The Method of Coefficients

Donatella Merlini, Renzo Sprugnoli, Maria Cecilia Verri

1 INTRODUCTION.

In their book *Mathematics for the Analysis of Algorithms*, Greene and Knuth [9, p. 7] show how the sum

$$S_m = \sum_{k=0}^{m} \binom{m}{k} \left(-\frac{1}{2}\right)^k \binom{2k}{k} \tag{1.1}$$

can be found in closed form by means of certain transformations on generating functions and the extraction of coefficients (see the end of section 3) suggested by Ira Gessel, and they comment: "He [Gessel] attributes this elegant technique, the 'method of coefficients,' to G. P. Egorychev." Actually, in his book *Integral Representation and Computation of Combinatorial Sums*, Egorychev [4] (see [3] for the Russian edition) deals with the representation of combinatorial expressions in terms of integrals and uses the notation $\mathrm{res}_t L(t)$ (in the Russian edition $\mathrm{Coef}_t L(t)$) for the *residue* of the formal Laurent series $L(t)$ (i.e., the coefficient of t^{-1} in $L(t)$ or in the present, widely accepted notation, $[t^{-1}]L(t)$). The residue notation has been widely used because of the "change of variables" formula $[t^{-1}]g(t) = [t^{-1}]g(f(t))f'(t)$, where $g(t)$ is a formal Laurent series and $f(t)$ is a formal power series having $f(0) = 0$ and $f'(0) \neq 0$. It appears that this result is equivalent to the Lagrange inversion formula (see rule (K6) and section 5) but is more compact and easy to remember. The change of variables formula was first proved by Jacobi [14]; a discussion with application can be found in [7, p. 15]. Egorychev gives an equivalent formula [4, p. 16]. With the residue notation, the coefficient of t^n in $L(t)$ is written $\mathrm{res}_t t^{-n-1}L(t)$, and this is equivalent to $[t^{-1}]t^{-n-1}L(t) = [t^n]L(t)$, which is certainly more direct.

Egorychev's method is especially elegant when he does not complicate proofs with the use of integral representations. An interesting example [4, p. 28] is his derivation of the Grosswald identity,

$$\sum_{k=0}^{n-r} (-2)^{-k} \binom{n}{r+k} \binom{n+r+k}{k} = (-1)^{\frac{n-1}{2}} 2^{r-n} \binom{n}{\frac{n-r}{2}}, \tag{1.2}$$

which we establish in section 3.

The aim of this paper is to present the method of coefficients. From our perspective it can be roughly described in the following way: we suppose that we have to prove some identity or to evaluate some expression (i.e., to find a closed or asymptotic formula for the expression). Instead of operating directly on the identity or on the expression we look for the generating functions related to the quantities involved in the problem. We manipulate these functions in order to arrive at a single expression, which we obtain by extracting the coefficient of the relevant power t^n. This gives the solution.

Generating functions have emerged as one of the most popular approaches to combinatorial problems, above all to problems arising in the analysis of algorithms (see, for example, D. E. Knuth [15] and R. Sedgewick and Ph. Flajolet [22]). A clear exposition of this concept is given in three books: namely, those of I. P. Goulden and D. M. Jackson [7], R. Stanley [26], and H.

S. Wilf [28]. In particular, Wilf develops the *snake oil method*, which is closely related to the method of coefficients.

Somewhat less familiar are the "coefficient of" functionals, whose notation has become standard in the form $[t^n]f(t)$ only in the last ten to fifteen years (the reader is referred to the criticism of Egorychev [4, p. 41]). The interpretation of $[t^n]f(t)$ has also created problems, about which the interested reader can see Knuth's paper [16].

What we wish to show is how a simple set of rules taken from Egorychev [4, sec. 1.2] or [5] (see also [8, Table 320]) can provide us with a way of solving a number of combinatorial problems that lend themselves to expressions in terms of generating functions. We start with very simple examples and go on to treat more complex cases, trying to convince the reader of the effectiveness of the method of coefficients. Consequently, while this paper does not contain any really new results, it is rich in examples "seen from a slightly different point of view." We hope that this small change of perspective will be instructive to the general reader and be of special interest to anybody involved in combinatorics and in the analysis of algorithms.

