2}{1}$, and $\binom{2}{2}$. The base case from here is nearly identical to the one used in the proof of Theorem 2.2.

Inductive Case:

Assume that the q-binomial coefficient equals the major index generating function for all bit strings up to length $m + n - 1$ having k ones. Symbolically, this means:

$$\binom{j}{k}_q = \sum_w q^{Maj(w)}$$

for all j, with $j \leq m + n - 1$. Our sum is over all bit strings w of length j having k ones. We now consider the set of all strings of length $m + n$ and categorize them based on the location of the rightmost one. Notice that a string that ends in a one has the same major index as the string preceding that final one, so it can thus be said that the major index of a bit string ending in a one is the same as the major index of the string created by removing that trailing digit. That is:

$$Maj(w1) = Maj(w),$$

$$q^{Maj(w1)} = q^{Maj(w)},$$

and

$$\sum_w q^{Maj(w1)} = \sum_w q^{Maj(w)}.$$

This means that the generating function for these strings is

$$\binom{m+n-1}{k-1}_q.$$

If the rightmost one is in position $m + n - 1$ so it is followed by a zero, then w has a descent that is not present when the final two bits are removed. Consequently,

$$Maj(w_1 10) = Maj(w_1) + m + n - 1,$$

and

$$\sum_{w_2} q^{Maj(w_2 10)} = \sum_{w_2} q^{Maj(w_2)+m+n-1} = \sum_{w_2} q^{Maj(w_2)} q^{m+n-1} = q^{m+n-1} \binom{m+n-2}{k-1}_q.$$

We continue:

$$Maj(w_3 100) = Maj(w_3) + m + n - 2,$$

so

$$\sum_{w_3} q^{Maj(w_3 100)} = \sum_{w_3} q^{Maj(w_3)+m+n-2} = \sum_{w_3} q^{Maj(w_3)} q^{m+n-2} = q^{m+n-2} \binom{m+n-3}{k-1}_q .$$

This pattern will repeat until the final case, where we have all ones on the left and all zeroes on the right. This yields:

$$Maj(11 \cdots 100 \cdots 0) = k,$$

so

$$q^{Maj(11\cdots100\cdots0)} = q^k \binom{k-1}{k-1}_q = q^k .$$

The major index generating function of the set $S_{m+n,m}$ of bit strings having length $m+n$ and m ones can be written as $Maj(S_{m+n,m}, q)$. Thus, we can write the generating function as follows:

$$Maj(S_{m+n,m}, q) = \binom{m+n-1}{m-1}_q + q^{m+n-1} \binom{m+n-2}{m-1}_q + \cdots + q^m \binom{m-1}{m-1}_q .$$

By factoring out a common term, we get:

$$Maj\big(S_{m+n,m}, q\big) = \binom{m+n-1}{m-1}_q$$

$$+ q^m \left[q^{n-1} \binom{m+n-2}{m-1}_q + q^{n-2} \binom{m+n-3}{m-1}_q + \cdots + \binom{m-1}{m-1}_q \right].$$

Now, with a little bit of rewriting, one can see that the expression within the square brackets is simply a special case of equation (12). This means:

$$q^{n-1} \binom{m+n-2}{m-1}_q + q^{n-2} \binom{m+n-3}{m-1}_q + \cdots + \binom{m-1}{m-1}_q = \binom{m+n-1}{m}_q ,$$

and thus,

$$Maj\big(S_{m+n,m}, q\big) = \binom{m+n-1}{m-1}_q + q^m \binom{m+n-1}{m}_q .$$

This is equation (10), the second q-Pascal identity! Thus, we have

$$Maj(S_{m+n,m}, q) = \binom{m+n}{m}_q .$$

This establishes a combinatorial relationship between the major index generating function of a set $S_{m+n,m}$ and the q-binomial coefficient. ■

Continuing the example of $S_{6,3}$, we have

$$Inv(S_{6,3}, q) = Maj(S_{6,3}, q) = \binom{6}{3}_q$$

$$= \frac{(1-q^6)(1-q^5)(1-q^4)}{(1-q^3)(1-q^2)(1-q)}$$

$$= (1+q^3)(1+q+q^2+q^3+q^4)(1+q^2)$$

$$= 1+q+2q^2+3q^3+3q^4+3q^5+3q^6+2q^7+q^8+q^9 .$$

By Theorem 2.2 and Theorem 2.4, we can find $Inv(S_{6,3}, q)$ or $Maj(S_{6,3}, q)$ without computing the appropriate statistic in all 20 cases. In particular, without examining all possibilities, we know that there are 3 bit strings with 3 ones, 3 zeroes, and inversion number or major index equal to 6. Likewise, there are 2 such strings with inversion number or major index equal to 7, and so on.

