Kernel method and system of functional equations

Toufik Mansour\textsuperscript{a,}\textsuperscript{*}, Chunwei Song\textsuperscript{b}

\textsuperscript{a} Department of Mathematics, University of Haifa, 31905 Haifa, Israel
\textsuperscript{b} School of Mathematical Sciences, LMAM, Peking University, Beijing 100871, PR China

\textbf{A R T I C L E  I N F O}

Article history:
Received 18 October 2007
Received in revised form 9 April 2008

MSC:
05A15
05A19
39B05

Keywords:
Kernel method
Generating functions
Descents
Signed permutations

\textbf{A B S T R A C T}

Introduced by Knuth and subsequently developed by Banderier et al., Prodinger, and others, the kernel method is a powerful tool for solving power series equations in the form of $F(z, t) = A(z, t)f(z_0, t) + B(z, t)$ and several variations. Recently, Hou and Mansour [Q.-H. Hou, T. Mansour, Kernel Method and Linear Recurrence System, J. Comput. Appl. Math. (2007), (in press)] presented a systematic method to solve equation systems of two variables $F(z, t) = A(z, t)f(z_0, t) + B(z, t)$, where $A$ is a matrix, and $F$ and $B$ are vectors of rational functions in $z$ and $t$. In this paper we generalize this method to another type of rational function matrices, i.e., systems of functional equations. Since the types of equation systems we are interested in arise frequently in various enumeration questions via generating functions, our tool is quite useful in solving enumeration problems. To illustrate this, we provide several applications, namely the recurrence relations with two indices, and counting descents in signed permutations.

\textcopyright 2008 Elsevier B.V. All rights reserved.

1. Introduction

A lot of enumeration problems can be solved by introducing a bivariate generating function $F(z, t)$ that satisfies

\begin{equation}
F(z, t) = A(z, t)f(z_0, t) + B(z, t),
\end{equation}

where $z_0$ is a specific number, usually equal to either 0 or 1. Knuth [13] first introduced this powerful tool to solve such equations, which was later turned into a methodology called the \textit{kernel method} [2]. In the sense of generating functions, we deal with formal power series and do not worry about convergence domains. Thus this solution is for formal power series. A collection of examples are provided in [16].

Over the past few decades, the kernel method has been widely used [2,5] and has had various generalizations, such as the obstinate form [3–6,9,10]. For instance, in [10] several examples are illustrated where the following functional equation

\begin{equation}
F(z, t) = A(z, t)f(1, t) + B(z, t)f(1, zt) + C(z, t),
\end{equation}

plays a critical role in solving various combinatorial enumeration problems, with $A(z, t), B(z, t), C(z, t)$ being rational functions in $z$ and $t$. It turns out that in order to solve (1.2) we need to rewrite it as

\begin{equation}
K(z, t)f(z, t) = P(z, t)f(1, t) + Q(z, t)f(1, zt) + R(z, t),
\end{equation}

where $K(z, t), P(z, t), Q(z, t), R(z, t)$ are polynomials in $z$ and $t$. Then, by assuming that there is a small branch (around $(0, 0)$) $z = z_0(t)$ such that $K(z_0(t), t) = 0$, we obtain

\begin{equation}
P(z_0(t), t)f(1, t) + Q(z_0(t), t)f(1, z_0(t)) + R(z_0(t), t) = 0.
\end{equation}
In the case \( P(z_0(t), t) \neq 0 \) we get that

\[
F(1, t) = \frac{C(z_0(t), t)}{A(z_0(t), t)} - \frac{B(z_0(t), t)}{A(z_0(t), t)} F(1, t_0(t)),
\]

and by applying this equation infinite number of times, we obtain that

\[
F(1, t) = -\sum_{j=0}^{\infty} \frac{(-1)^j \left( C(z_0(\rho^j(t)), \rho^j(t)) - \rho^j(t) \right)}{A(z_0(\rho^j(t)), \rho^j(t))} \prod_{i=0}^{j-1} \frac{B(z_0(\rho^i(t)), \rho^i(t))}{A(z_0(\rho^i(t)), \rho^i(t))},
\]

where \( \rho(t) = t_0(t) \) and \( \rho^j = \rho \circ \rho \circ \cdots \circ \rho \) (j times).