2 COEFFICIENT EXTRACTION.

We denote by $\mathbf{F}$ any field of characteristics 0; in practice, we consider mainly the fields $\mathbf{R}$ and $\mathbf{C}$ of real and complex numbers. Let $\mathcal{F} = \mathbf{F}[\![t]\!]$ be the ring of *formal power series* in the indeterminate t with coefficients in $\mathbf{F}$. If $f(t) = \sum_{k=0}^{\infty} f_k t^k$ belongs to $\mathcal{F}$ and r is the minimum integer for which $f_r \neq 0$, then r is called the *order* of $f(t)$. The set of formal power series having order r is denoted by $\mathcal{F}_r$. In particular, $\mathcal{F}_0$ is the set of invertible formal power series, that is, series $f(t)$ for which a series $f^{-1}(t)$ exists in $\mathcal{F}$ such that $f(t)f^{-1}(t) = 1$. The ring $\mathcal{F}$ is an integral domain and therefore its quotient field is well defined. It is denoted by $\mathcal{L}$, and its elements are called *formal Laurent series.* The reader is referred to [7] or [11] for a complete treatment of formal power and formal Laurent series. Finally, we introduce an infinite number of linear functionals $[t^n] : \mathcal{F} \to \mathbf{F}$ $(n = 0, 1, 2, \ldots)$ that are defined by their behaviour on the monomials t^k:

$$[t^n]t^k = \delta_{n,k},$$

where $\delta_{n,k}$ is the Kronecker delta. From this definition we have:

$$[t^n]f(t) = [t^n]\sum_{k=0}^{\infty} f_k t^k = \sum_{k=0}^{\infty}[t^n]f_k t^k = \sum_{k=0}^{\infty} f_k [t^n]t^k = \sum_{k=0}^{\infty} f_k \delta_{n,k} = f_n$$

(i.e., $[t^n]f(t)$ equals the coefficient of t^n in $f(t)$). Because of this, the $[t^n]$ are called the *"coefficient of" functionals.*

As described in the introduction, the functionals $[t^n]$ are very important in the present approach to solving combinatorial problems or problems arising in the analysis of algorithms. We could establish a series of their properties, from which the "method of coefficients" derives. However, we choose to follow a more direct approach, as proposed by Egorychev [4]. We use six basic properties of the functionals $[t^n]$, which we state without proof. All except (K6) are straightforward consequences of the definitions of basic operations with formal power series. The consequences of these rules constitute the "method of coefficients." If $f(t)$ and $g(t)$ are formal power series, then the following statements hold:

K1 (linearity)	$[t^n](\alpha f(t) + \beta g(t))$	$=$	$\alpha[t^n]f(t) + \beta[t^n]g(t)$
K2 (shifting)	$[t^n]tf(t)$	$=$	$[t^{n-1}]f(t)$
K3 (differentiation)	$[t^n]f'(t)$	$=$	$(n+1)[t^{n+1}]f(t)$
K4 (convolution)	$[t^n]f(t)g(t)$	$=$	$\sum_{k=0}^{n}\left([y^k]f(y)\right)[t^{n-k}]g(t)$
K5 (composition)	$[t^n]f(g(t))$	$=$	$\sum_{k=0}^{\infty}([y^k]f(y))[t^n]g(t)^k$
K6 (inversion)	$[t^n]\overline{f}(t)^k$	$=$	$\frac{k}{n}[t^{n-k}]\left(\frac{t}{f(t)}\right)^n$

In $(K1)$ α and β are arbitrary constants from $\mathbf{F}$. In $(K4)$ and $(K5)$ the indeterminate y is used only to distinguish the action of the functionals on different formal power series. In $(K5)$ the composition is only possible when $g(0) = 0$ or $f(t)$ is a polynomial, so the sum is actually finite. In $(K6)$, $\overline{f}(t)$ denotes the *compositional inverse* of $f(t)$, that is, the formal power series such that $f(\overline{f}(t)) = \overline{f}(f(t)) = t$. It is known that a formal power series $f(t)$ has a compositional inverse if and only if $f(t)$ belongs to $\mathcal{F}_1$. Finally, we state explicitly the *principle of identity*, a very important rule that is often applied without mention: the formal power series $f(t)$ and $g(t)$ are equal if and only if $[t^n]f(t) = [t^n]g(t)$ holds for $n = 0, 1, 2, \ldots$.