Thus far, it has been established that the normal Pascal identity can be extended to the q-binomial case, and that the q-Pascal identities can be explained with combinatorial arguments for the generating functions for both inversion numbers and major indices. Because of the analogy between bit strings and binomial coefficients which was established in Section 2.2, it can be said that the inversion number and major index are counting the same thing when analyzing a set $S_{n,k}$ of bit strings. The upshot of this is that one need not calculate both generating functions of a set of bit strings of length n with k ones, as they will always be equivalent. Moreover, if we can

devise a method to quickly determine the generating function of such a set using one Mahonian statistic, it will predict the polynomial using the other statistic as well.

CHAPTER 3 – Calculating Generating Functions for Necklaces

Any set $S_{m+n,m}$ can be written as the union of disjoint necklaces in a natural way. One need only choose a string from the set, generate the necklace through cyclic rotations, and then choose a string that hasn't yet been used and repeat. For example, the set $S_{7,3}$ can be separated into the following necklaces:

$$\{0000111, 1000011, 1100001, 1110000, 0111000, 0011100, 0001110\},$$

$$\{0100011, 1010001, 1101000, 0110100, 0011010, 0001101, 1000110\},$$

$$\{0010011, 1001001, 1100100, 0110010, 0011001, 1001100, 0100110\},$$

$$\{0001011, 1000101, 1100010, 0110001, 1011000, 0101100, 0010110\},$$

and

$$\{0101001, 1010100, 0101010, 0010101, 1001010, 0100101, 1010010\}.$$

If one finds the generating function of each disjoint necklace, the result is an equation of the form

$$\binom{m+n}{m}_q = f_1(q) + f_2(q) + \cdots .$$

We will denote each necklace by the first element. The major index generating function for the 0000111 necklace is

$$f_1(q) = 1 + q + q^2 + q^3 + q^4 + q^5 + q^6 .$$

The major index generating function for the 0100011 necklace is

$$f_2(q) = q^2 + q^4 + q^5 + q^6 + q^7 + q^8 + q^{10} .$$

If we continue for the last three necklaces, we find

$$f_3(q) = q^3 + q^4 + q^5 + q^6 + q^7 + q^8 + q^9 ,$$

$$f_4(q) = q^3 + q^4 + q^5 + q^6 + q^7 + q^8 + q^9 ,$$

and

$$f_5(q) = q^6 + q^7 + q^8 + q^9 + q^{10} + q^{11} + q^{12} .$$

We now combine these:

$$f_1(q) + f_2(q) + f_3(q) + f_4(q) + f_5(q)$$

$$= 1 + q + 2q^2 + 3q^3 + 4q^4 + 4q^5 + 5q^6 + 4q^7 + 4q^8 + 3q^9 + 2q^{10} + q^{11} + q^{12} .$$

Similarly,

$$\binom{7}{3}_q = \frac{(1-q^7)(1-q^6)(1-q^5)}{(1-q^3)(1-q^2)(1-q)}$$

$$= 1 + q + 2q^2 + 3q^3 + 4q^4 + 4q^5 + 5q^6 + 4q^7 + 4q^8 + 3q^9 + 2q^{10} + q^{11} + q^{12} .$$

Thus,

$$\binom{7}{3}_q = f_1(q) + f_2(q) + f_3(q) + f_4(q) + f_5(q) .$$

What we see here is the internal structure of the q-binomial coefficient with respect to necklaces. In this instance, it matters whether we use the inversion number or the major index, and we shall begin by analyzing some results for the inversion number.

3.1 – Generating Functions for the Inversion Number

In analyzing generating functions that result from both of our Mahonian statistics, certain patterns begin to emerge. To efficiently analyze these patterns, it is often helpful to find a meaningful way to organize the necklaces of bit strings. It is helpful to start with a member of the necklace that begins with a zero and ends with a one. That means that the necklace of the example string used earlier in the paper, 01010011000, could be represented by the string 01001100001. By doing this, we are able to quickly group related terms in the polynomial that facilitate finding a closed form.

Again, the terms of this necklace's inversion polynomial are as follows:

$$01010011000 \to q^{17},$$

$$00101001100 \to q^{13},$$

$$00010100110 \to q^{9},$$

$$00001010011 \to q^{5},$$

$$10000101001 \to q^{12},$$

$$11000010100 \to q^{19},$$

$$01100001010 \to q^{15},$$

$$00110000101 \to q^{11},$$

$$10011000010 \to q^{18},$$

$$01001100001 \to q^{14},$$

and

$$10100110000 \to q^{21}.$$

This polynomial could be written as:

$$(q^{21} + q^{17} + q^{13} + q^{9} + q^{5}) + (q^{12}) + (q^{19} + q^{15} + q^{11}) + (q^{18} + q^{14}). \qquad \textbf{(13)}$$

Terms are grouped to form geometric series. The following is used repeatedly:

$$q^{m} + q^{m+k} + q^{m+2k} + \cdots + q^{m+jk} = \frac{q^{m} - q^{m+(j+1)k}}{1 - q^{k}}.$$