One might get trapped into thinking that (1.2) could be solved by using iterations from the very beginning, which is not the case. For instance, consider the equation

\[
F(z, t) = \frac{t}{(1-z)(t-z)} F(1, t) - \frac{zt}{(1-z)(t-z)} F(1, zt) + 1 - \frac{t + t^2}{t - z}.
\]

Since \( z = 1 \) is a pole, we must rewrite it as

\[
(1-z)(t-z) F(z, t) = t F(1, t) - z t F(1, zt) + (1-z)(t-z) - (1-z)(t + t^2),
\]

and study the small branch \( z = t \).

In [12], a systematic method is presented to solve equation systems of two variables of the type (1.3) with \( Q(z, t) = 0 \), where \( P \) is a matrix, and \( F \) and \( R \) are vectors of rational functions in \( z \) and \( t \).

Here in this paper, we extend the kernel method to general equation systems of the form

\[
K(z, t) F(z, t) = A(z, t) G(t) + B(z, t) G(zt) + C(z, t),
\]

where \( K(z, t), A(z, t) \) and \( B(z, t) \) are \( n \times n \) matrices of rational functions in \( z \) and \( t \), \( C(z, t) \) is a column vector of rational functions in \( z \) and \( t \), and both \( F(z, t) \) and \( G(t) \) are unknown column vectors. In real applications, we often let \( G(t) = F(1, t) \).

This paper is organized as follows. In Section 2, we describe the principles of the kernel method to solve equation systems of the form (1.4). Our aim is to find formal power series solutions \( F(z, t) \) and \( G(t) \). Indeed it is sufficient to find all formal power series solutions \( G(t) \) of (1.4).

In Section 3, we provide a relatively small system of equations to illustrate the power of our method. We consider the solution of several recurrence relations and the generating function for the number of signed permutations of length \( n \) according to several statistics.

2. The method

2.1. General problem

The kernel method is a powerful tool in solving equations of generating functions. The standard model deals with the case of a functional equation of the form

\[
K(z, t) F(z, t) = A(z, t) G(t) + C(z, t),
\]

where \( F(z, t) \) and \( G(t) \) are unknown functions. Assume that there is a small branch \( z = z_0(t) \) such that \( K(z_0(t), t) = 0 \). Then

\[
G(t) = -C(z_0(t), t)/A(z_0(t), t).
\]

This implies that

\[
F(z, t) = \frac{-A(z, t) C(z_0(t), t)/A(z_0(t), t) + C(z, t)}{K(z, t)}.
\]

Meanwhile, in some enumeration problems, we encounter the following functional equation

\[
K(z, t) F(z, t) = A(z, t) G(t) + B(z, t) G(zt) + C(z, t),
\]

where \( F(z, t) \) and \( G(t) \) are both unknown functions. Again, assume that there is a small branch \( z = z_0(t) \) such that \( K(z_0(t), t) = 0 \). Then

\[
G(t) = -\frac{C(z_0(t), t)}{A(z_0(t), t)} - \frac{B(z_0(t), t)}{A(z_0(t), t)} G(t_0(t)),
\]

and by applying this equation infinite number of times, we obtain that

\[
G(t) = -\sum_{j=0}^{\infty} \frac{(-1)^j \left( C(z_0(\rho^j(t)), \rho^j(t)) - \rho^j(t) \right)}{A(z_0(\rho^j(t)), \rho^j(t))} \prod_{i=0}^{j-1} \frac{B(z_0(\rho^i(t)), \rho^i(t))}{A(z_0(\rho^i(t)), \rho^i(t))},
\]

where \( \rho(t) = t_0(t) \) and \( \rho^j = \rho \circ \rho \circ \cdots \circ \rho \) (j times).
As evidenced in the next section, it will be great if we can deal with the following equation system, which is of more general use.