Some points require more lengthy comments. The property of shifting can be easily generalized to $[t^n]t^k f(t) = [t^{n-k}]f(t)$ and also to negative powers: $[t^n]f(t)/t^k = [t^{n+k}]f(t)$. The property of differentiation for $n = -1$ gives $[t^{-1}]f'(t) = 0$, a well-known and important situation. Finally, rule $(K3)$ can be written in the form

$$[t^n]f(t) = \frac{1}{n}[t^{n-1}]f'(t). \tag{2.1}$$

Certain power series are used with great frequency in what follows:

- the exponential series: $e^t = \exp(t) = \sum_{k=0}^{\infty} \frac{1}{k!}t^k$
- the logarithm series: $\log \frac{1}{1-t} = \sum_{k=1}^{\infty} \frac{1}{k}t^k$
- the binomial series: $(1+t)^r = \sum_{k=0}^{\infty} \binom{r}{k} t^k$

Rule (K5) leads to the more general formula (known as *Newton's rule*)

$$(1+\alpha t)^r = \sum_{n=0}^{\infty} \binom{r}{n} \alpha^n t^n$$

or, equivalently,

$$[t^n](1+\alpha t)^r = \binom{r}{n}\alpha^n.$$

We remark that when $r = -1$ we thus obtain

$$[t^n]\frac{1}{1-\alpha t} = \binom{-1}{n}(-\alpha)^n = \binom{1+n-1}{n}\alpha^n = \alpha^n.$$

In the sequel we also use two familiar properties of binomial coefficients:

$$\binom{-r}{n} = \binom{r+n-1}{n}(-1)^n, \qquad \binom{r}{n}\binom{n}{k} = \binom{r}{k}\binom{r-k}{n-k}.$$

The following result will be called the *partial sum theorem*:

Theorem 2.1 (Partial Sum Theorem). *If $f(t)$ belongs to $\mathcal{F}$, then*

$$[t^n]\frac{f(t)}{1-t} = \sum_{k=0}^{n} [t^k]f(t) = f_0 + f_1 + \cdots + f_n.$$

Proof: By applying the previous observation when $\alpha = 1$ we conclude on the basis of (K4) that

$$[t^n]\frac{f(t)}{1-t} = \sum_{k=0}^{n} \left([t^k]f(t)\right)\left([y^{n-k}]\frac{1}{1-y}\right) = \sum_{k=0}^{n} [t^k]f(t).$$

■

Theorem 2.1 allows us to establish a number of important properties. For example, we can easily find the sum of a geometric progression $1 + \alpha + \alpha^2 + \cdots + \alpha^n$. Since, as observed before, $\alpha^k = [t^k](1 - \alpha t)^{-1}$, we have

$$\sum_{k=0}^{n} \alpha^k = [t^n]\frac{1}{1-t}\,\frac{1}{1-\alpha t} = [t^n]\frac{1}{\alpha - 1}\left(\frac{\alpha}{1-\alpha t} - \frac{1}{1-t}\right) = \frac{\alpha^{n+1}-1}{\alpha - 1}.$$

We are now ready to prove a classical result in binary searching (see, for example, [15] and [22]). Suppose that we have a sorted table T with n keys: $T = (k_1, k_2, \ldots, k_n)$ with $k_1 < k_2 < \cdots < k_n$. Given a key k in T, we wish to find the index j for which $k = k_j$ (successful searching). A binary search consists in comparing k against the median element k_r, with $r = \lfloor (n+1)/2 \rfloor$. (Here $\lfloor x \rfloor$ signifies the greatest integer not exceeding x.) If $k = k_r$ we are done; otherwise if $k < k_r$ we proceed by using the subtable $T' = (k_1, \ldots, k_{r-1})$, while if $k > k_r$ we proceed by using the subtable $T'' = (k_{r+1}, \ldots, k_n)$. We are interested in finding the average number b_n of comparisons necessary to conclude the search for any k in T. There is only one element that can be found with a single comparison, the median element in T. There are two elements that can be found with two comparisons, the median elements in T' and T''. Four elements can be found with three comparisons and so on up to a maximum of $1 + \lfloor \log_2 n \rfloor$ comparisons. Therefore, summing the number of comparisons corresponding to each of the n keys, we have

$$b_n = \frac{B_n}{n} = \frac{1}{n}\left(1 + 2 + 2 + 3 + 3 + 3 + 3 + \cdots + (1 + \lfloor \log_2 n \rfloor)\right).$$