For the expression in (13), we have

$$\frac{q^{5} - q^{25}}{1 - q^{4}} + \frac{q^{12} - q^{16}}{1 - q^{4}} + \frac{q^{11} - q^{23}}{1 - q^{4}} + \frac{q^{14} - q^{22}}{1 - q^{4}}$$

$$= \frac{q^{5} + q^{11} + q^{12} + q^{14} - q^{16} - q^{22} - q^{23} - q^{25}}{1 - q^{4}}$$

$$= \frac{(1 - q^{11})(q^{5} + q^{11} + q^{12} + q^{14})}{1 - q^{4}}.$$

Unfortunately, the Inversion number generating function for this string cannot be simplified much beyond this. There are a few results that simplify neatly, however.

Consider the bit string 100001000. Calculating the generating function for its necklace yields:

$$q^3 + q^4 + q^5 + q^6 + q^7 + q^8 + q^9 + q^{10} + q^{11} .$$

This can then be simplified down to:

$$\frac{q^3(1-q^9)}{1-q} = \frac{q^3(1+q)(1-q^9)}{(1-q^2)} .$$

In this polynomial, there are a few points of interest. The term q^3 in the numerator represents the number of zeroes present in the smaller of the two blocks. The term $(1+q)$ in the numerator can be seen to represent the difference in sizes of the two blocks (as $5-4=1$, and thus we have q^1), and the term $(1-q^9)$ seems to represent the length of the string. The denominator then counts the number of ones in the string. This example can be generalized.

Theorem 3.1

Given a bit string necklace where one member is of the form

$$1\underbrace{0\cdots0}_{m}1\underbrace{0\cdots0}_{n} , m > n ,$$

the inversion number generating function for that necklace is

$$\frac{q^m(1-q^{n+m+2})(1+q^{n-m})}{1-q^2} .$$

Proof

Consider the necklace for $1\underbrace{0\cdots0}_{m}1\underbrace{0\cdots0}_{n}$.

The generating function calculation proceeds thusly:

$$1\underbrace{0\cdots0}_{m}1\underbrace{0\cdots0}_{n} \rightarrow q^{m+2n} ,$$

$$01\underbrace{0\cdots0}_{m}1\underbrace{0\cdots0}_{n-1}\rightarrow q^{m+2n-2}\,,$$

and so on, until

$$\underbrace{0\cdots0}_{n}1\underbrace{0\cdots0}_{m}1\rightarrow q^{m}\,.$$

Then, the pattern switches to:

$$1\underbrace{0\cdots0}_{n}1\underbrace{0\cdots0}_{m}\rightarrow q^{n+2m}\,,$$

$$01\underbrace{0\cdots0}_{n}1\underbrace{0\cdots0}_{m-1}\rightarrow q^{n+2m-2}\,,$$

until

$$\underbrace{0\cdots0}_{m}1\underbrace{0\cdots0}_{n}1\rightarrow q^{n}\,.$$

The final function is then

$$(q^{m+2n}+q^{m+2n-2}+\cdots+q^{m})+(q^{n+2m}+q^{n+2m-2}+\cdots q^{n})\,,$$

which can be rewritten as

$$\frac{(q^{m}-q^{m+2n+2})+(q^{n}-q^{n+2m+2})}{1-q^{2}}.$$

If we assume that $m<n$, (a trivial assumption, as we can simply reorder the string to fit this form if need be) then the following factoring can take place:

$$\frac{q^{m}[(1-q^{2n+2})+(q^{n-m}-q^{n+m+2})]}{1-q^{2}}=\frac{q^{m}(1-q^{n+m+2})(1+q^{n-m})}{1-q^{2}}.$$

The explanations for the different portions of this expression are similar to the example – there is a term describing the length of the string, a term describing the difference in lengths of the two blocks, and a term describing the smaller of the two blocks. ■

Theorem 3.2

The inversion number generating function for the necklace of any string that is of the form

$$\underbrace{0\cdots0}_{n}\underbrace{1\cdots1}_{m},$$

is of the form

$$\frac{(1-q^{m+n})(1-q^{mn})}{(1-q^{m})(1-q^{n})}.$$

Proof

Through some hand work, one can find generating function of such a necklace to be

$$1+q^{n}+q^{2n}+\cdots+q^{mn}+q^{m(n-1)}+q^{m(n-2)}+\cdots+q^{m}.$$

This can be rewritten as

$$\frac{1-q^{mn}}{1-q^{n}}+\frac{q^{m}-q^{m(n+1)}}{1-q^{m}}=\frac{(1-q^{mn})(1-q^{m})+(q^{m}-q^{n+1})(1-q^{n})}{(1-q^{m})(1-q^{n})}$$

$$=\frac{1-q^{m}-q^{mn}+q^{mn+m}+q^{m}-q^{m+n}-q^{mn+m}+q^{mn+m+n}}{(1-q^{m})(1-q^{n})}$$

$$=\frac{1-q^{m+n}-q^{mn}+q^{mn+m+n}}{(1-q^{m})(1-q^{n})}=\frac{(1-q^{m+n})(1-q^{mn})}{(1-q^{m})(1-q^{n})}.$$

In this way, we see that this polynomial has a term describing the length of the string, the product of the block lengths, and the length of each individual block. ■

We now consider general forms that contain more than a single one per block.