\[ K(z, t)F(z, t) = A(z, t)G(t) + B(z, t)G(zt) + C(z, t), \]

where \( K(z, t) = (K_{ij}(z, t)) \), \( A(z, t) = (A_{ij}(z, t)) \) and \( B(z, t) = (B_{ij}(z, t)) \) are \( n \times n \) matrices, \( C(z, t) = (C_{ij}(z, t)) \) is a column vector, and both \( F(z, t) = (F_{i}(z, t)) \) and \( G(t) = (G_{i}(t)) \) are unknown column vectors. We also assume that all the entries of \( K, A, B, C \) are rational functions and \( K(z, t) \) is invertible as a rational matrix.

Next we proceed to provide an algorithmic solution of (2.1).  

2.2. Solution

In order to present the solution for (2.1) we introduce a lemma.

Lemma 2.1. Suppose \( G(t) \) satisfies

\[ G(t) = P(t) + \sum_{j=1}^{n} Q^{(j)}(t)G(\rho_{j}(t)), \]

where \( P(t), G(t) \) are column vectors and \( Q^{(j)} \) is an \( n \times n \) matrix. Then \( G(t) \) is given by

\[ \sum_{j=0}^{n} \sum_{i_{1}, i_{2}, \ldots, i_{j} \in [n]} P(\rho_{i_{1}} \circ \cdots \circ \rho_{i_{j}})(t) \left( \prod_{k=1}^{j} Q^{(k)}(\rho_{i_{k+1}} \circ \cdots \circ \rho_{i_{j}}(t)) \right). \]

Proof. Let

\[ H(t) = \sum_{j=0}^{n} \sum_{i_{1}, i_{2}, \ldots, i_{j} \in [n]} \tilde{P}_{i_{1}, \ldots, i_{j}}(t) \prod_{k=1}^{j} \tilde{Q}^{(k)}_{i_{k+1}, \ldots, i_{j}}(t), \]

where \( \tilde{P}_{i_{1}, \ldots, i_{j}}(t) = P(\rho_{i_{1}} \circ \cdots \circ \rho_{i_{j}}(t)) \) and \( \tilde{Q}^{(k)}_{i_{k+1}, \ldots, i_{j}}(t) = Q^{(k)}(\rho_{i_{k+1}} \circ \cdots \circ \rho_{i_{j}}(t)). \) Then \( P(t) + \sum_{j=1}^{n} Q^{(j)}(t)H(\rho_{j}(t)) \) equals

\[ P(t) + \sum_{i=1}^{n} Q^{(i)}(t) \sum_{j=0}^{n} \sum_{i_{1}, i_{2}, \ldots, i_{j} \in [n]} \tilde{P}_{i_{1}, \ldots, i_{j}}(t) \prod_{k=1}^{j} \tilde{Q}^{(k)}_{i_{k+1}, \ldots, i_{j}}(t) \]

\[ = P(t) + \sum_{i=1}^{n} \sum_{j=0}^{n} \sum_{i_{1}, i_{2}, \ldots, i_{j+1} \in [n]} \tilde{P}_{i_{1}, \ldots, i_{j+1}}(t)Q^{(j+1)}_{i_{j+2}, \ldots, i_{j+1}}(t) \prod_{k=1}^{j} \tilde{Q}^{(k)}_{i_{k+1}, \ldots, i_{j}}(t) \]

\[ = P(t) + \sum_{j=1}^{n} \sum_{i_{1}, i_{2}, \ldots, i_{j} \in [n]} \tilde{P}_{i_{1}, \ldots, i_{j}}(t) \prod_{k=1}^{j} \tilde{Q}^{(k)}_{i_{k+1}, \ldots, i_{j}}(t) \]

\[ = H(t), \]

completing the proof. \( \square \)

Now we are ready to describe our algorithm for finding an explicit formula for \( G(t) \) that satisfies (2.1):

Step 1. Multiply both sides of (2.1) by \( K^{-1}(z, t) \) so that the equation becomes

\[ F(z, t) = A^{(1)}(z, t)G(t) + B^{(1)}(z, t)G(zt) + C^{(1)}(z, t). \]