In order to compute the sum, we consider the infinite sequence $L = (0, 1, 2, 2, 3, 3, 3, 3, 4, \ldots)$ obtained by summing sequences $(0, 1, 1, 1, \ldots)$, $(0, 0, 1, 1, 1, \ldots)$, $(0, 0, 0, 0, 1, 1, \ldots)$, and so on, the kth sequence starting with 2^k zeroes. This means that the generating function of L is:

$$L(t) = \frac{t}{1-t} + \frac{t^2}{1-t} + \frac{t^4}{1-t} + \cdots = \sum_{k=0}^{\infty} \frac{t^{2^k}}{1-t}.$$

The total number B_n of comparisons for T is therefore given by the partial sum theorem (Theorem 2.1), and we can proceed as follows:

$$B_n = [t^n]\frac{1}{1-t}\sum_{k=0}^{\infty}\frac{t^{2^k}}{1-t} = [t^n]\sum_{k=0}^{\infty}\frac{t^{2^k}}{(1-t)^2} = \sum_{k=0}^{\infty}[t^{n-2^k}]\frac{1}{(1-t)^2} = \sum_{k=0}^{\lfloor \log_2 n\rfloor}[t^{n-2^k}]\frac{1}{(1-t)^2}$$

(for k greater than $\lfloor \log_2 n\rfloor$ the coefficients are zero). Finally,

$$B_n = \sum_{k=0}^{\lfloor \log_2 n\rfloor}\binom{n-2^k+1}{n-2^k} = \sum_{k=0}^{\lfloor \log_2 n\rfloor}(n+1-2^k) = (n+1)(\lfloor \log_2 n\rfloor+1) - \sum_{k=0}^{\lfloor \log_2 n\rfloor}2^k =$$

$$= (n+1)\lfloor \log_2 n\rfloor + n - 2^{\lfloor \log_2 n\rfloor+1} + 2,$$

where we have used the formula for the sum of a geometric progression.

As another example, we now introduce a recurrence arising in the analysis of quicksort (a fast method to sort a list of keys) or of the internal pathlength of a binary tree (see [15] and [22]). For quicksort the average number C_n of key comparisons satisfies

$$C_{n+1} = n+2+\frac{2}{n+1}\sum_{k=0}^{n}C_k$$

or

$$(n+1)C_{n+1} = (n+1)(n+2)+2\sum_{k=0}^{n}C_n,$$

with initial condition $C_0 = 0$. We can transform this recurrence into a relation concerning the generating function $C(t) = \sum_{n\geq 0}C_n t^n$ by multiplying everything by t^n and then summing over all n ($n = 0, 1, 2, \ldots$). The term $(n+1)C_{n+1}$ is transformed into $C'(t)$ by rule (K3) and the sum $\sum_{k=0}^{n}C_k$ into $C(t)/(1-t)$ by the partial sum theorem. Moreover, by Newton's rule, we have

$$[t^n]\frac{1}{(1-t)^3} = \binom{-3}{n}(-1)^n = \binom{3+n-1}{n} = \binom{n+2}{n} = \frac{(n+1)(n+2)}{2}.$$

Consequently, the recurrence corresponds to the differential equation

$$C'(t) = \frac{2}{(1-t)^3}+\frac{2}{1-t}C(t)$$

whose solution is

$$C(t) = \frac{2}{(1-t)^2}\ln\frac{1}{1-t}.$$

In order to find an explicit formula for C_n we have to extract the coefficient $[t^n]C(t)$. By using formula (2.1) we obtain

$$[t^n]\ln\frac{1}{1-t} = \frac{1}{n}[t^{n-1}]\frac{1}{1-t} = \frac{1}{n}$$

for positive n, while the coefficient is 0 for $n = 0$. From the partial sum theorem we infer that

$$[t^n]\frac{1}{1-t}\ln\frac{1}{1-t} = 0+1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{n} = H_n,$$

where H_n is the nth harmonic number. Differentiating this expression, we find by (K3) that

$$[t^n]\frac{1}{1-t}\ln\frac{1}{1-t} = \frac{1}{n}[t^{n-1}]\left(\frac{1}{(1-t)^2}\ln\frac{1}{1-t}+\frac{1}{(1-t)^2}\right).$$