Theorem 3.3

The inversion number generating function for a bit string that is of the form

$$011\underbrace{0\cdots0}_{n}1$$

is of the form

$$\frac{q(1-q^{n+4})(1+q^{n+1}+q^{2n-1})}{1-q^3}.$$

Proof

Note that this is a slight modification of the form used in Theorem 3.1. The generating function calculation proceeds as follows:

$$011\underbrace{0\cdots0}_{n}1 \to q^{2n},$$

$$1011\underbrace{0\cdots0}_{n} \to q^{3n+1},$$

$$01011\underbrace{0\cdots0}_{n-1} \to q^{3n-2},$$

$$001011\underbrace{0\cdots0}_{n-2} \to q^{3n-5},$$

and so on, until

$$\underbrace{0\cdots0}_{n-1}10110 \to q^{4},$$

$$\underbrace{0\cdots0}_{n}1011 \to q^{1},$$

$$1\underbrace{0\cdots0}_{n}101 \to q^{n+2},$$

and

$$11\underbrace{0\cdots0}_{n}10 \to q^{2n+3}.$$

Using simplification methods similar to those used for previous functions, this function reduces to

$$\frac{q - q^{3n+4}}{1 - q^3} + q^{n+2} + q^{2n+3} + q^{2n}$$

$$= \frac{q(1 - q^{n+4})(1 + q^{n+1} + q^{2n-1})}{1 - q^3}. \blacksquare$$

We can see this in action with an example. Let $n = 15$. The necklace begins with 0110000000000000001. The generating function for this necklace can be written as follows:

$$q + q^4 + q^7 + q^{10} + q^{13} + q^{16} + q^{17} + q^{19} + q^{22} + q^{25} + q^{28} + q^{30} + q^{31} + q^{33} + q^{34} + q^{37}$$
$$+ q^{40} + q^{43} + q^{46}$$

$$= \frac{q(1 - q^{19})(1 + q^{16} + q^{29})}{1 - q^3}.$$

The $(1 - q^{19})$ portion of this function represents the length of the string. The denominator, $1 - q^3$, represents the number of ones in the string. The leading q counts the minimum inversion number of any string in the necklace. The remaining term in the numerator, however, does not seem to be easily explained, and can be considered to be simply a byproduct of the calculation method.

As the strings used for calculations became more and more complicated, simplification of the general form became increasingly difficult. If we again modify the form used in Theorem 3.3 by adding another one to the first block, we can see how quickly the general form of the inversion generating function will grow.

Theorem 3.4

The inversion number generating function for a bit string that is of the form

$$0111\underbrace{0\cdots0}_{n}1$$

is of the form

$$\frac{q(1-q^{n+5})(1+q^{3n-1}+q^{1+n}+q^{2+2n})}{1-q^4}.$$

Proof

The generating function calculation proceeds as follows:

$$0111\underbrace{0\cdots0}_{n}1 \to q^{3n},$$

$$10111\underbrace{0\cdots0}_{n} \to q^{4n+1},$$

$$010111\underbrace{0\cdots0}_{n-1} \to q^{4n-3},$$

$$0010111\underbrace{0\cdots0}_{n-2} \to q^{4n-7},$$

and so on, until

$$\underbrace{0\cdots0}_{n-1}101110 \to q^{5},$$

$$\underbrace{0\cdots0}_{n}10111 \to q,$$

$$1\underbrace{0\cdots0}_{n}1011 \to q^{n+2},$$

$$11\underbrace{0\cdots0}_{n}101 \to q^{2n+3},$$

and

$$111\underbrace{0\cdots0}_{n}10 \to q^{3n+4}.$$

Once again, using familiar simplification methods, this function reduces to

$$\frac{q - q^{4n+5}}{1 - q^4} + q^{n+2} + q^{2n+3} + q^{3n+4} + q^{3n}$$

$$= \frac{q(1 - q^{n+5})(1 + q^{3n-1} + q^{1+n} + q^{2+2n})}{1 - q^4}. \blacksquare$$

The $(1 - q^{n+5})$ portion represents the total length of the string, and the denominator represents the total number of ones in the string. Once again, however, the remaining information seems difficult to explain. We also see that the final portion of the numerator has four terms, one more than the previous case. This seems to indicate that the generating functions become complicated very quickly when using the inversion number.