Let \( d(z, t) \) be the common denominator of the entries of

\[ A^{(1)}, A^{(2)}, \ldots, A^{(n)}, B^{(1)}, B^{(2)}, \ldots, B^{(n)}, C^{(1)} \]

Multiply both sides by the diagonal matrix

\[ D(z, t) = \text{diag}(d_{1}(z, t), d_{2}(z, t), \ldots, d_{n}(z, t)), \]

and the equation becomes

\[ D(z, t)F(z, t) = A^{(2)}(z, t)G(t) + B^{(2)}(z, t)G(zt) + C^{(2)}(z, t), \]

where the entries of \( D, A^{(2)}, B^{(2)} \) and \( C^{(2)} \) are all polynomials in \( z \) and \( t \).
Step 2. Let \( u_i(t) \in C^{in}(t) \) be a root of the polynomial \( d_i(z, t) \), for all \( i = 1, 2, \ldots, n \). Then the \( i \)th equation of (2.2) gives
\[
\sum_{j=1}^{n} A_{ij}^{(2)}(u_i(t), t)G_j(t) + \sum_{j=1}^{n} B_{ij}^{(2)}(u_i(t), t)G_j(tu_i(t)) + C^{(2)}(u_i(t), t) = 0,
\]
for all \( i = 1, 2, \ldots, n \).

Step 3. Suppose that the \( n \times n \) matrix \( (A_{ij}^{(2)})_{1 \leq i,j \leq n} \) is invertible as a matrix in \( C^{in}(t) \). Then (2.3) can be written as
\[
G(t) = P(t) + \sum_{j=1}^{n} Q^{(0)}(t)G(tu_j(t)),
\]
where \( P(t) \) is a column vector and \( Q^{(0)} \) is an \( n \times n \) matrix, both having rational entries in \( u_1(t), u_2(t), \ldots, u_n(t) \).

Step 3. Applying (2.4) infinite number of times we will get an explicit formula for the column vector \( G(t) \) as described in Lemma 2.1 with \( \rho_j(t) = tu_j(t), j = 1, 2, \ldots, n \).

We remark that there is no restriction on choosing the zero \( z = u_i(t) \in C^{in}(t) \) s.t. \( d_i(z, t) = 0 \). Thus for any given set of zeros \( \{u_1(t), \ldots, u_n(t)\} \) with \( d_i(u_i(t), t) = 0, i = 1, 2, \ldots, n \), column vectors of power series \( G(t) \) and \( F(z, t) \) can always be found to satisfy (2.1). Also, note that while our method gives all the solutions, sometimes the infinite sum obtained may not be convergent. In that case, the result should be viewed as a formal power series. In the next section we give three examples to illustrate how to apply our method effectively.

3. Applications

Now we provide some applications of our method.

3.1. Recurrence relations with two indices

A sequence with \( k \) indices is a function \( a : A^k \rightarrow B \), and denoted by \( \{a_{n_1, \ldots, n_k}\}_{n_1, \ldots, n_k \in A} \) or \( \{a_n\}_{n \in A^k} \), where \( A \subseteq N \). The element \( a_n \) of a sequence \( \{a_n\}_{n \in A} \) is called the \( n \)th term, and the vector \( n \) of integers is the sequence vector of indices. A recurrence relation is an equation which defines a sequence recursively, that is, each term of the sequence is defined as a function of the preceding terms, together with the initial conditions. The initial conditions are necessary to ensure an uniquely defined sequence. The aim of this subsection is to use our algorithm to solve several types of recurrence relations with two indices. (Standard types of recurrence relations and the kernel method may be found in [14,15,18].)

Example 3.1. Let \( a_n \) be a sequence defined by \( a_1 = 1 \) and \( a_n = \sum_{i=1}^{n} a_n(i) \), where \( a_1(1) = 1, a_n(n) = a_{n-1} \) and \( a_n(i) = a_{n-1}(1) + \cdots + a_{n-1}(i-1) + qa_{n-1} \) for all \( i = 1, 2, \ldots, n-1 \).