Here we know the left-hand side and the last term, so we obtain

$$[t^n]\frac{1}{(1-t)^2}\ln\frac{1}{1-t} = (n+1)H_{n+1} - (n+1) = (n+1)(H_{n+1}-1). \tag{2.2}$$

We conclude that

$$C_n = 2(n+1)(H_{n+1}-1).$$

3 CONVOLUTION AND COMPOSITION.

We have applied the rule of convolution in the proof of Theorem 2.1. Another classical application is in counting binary trees according to the number of their nodes. A binary tree is either empty or it consists in a node (called the *root*) to which are appended two binary trees (called the *left* and *right* subtrees). In a binary tree with $n+1$ nodes, the left and right subtrees contain k and $n-k$ nodes, respectively ($k = 0, 1, 2, \ldots, n$). If C_{n+1} denotes the number of binary trees with $n+1$ nodes, we immediately find the recurrence $C_{n+1} = \sum_{k=0}^{n} C_k C_{n-k}$. If $C(t) = \sum_{k\geq 0} C_k t^k$ denotes the generating function of the sequence, we can observe by rule (K2) that

$$[t^{n+1}]C(t) = [t^n]\frac{C(t)-C(0)}{t} = [t^n]\frac{C(t)-1}{t},$$

since only the empty tree has zero nodes. Therefore we have

$$[t^n]\frac{C(t)-1}{t} = [t^n]C(t)^2$$

or

$$C(t) = 1 + tC(t)^2,$$

where we have applied the convolution rule and the principle of identity to pass from a coefficient to a formal power series identity. By solving this functional equation (taking into account the initial condition $C(0) = C_0 = 1$) we arrive at

$$C(t) = \frac{1-\sqrt{1-4t}}{2t}.$$

In order to obtain an explicit formula for C_n we extract the nth coefficient:

$$[t^n]C(t) = [t^{n+1}]\frac{1-\sqrt{1-4t}}{2} = 0 - \frac{1}{2}[t^{n+1}]\sqrt{1-4t} = -\frac{1}{2}\binom{1/2}{n+1}(-4)^{n+1} =$$

$$= -\frac{1}{2}\cdot\frac{(-1)^n}{4^{n+1}(2n+1)}\binom{2n+2}{n+1}(-4)^{n+1} = \frac{1}{2}\frac{2n+2)(2n+1)}{(2n+1)(n+1)^2}\binom{2n}{n} = \frac{1}{n+1}\binom{2n}{n}.$$

We have therefore proved that binary trees are counted by the well-known *Catalan numbers* (see, for example, R. Stanley [27, p. 173]).

Sometimes, in order to apply the convolution rule some manipulation is necessary. For example, consider the sum

$$S_{m,n} = \sum_{k=0}^{n}\binom{k-m-1}{m-2}\binom{n-k+m-1}{m-1}$$

taken from the paper by C. Ó'Dúnlaing and L. Erickson [18]. The summation variable k is in the numerator of the binomial coefficient, but it is more convenient to have it in the denominator in order to apply the method of coefficients. This is easily done as follows:

$$\binom{k-m-1}{m-2} = \binom{k-m-1}{k-2m+1} = \binom{-m+1}{k-2m+1}(-1)^{k-2m+1} = [t^k]\frac{t^{2m-1}}{(1-t)^{m-1}},$$

$$\binom{n-k+m-1}{m-1} = \binom{n-k+m-1}{n-k} = \binom{1-m-1}{n-k}(-1)^{n-k} = [t^{n-k}]\frac{1}{(1-t)^m}.$$