A more interesting and more easily simplified case results from using the major index to find the generating function.

3.2 –Major Index Generating Functions

We may begin by calculating the major index generating function for our example necklace from the beginning of section 3.1, starting with the "first" string of the necklace.

$$01001100001 \to q^8,$$

$$10100110000 \to q^{11},$$

$$01010011000 \to q^{14},$$

$$00101001100 \to q^{17},$$

$$00010100110 \to q^{20},$$

$$00001010011 \to q^{12},$$

$$10000101001 \to q^{15},$$

$$11000010100 \to q^{18},$$

$$01100001010 \to q^{21},$$

$$00110000101 \to q^{13},$$

and

$$10011000010 \to q^{16}\,.$$

Enumerating these necklaces for major index calculations is best achieved by breaking the first string down into blocks of ones and zeroes that begin with a zero and end with a one. That means that our example bit string could be described as follows:

$$\underbrace{01}_{1}\,\underbrace{0011}_{2}\,\underbrace{00001}_{3}\,.$$

In this case, we need only describe the length of each individual block – we can represent the length of block i by b_i. In this case, we can describe our example bit string as having $b_1 = 2, b_2 = 4, b_3 = 5$. This enumeration could also describe the necklace generated by the string 01011101111, or by the string 01000100001. While there are many different bit strings with the same set of b's, they all have the same major index! Indeed, this will be true for the entire necklace of all such strings. This will be proven in Lemma 3.2. Now, in order, the major index generating function of this necklace is $(q^8 + q^{11} + q^{14} + q^{17} + q^{20}) + (q^{12} + q^{15} + q^{18} + q^{21}) + (q^{13} + q^{16})$, which reduces to the following:

$$\frac{q^8 - q^{23}}{1 - q^3} + \frac{q^{12} - q^{24}}{1 - q^3} + \frac{q^{13} - q^{19}}{1 - q^3}$$

$$= \frac{q^8 + q^{12} + q^{13} - q^{19} - q^{23} - q^{24}}{1 - q^3}$$

$$= \frac{(1 - q^{11})(q^8 + q^{13} + q^{12})}{1 - q^3}\,.$$

This is already a bit cleaner than the function generated by the inversion number. One can also describe the parts of this expression in terms of the bit string that generated it! The term $(1 - q^{11})$ can be seen to describe the length of the bit strings in this necklace, which is 11. The term $(1 - q^3)$ in the denominator can be seen to describe the number of blocks in the strings, which is

3. The final term, $(q^8 + q^{13} + q^{12})$ is a bit trickier to describe. Notice that we have three blocks, and we also have three separate terms in this part of the expression. Notice also that 8 is equal to twice the length of the first block added to the length of the second block; that is, $2 \cdot 2 + 4 = 8$. Notice also that 13 is equal to twice the length of the second block plus the length of the third block; that is, $2 \cdot 4 + 5 = 13$. Finally, 12 is equal to twice the length of the third block plus the length of the first block; or $2 \cdot 5 + 2 = 12$. As it turns out, this pattern holds true for every possible necklace of bit strings when using the major index. To conceptualize why this happens, we will consider the general case.

3.3 – A General Form for Major Index Generating Functions

To concisely calculate the generating function for any necklace using the major index, we first have to define a new operator. Let $R^j(\vec{x})$ operate by rotating some vector $\vec{x}$ cyclically j times. That is, if we let $\vec{v} = (a, b, c, d)$and $\vec{w} = (e, f, g, h)$, then the operation $\sum_{j=0}^{3} \vec{v} \cdot R^j(\vec{w})$ yields the following:

$$(ae + bf + cg + dh) \text{ for } j = 0\,,$$

$$(af + bg + ch + de) \text{ for } j = 1\,,$$

$$(ag + bh + ce + df) \text{ for } j = 2\,,$$

$$(ah + be + cf + dg) \text{ for } j = 3\,,$$

and thus,

$$\sum_{j=0}^{3} \vec{v} \cdot R^j(\vec{w}) = ae + af + ag + ah + bf + bg + bh + be + cg + ch + ce + cf + dh + de + df + dg\,.$$

Lemma 3.5

When calculating the generating function of the necklace for a bit string that is enumerated in blocks beginning with a zero and ending in a one, the only measure that matters is the length of each block, not the number of ones and zeroes in these blocks.

Proof

For some bit string of the form

$$\underbrace{0\cdots01\cdots1}_{b_1}\underbrace{0\cdots01\cdots1}_{b_2}\cdots\underbrace{0\cdots01\cdots1}_{b_k}, \tag{14}$$

where b_i represents the length of the i-th block, and k represents the total number of blocks, let us look at the first b_k terms in the necklace. We look at two cases to prove this.