In order to get an explicit formula for the generating function \( f(x) = \sum_{n \geq 1} a_n x^n \), we consider \( F_n(v) = \sum_{i \geq 1} a_n(i)v^{i-1} \). For \( n \geq 2 \),
\[
F_n(v) = q(v^0 + v^1 + \cdots + v^{n-1})F_{n-1}(1) + \sum_{i=1}^{n} v^{i-1} \sum_{j=1}^{n-1} a_{n-1}(j)
\]
\[= q \left( \frac{1 - v^n}{1 - v} \right) F_{n-1}(1) + \sum_{j=1}^{n-1} \frac{v^j - v^{n-1}}{1 - v} a_{n-1}(j)
\]
\[= q \left( \frac{1 - v^n}{1 - v} \right) F_{n-1}(1) + \frac{v}{1 - v} (F_{n-1}(v) - v^{n-2}F_{n-1}(1)).
\]

Let \( F(x, v) = \sum_{n \geq 1} F_n(v)x^n \). Then writing the above recurrence relation in terms of the generating function and using the initial condition \( F_1(v) = 1 \) we obtain that
\[F(x, v) = x + \frac{xq}{1 - v} F(x, 1) - \frac{xq}{1 - v} F(xv, 1) + \frac{x}{1 - v} (xF(x, v) - F(xv, 1)),
\]
which is equivalent to
\[
\left( 1 - \frac{xv}{1 - v} \right) F(x, v) = x + \frac{xq}{1 - v} F(x, 1) - \frac{x(1 + vq)}{1 - v} F(xv, 1).
\]

Our algorithm with \( v = v_0(x) = \frac{1}{1+x} \) gives
\[0 = x + (1 + x)qF(x, 1) - (1 + x + q)F(x/(1 + x), 1).
\]