Case 1:

If j is less than or equal to the number of ones in the rightmost block, then a rotation of j units results in a string of the form

$$\underbrace{1\cdots1}_{j}\underbrace{0\cdots01\cdots1}_{b_1}\underbrace{0\cdots01\cdots1}_{b_2}\cdots\underbrace{0\cdots01\cdots1}_{b_k-j},$$

and the major index of that string is

$$maj\left(\underbrace{1\cdots1}_{j}\underbrace{0\cdots01\cdots1}_{b_1}\underbrace{0\cdots01\cdots1}_{b_2}\cdots\underbrace{0\cdots01\cdots1}_{b_k-j}\right)$$

$$= j + (j + b_1) + (j + b_1 + b_2) + \cdots + (j + b_1 + b_2 + \cdots + b_{k-1})$$

$$= kj + (k-1)b_1 + (k-2)b_2 + \cdots + b_{k-1}.$$

Case 2:

If j is greater than the number of ones in the rightmost block, then a rotation of j units leaves the string

$$\underbrace{0\cdots01\cdots1}_{j}\underbrace{0\cdots01\cdots1}_{b_1}\underbrace{0\cdots01\cdots1}_{b_2}\cdots\underbrace{0\cdots0}_{b_k-j},$$

which has a major index of

$$maj\left(\underbrace{0\cdots01\cdots1}_{j}\underbrace{0\cdots01\cdots1}_{b_1}\underbrace{0\cdots01\cdots1}_{b_2}\cdots\underbrace{0\cdots0}_{b_k-j}\right)$$

$$= j + (j + b_1) + (j + b_1 + b_2) + \cdots + (j + b_1 + b_2 + \cdots + b_{k-1})$$

$$= kj + (k-1)b_1 + (k-2)b_2 + \cdots + b_{k-1}.$$

Note that the b_k-th element of the necklace brings block k to the front, giving the string

$$\underbrace{0\cdots01\cdots1}_{b_k}\underbrace{0\cdots01\cdots1}_{b_1}\cdots\underbrace{0\cdots01\cdots1}_{b_{k-1}}.$$

Thus, for $b_k \le j < b_k + b_{k-1}$ for the original string, the calculation is the same as having $0 \le j < b_{k-1}$ in this new string. Again, the major index for each of these strings depends on only the b's. We continue this way for the rest of the blocks. ■

Theorem 3.6

Given some bit string necklace, generated cyclically from the general form in (14), the major index generating function for that necklace is equal to

$$\frac{\left(1-q^{l(w)}\right)\left(\sum_{j=0}^{k-1} q^{\vec{v}\cdot R^j(\vec{b})}\right)}{1-q^k}, \qquad \textbf{(15)}$$

where $l(w)$ is the length of the string, $\vec{v} = \{k-1, k-2, \cdots 2,1,0\}$, and $\vec{b} = \{b_1, b_2, \cdots, b_k\}$.

Proof

By Lemma 3.5, without loss of generality, we may assume w has the form

$$\underbrace{0\cdots01}_{b_1}\underbrace{0\cdots01}_{b_2}\cdots\underbrace{0\cdots01}_{b_k}.$$

As in the previous lemma, we see that for $0 \leq j < b_k$, making j rotations of the string yields an associated q term of $q^{kj+(k-1)b_1+(k-2)b_2+\cdots+b_{k-1}}$. The terms for $0 \leq j < b_k$ form a single geometric series as in the example in Section 3.1. The last term in this group, found by setting $j = b_k - 1$, is $q^{kj+(k-1)b_1+(k-2)b_2+\cdots+b_{k-1}}$. We can factor $q^{(k-1)b_1+(k-2)b_2+\cdots+b_{k-1}}$ from each term in this set to obtain

$$q^{(k-1)b_1+(k-2)b_2+\cdots+b_{k-1}}\left(1 + q^k + q^{2k} + \cdots + q^{(b_k-1)k}\right).$$

By utilizing the geometric series, this becomes

$$q^{(k-1)b_1+(k-2)b_2+\cdots+b_{k-1}}\,\frac{1-q^{kb_k}}{1-q^k}.$$

After b_k rotations, the string looks like

$$\underbrace{0\cdots01}_{b_k}\,\underbrace{0\cdots01}_{b_1}\cdots\underbrace{0\cdots01}_{b_{k-1}}.$$

We now begin to work with this new set of strings. As before, these strings generate their own unique terms, grouped together to form a geometric series. The pattern continues from the previous set, and the following expression results:

$$q^{(k-1)b_k+(k-2)b_1+\cdots+2b_{k-3}+b_{k-2}}\,\frac{1-q^{kb_{k-1}}}{1-q^k}.$$

Likewise, the third set of strings form the expression

$$q^{(k-1)b_{k-1}+(k-2)b_k+\cdots+2b_{k-4}+b_{k-3}}\,\frac{1-q^{kb_{k-2}}}{1-q^k}.$$

This continues until the final set of terms yield the following:

$$q^{(k-1)b_2+(k-2)b_3+\cdots+2b_{k-1}+b_k}\,\frac{1-q^{kb_1}}{1-q^k}.$$

These closed forms are summed together to form the final polynomial. We will arrange the numerator in a slightly different way. If we group the positive numerator term of the first segment with the negative term from the final segment, the result is

$$\frac{q^{(k-1)b_1+(k-2)b_2+\cdots+2b_{k-2}+b_{k-1}} - q^{kb_1+(k-1)b_2+(k-2)b_3+\cdots+2b_{k-1}+b_k}}{1-q^k}$$

$$= q^{(k-1)b_1+(k-2)b_2+\cdots+2b_{k-2}+b_{k-1}} \frac{1-q^{b_1+b_2+\cdots+b_k}}{1-q^k}.$$

Similarly, if we group the positive numerator term from the second segment with the negative term from the first segment, the result is

$$\frac{q^{(k-1)b_k+(k-2)b_1+\cdots+2b_{k-3}+b_{k-2}} - q^{kb_k+(k-1)b_1+(k-2)b_2+\cdots+2b_{k-2}+b_{k-1}}}{1-q^k}$$

$$= q^{(k-1)b_k+(k-2)b_1+\cdots+2b_{k-3}+b_{k-2}} \frac{1-q^{b_1+b_2+\cdots+b_k}}{1-q^k}.$$

This pattern continues until all strings have been used, and the final generating function is

$$\frac{(1-q^{b_1+b_2+\cdots+b_k})(q^{(k-1)b_1+(k-2)b_2+\cdots+2b_{k-2}+b_{k-1}} + q^{(k-1)b_k+(k-2)b_1+\cdots+2b_{k-3}+b_{k-2}} + \cdots + q^{(k-1)b_2+(k-2)b_3+\cdots+2b_{k-1}+b_k})}{1-q^k}$$

Using the previously defined operator R^j, we can condense our notation to something more manageable. For some bit string w with length $l(w)$ comprised of k blocks of ones and zeroes that begin with a zero and end with a one, with each subsequent block's length represented by b_i, let $\vec{v} = \{k-1, k-2, \cdots 2, 1, 0\}$ and $\vec{b} = \{b_1, b_2, \cdots, b_k\}$. This would mean that $\vec{v} \cdot R^0(\vec{b}) = (k-1)b_1 + (k-2)b_2 + \cdots + 2b_{k-2} + b_{k-1}$, $\vec{v} \cdot R^1(\vec{b}) = (k-1)b_k + (k-2)b_1 + \cdots + 2b_{k-3} + b_{k-2}$, and so on. In this way, we can simplify our generating function to equal

$$\frac{\left(1-q^{l(w)}\right)\left(\sum_{j=0}^{k-1} q^{\vec{v}\cdot R^j(\vec{b})}\right)}{1-q^k}.$$

In this expression, we have a term that represents the length of the string, the number of blocks in the string, and a rotating sum that counts the number of times each block is counted in each separate group of terms. ■

This form is considerably more concise than the previous one, and given that one has the proper definition of the cyclic rotation operator R^j, this expression equals the major index generating function for any bit string of any length.

We thus consider a few examples. First recall the string used for the example in section 3.2, 01001100001. This has $k = 3$ blocks, so $l(w) = 11,\ \vec{v} = \{2, 1, 0\}$ and $\vec{b} = \{2, 4, 5\}$. Our generating function then factors to the following:

$$\frac{(1-q^{11})(q^{2\cdot 2+4\cdot 1+5\cdot 0} + q^{5\cdot 2+2\cdot 1+4\cdot 0} + q^{4\cdot 2+5\cdot 1+2\cdot 0})}{1-q^3}$$

$$= \frac{(1-q^{11})(q^8 + q^{12} + q^{13})}{1-q^3}.$$

This is identical to the result found at the end of section 3.2! It seems to work well – and it works for much larger bit strings that would require too much work by hand or too much processor time. Consider now the string 011010101101001001000010111100011000. To calculate the generating function for its necklace by hand would be tedious, but by analyzing its parts, we streamline the process. In this situation, we must first reorganize the string into our standard form. Placing the trailing zeroes at the beginning of the string, we are now left with 000011010101101001001000010111100011. We now have $k = 10$ blocks, with $l(w) = 36$, $\vec{v} = \{9, 8, 7, 6, 5, 4, 3, 2, 1, 0\}$ and $\vec{b} = \{6, 2, 2, 3, 2, 3, 3, 5, 5, 5\}$. The resulting polynomial will then be:

$$\frac{(1-q^{36})(q^{148}+q^{162}+q^{176}+q^{190}+q^{184}+q^{178}+q^{162}+q^{156}+q^{140}+q^{124})}{1-q^{10}}.$$

The reader may check this result if desired – the Mathematica code in the appendix will assist in this.