This implies that \( F(x, 1) = \frac{x/(1-x)}{1/(1-x)+q} + \frac{q/(1-x)}{1/(1-x)+q} F(x/(1 - x), 1). \) Hence,
\[f(x) = \frac{x}{1 + (1-x)q} + \sum_{i \geq 1} \frac{xq}{(1 - x)(1 - (i+1)x)} \prod_{j=0}^{i} \left( 1 - \frac{q}{1 - y - x} + q \right).
\]
Example 3.2. Let \( a_n = \sum_{i=1}^{n} a_i(i) \) and \( b_n = \sum_{i=1}^{n} b_i(i) \) be two sequences defined as follows:
\[
\begin{align*}
    a_n(i) &= a_{n-1}(1) + \cdots + a_{n-1}(i-1) + qa_{n-1}, \quad i = 1, 2, \ldots, n-1 \\
    b_n(i) &= b_{n-1}(1) + \cdots + b_{n-1}(i-1) + qa_{n-1}, \quad i = 1, 2, \ldots, n-1 \\
    a_n(n) &= \pi_n \\
    b_n(n) &= \pi_{n-1},
\end{align*}
\]
with the initial conditions \( a_1 = 1 \) and \( b_1 = 0 \). We aim to solve the system and find explicit formulas for the generating functions \( f(x) = \sum_{n \geq 1} a_n x^n \) and \( g(x) = \sum_{n \geq 1} b_n x^n \). Define \( F_n(v) = \sum_{i=1}^{n} a_i(v)^{i-1} \) and \( G_n(v) = \sum_{i=1}^{n} b_i(v)^{i-1} \). Using similar arguments as in the example above, we obtain that for all \( n \geq 2 \),
\[
\begin{align*}
    F_n(v) &= \frac{1}{1-v} (vF_{n-1}(v) - v^2F_{n-1}(1)) + q \frac{1-v^{n-1}}{1-v} G_{n-1}(1) \\
    G_n(v) &= \frac{1}{1-v} (vG_{n-1}(v) - v^2G_{n-1}(1)) + q \frac{1-v^{n-1}}{1-v} F_{n-1}(1),
\end{align*}
\]
with the initial conditions \( F_1(v) = 1 \) and \( G_1(v) = 0 \). Let \( f(x, v) = \sum_{n \geq 1} F_n(v)x^n \) and \( g(x, v) = \sum_{n \geq 1} G_n(v)x^n \). We shall get
\[
\begin{align*}
    F(x, v) &= x + \frac{xv}{1-v} (F(x, v) - F(x, 1)) + \frac{xq}{1-v} (G(x, 1) - G(x, 1)) \\
    G(x, v) &= \frac{xv}{1-v} (G(x, v) - G(x, 1)) + \frac{xq}{1-v} (F(x, 1) - F(x, 1)),
\end{align*}
\]
which is equivalent to
\[
\begin{align*}
    \left(1 - \frac{xv}{1-x} \right) F(x, v) G(x, 1) = (x + \frac{xq}{1-v} \left[ \begin{array}{c} 0 \\ 1 \end{array} \right]) \left( F(x, 1) G(x, 1) \right) - \frac{x}{1-v} \left[ \begin{array}{c} q \\ v \end{array} \right] \left( F(x, 1) G(x, 1) \right).
\end{align*}
\]
Our method (substituting \( v = \frac{1}{1+x} \)) gives that
\[
q \left[ \begin{array}{c} 0 \\ 1 \end{array} \right] F(x, 1) G(x, 1) + \frac{x}{1+x} \left[ \begin{array}{c} 1 \\ 0 \end{array} \right] = \frac{1}{1+x} \left( x + \frac{xq}{1-v} \left[ \begin{array}{c} 0 \\ 1 \end{array} \right] \right) \left( F(x, 1) G(x, 1) \right) - \frac{x}{1-v} \left[ \begin{array}{c} q \\ v \end{array} \right] \left( F(x, 1) G(x, 1) \right).
\]
Now replacing \( x \) by \( \frac{x}{1-x} \) we obtain
\[
\left[ \begin{array}{c} 0 \\ q \end{array} \right] \left( F \left( \frac{x}{1-x}, \frac{1}{1-x} \right) \right) + \frac{1}{1-x} \left[ \begin{array}{c} 1 \\ 0 \end{array} \right] = \left( q \right) \left( F(x, 1) G(x, 1) \right),
\]
and hence
\[
F(x, 1) G(x, 1) = q \left( \frac{x}{1-x} \right)^2 - q \left( x - 1 - q \right) \left( \frac{x}{1-x} \right)^2 + \frac{x}{(1-x)^2} - q \left( 1 - x \right).
\]
Again, this implies that the vector \( \left( F(x, 1) G(x, 1) \right) \) is given as a vector power series by
\[
\sum_{j=0}^{\infty} \frac{(-q)^j x}{(1-x)^2 - q} \prod_{i=0}^{j-1} A(x/(1-jx)) \left( 1 - \frac{x}{1-ix} \right),
\]
where \( A(x,j) = \left( \frac{x}{(x-1), q} \right) \).

3.2. Counting descents in signed permutations

More generally, let \( b_n \) be the set of signed permutations \( \pi = \pi_1 \pi_2 \cdots \pi_n \) of length \( n \) (permutations of the set \( \{1, 2, \ldots, n\} \) where each letter can be either negative or positive). We say that a signed permutation \( \pi \in b_n \) has a monochromatic descent at \( i \), if \( |\pi_i| > |\pi_{i+1}| \) and both \( \pi_i \) and \( \pi_{i+1} \) are either negative or positive letters, for \( 1 \leq i \leq n-1 \). The number of monochromatic descents in \( \pi \) is denoted by \( md(\pi) \). For example, the signed permutation \( 2(-1)43(-6)(-5) \) has two
monochromatic descents, namely at 3 (for $|4| > |3|$) and at 5 (for $| - 6| > | - 5|$). We say a signed permutation $\pi \in B_n$ has a rise at $i$ if $|\pi_i| < |\pi_{i+1}|$. For example, in the above example $\pi$ has one rise, namely at 2. The number of rises in $\pi$ is denoted by $\text{rise}(\pi)$. Here we are interested in finding an explicit formula for the generating function for the number of signed permutations of length $n$ according to the number of monochromatic descents (for $B_n$ and various statistics imposed, see [17,11,18,7] and references therein). That is, we study

$$F(x) = \sum_{n \geq 1} \sum_{\pi \in B_n} x^n q^{md(\pi)} p^{\text{rise}(\pi)}.$$ 