Formula (15) now allows for the quick calculation of any bit string necklace's generating function. Formerly, a string of period 100 would need 100 very complex calculations to find the generating function. With this formula, one needs only one dot product of two vectors per block, as well as a measure of the string's length, and a measure of the number of blocks. Additionally, it is not necessary to rotate the bit string itself at all – the entire calculation can be done from the "basic" string.

CHAPTER 4 – Further Work

4.1 – Inversion Calculations

As mentioned in Section 3.1, my results for the generating functions of inversion numbers were mostly incomplete. The generating function calculations appeared to become very complicated as the strings became more complex. This doesn't, however, imply that there isn't an easier way to enumerate the strings, or a more straightforward relationship between the string and its generating function. Further research could certainly be done on such a topic – being able to swiftly calculate the inversion number generating function is well within the realm of possibility.

One must also be aware that all the work discussed in this paper was done with bit strings and the binomial coefficients. Other multiset permutations could easily be considered, as well as their relationship to the multinomial coefficients. An interesting investigation could be done into the similarities between bit string generating functions and those of more complicated multisets – it is entirely possible that the formula for finding the major index generating function would apply to other multisets. One could also investigate the inversion generating functions for other multisets – it would be very interesting to see whether the calculation becomes even more complex as the sets grow.

A clear upshot of formula (15) in Chapter 3 is that it allows for much more efficient computer calculation of bit string generating functions. A possible task for someone with the proper skills would be to make such a computer program as streamlined as possible. The obvious method is to have the dot product of a static vector and a rotating vector generate the terms in the function. However, it may be possible to simplify such a calculation even further to make this task as lightweight as possible for a computer.

REFERENCES

[1] Andrews, George E. "Restricted Partitions and Permutations." *The Theory of Partitions.* Reading, MA: Addison-Wesley Pub., Advanced Book Program, 1976. 33-53.

[2] Andrews, George E., Richard Askey, and Ranjan Roy. *Special Functions*. Cambridge, UK: Cambridge UP, 1999.

[3] Bona, Miklos. *Combinatorics of Permutations.* Boca Raton: Chapman & Hall/CRC, 2004.

[4] Cameron, Peter J. *Combinatorics: Topics, Techniques, Algorithms*. Cambridge: Cambridge UP, 1994.

[5] Cohen, Daniel I. *Basic Techniques of Combinatorial Theory*. New York: Wiley, 1978.

[6] Graham, Ronald Lewis, Donald Ervin Knuth, and Oren Patashnik. *Concrete Mathematics*. Reading, MA: Addison-Wesley, 1994.

[7] MacMahon, Percy Alexander. *Combinatory Analysis.* Vol. 2. Cambridge: Univ. Pr., 1915.

[8] Tucker, Alan. *Applied Combinatorics.* 5th ed. Hoboken, NJ: Wiley, 2007.

APPENDIX

The following code was used for calculating the generating functions of bit string necklaces by way of inversion number and major index. Wolfram Mathematica 7 was used to write and execute this code.

```
SingleInversionNecklace[x_] := (workingbit = x;
  inversionarray = {};
  necklace = 0;
  For[k = 0, k < Length[workingbit], k++,
   inversionnumber = 0;
   For[i = 0, i < Length[workingbit], i++,
    If[workingbit[[i]] == 1,
      For[l = i, l < Length[workingbit] + 1, l++,
       If[workingbit[[l]] == 0, inversionnumber++, Indeterminate]
       ], Indeterminate];
    ];
   AppendTo[inversionarray, inversionnumber];
   If[RotateRight[workingbit] == x,
    Break[],
    workingbit = RotateRight[workingbit];
    ];
   ];
  For[m = 1, m < Length[inversionarray] + 1, m++,
    necklace = necklace + q^inversionarray[[m]];]
   Print["Length of this bit string = ", Length[x]];
  Print[inversionarray];
  Print[necklace];
  Print[Factor[necklace]])
```

```
SingleMajorNecklace[x_] := (workingbit = x;
  majorarray = {};
  necklace = 0;
  For[k = 0, k < Length[workingbit], k++,
   majorindex = 0;
   For[i = 0, i < Length[workingbit], i++,
    If[workingbit[[i]] == 1,
      If[workingbit[[i + 1]] == 0, majorindex = majorindex + i,
       Indeterminate]
      , Indeterminate];
    ];
   AppendTo[majorarray, majorindex];
   If[RotateRight[workingbit] == x,
    Break[],
    workingbit = RotateRight[workingbit];
    ];
   ];
  For[m = 1, m < Length[majorarray] + 1, m++,
    necklace = necklace + q^majorarray[[m]];]
   Print["Length of this bit string = ", Length[x]];
  Print[majorarray];
  Print[necklace];
  Print[Factor[necklace]])
```