Let $f_n(q,i_1, \ldots, i_s)$ be the generating function for the number of signed permutations $\pi$ of length $n$ according to the number of monochromatic descents such that $\pi_i = i_j, j = 1, 2, \ldots, s$. By definition it is not hard to check that

$$f_n(p, q | i) = \sum_{i=1}^{n-1} (f_n(p, q | ij) + f_n(p, q | i-j)) + p \sum_{i=1}^{n-1} (f_n(p, q | ij) + f_n(p, q | i-j))$$

and

$$f_n(p, q | -i) = \sum_{i=1}^{n-1} (f_n(p, q | -ij) + f_n(p, q | -i-j)) + p \sum_{i=1}^{n-1} (f_n(p, q | -ij) + f_n(p, q | -i-j)).$$

Introducing $g_n(p, q | i) = f_n(p, q | i) + f_n(p, q | -i)$, we find

$$g_n(p, q | i) = (1 + q) \sum_{i=1}^{n-1} g_{n-1}(p, q | ij) + 2p \sum_{i=1}^{n-1} g_{n-1}(p, q | j).$$

Let $G_n(p, q, v) = \sum_{i=1}^{n-1} g_n(p, q | i)v^{i-1}$. The above recurrence relation can be rewritten in terms of $G_n(p, q, v)$ as

$$G_n(p, q, v) = \frac{(1 + q)v}{1-v} (G_{n-1}(p, q, v) - v^{n-1} G_{n-1}(p, q, 1) + \frac{2p}{1-v} (G_{n-1}(p, q, 1) - v G_{n-1}(p, q, v)).$$

Moreover, define $G(x, p, q, v) = \sum_{n \geq 1} G_n(p, q, v)x^n$. Multiplying the above recurrence relation by $x^n$ and summing over all possible values of $n \geq 1$ and using the initial condition $G_1(p, q, v) = 2$, we get

$$G(x, p, q, v) = 2x + \frac{(1 + q)vx}{1-v} (G(x, p, q, v) - G(xv, p, q, 1)) + \frac{2xp}{1-v} (G(x, p, q, 1) - v G(x, p, q, v)).$$

Hence,

$$\left( 1 - \frac{(1 + q)vx}{1-v} + \frac{2vx}{1-v} \right) G(x, p, q, v) = 2x - \frac{(1 + q)vx}{1-v} G(xv, p, q, 1) + \frac{2xp}{1-v} G(x, p, q, 1).$$

Our method for $v = \frac{1}{1+(q-1)x}$ gives that

$$2x + \frac{1+q}{q-1} G(x/(1+(q-1)x), p, q, 1) + \frac{2p(1+(q-1)x)}{q-1} G(x, p, q, 1) = 0.$$ 

Substituting $\frac{x}{1-(q-1)x}$ for $x$, we obtain that the generating function $F(x, p, q) = G(x, p, q, 1)$ satisfies

$$F(x, p, q) = 2 \frac{(q-1)x}{(q+1)(1-(q-1)x)} + \frac{2p}{(q+1)(1-(q-1)x)} f\left( \frac{x}{1-(q-1)x}, p, q \right).$$

Applying this equation infinite number of times, we have the following result.

**Theorem 3.3.** The generating function $F(x, p, q)$ for the number of signed permutations of length $n$ according to the number of monochromatic descents and number of rises is given by

$$\sum_{j \geq 1} \frac{2^j p^{-1}(q-1)x}{(q+1)(1-(j-1)(q-1)x)(1-j(q-1)x)}.$$
References