Now the convolution is clearly identified, and the sum reduces to a special case of the so-called Vandermonde convolution (see, for example, [8, p. 169]). We obviously have

$$S_{m,n} = [t^n]\frac{t^{2m-1}}{(1-t)^{2m-1}} = [t^{n-2m+1}]\frac{1}{(1-t)^{2m-1}} = \binom{n-1}{2m-2}.$$

In our opinion, the most striking aspect of the method of coefficients is the rule of composition. For instance, consider the Grosswald sum (1.2) after the change of variable $k \mapsto n-r-k$:

$$G_{n,r} = \sum_{k=0}^{n-r}(-2)^{-k}\binom{n}{r+k}\binom{n+r+k}{k} = \sum_{k=0}^{n-r}(-2)^{-n+r+k}\binom{n}{k}\binom{2n-k}{n-r-k}.$$

Applying Newton's rule and (K5), we compute:

$$G_{n,r} = (-2)^{r-n}\sum_{k=0}^{n-r}[u^k](1-2u)^n[v^{n-r-k}](1+v)^{2n-k} =$$

$$= (-2)^{r-n}[v^{n-r}](1+v)^{2n}\sum_{k=0}^{n-r}[u^k](1-2u)^n\frac{v^k}{(1+v)^k}.$$

The crucial point is that we can invoke the composition rule, so the whole expression collapses:

$$G_{n,r} = (-2)^{r-n}[v^{n-r}](1+v)^{2n}\left(1-\frac{2v}{1+v}\right)^n = (-2)^{r-n}[v^{n-r}](1-v^2)^n.$$

We conclude that

$$G_{n,r} = (-1)^{\frac{n-r}{2}}2^{r-n}\binom{n}{\frac{n-r}{2}},$$

which is 0 when $n-r$ is odd.

We now adopt a notational convention. When we wish to denote the substitution of a formal power series $g(t)$ for the indeterminate y in $f(y)$ (i.e., to form the composition $f(g(t))$), we write

$$f(g(t)) = [f(y) \mid y = g(t)],$$

which is a variant of the common notation $f(y)|_{y=g(t)}$. The latter notation becomes awkward when $g(t)$ is a complicated expression.

The following general result deals with the so-called Euler (or binomial) transformation and involves sums containing binomial coefficients:

Theorem 3.1 (Euler Transformation). *If $f(t)$ belongs to $\mathcal{F}$, then*

$$\sum_{k=0}^{n}\binom{n}{k}f_k = \sum_{k=0}^{n}\binom{n}{k}[t^k]f(t) = [t^n]\frac{1}{1-t}f\left(\frac{t}{1-t}\right).$$

Proof: From the properties of binomial coefficients (see section 2) it follows that

$$\binom{n}{k} = \binom{n}{n-k} = \binom{-n+n-k-1}{n-k}(-1)^{n-k} = \binom{-k-1}{n-k}(-1)^{n-k},$$

which is the coefficient of t^{n-k} in $(1-t)^{-k-1}$. We then observe that the sum in the theorem's statement can be extended to infinity, giving

$$\sum_{k=0}^{n}\binom{n}{k}f_k = \sum_{k=0}^{n}\binom{-k-1}{n-k}(-1)^{n-k}f_k = \sum_{k=0}^{\infty}[t^{n-k}](1-t)^{-k-1}[y^k]f(y) =$$

$$= [t^n]\frac{1}{1-t}\sum_{k=0}^{\infty}\left([y^k]f(y)\right)\left(\frac{t}{1-t}\right)^k = [t^n]\frac{1}{1-t}f\left(\frac{t}{1-t}\right),$$

in which we have applied the composition rule $(K5)$. ■

This same theorem allows us to perform a two-line procedure to obtain the closed form of S_m, the sum reported by Greene and Knuth in [9] and quoted in the introduction. It is well known that

$$[t^n]\frac{1}{\sqrt{1-4t}} = \binom{2n}{n},$$

$$[t^n]\frac{1}{\sqrt{1+2t}} = \left(-\frac{1}{2}\right)^n\binom{2n}{n}$$

(see also section 5). Hence we have

$$\sum_{k=0}^{m}\binom{m}{k}\left(-\frac{1}{2}\right)^k\binom{2k}{k} = [t^m]\frac{1}{1-t}\left[\frac{1}{\sqrt{1+2y}}\,\middle|\, y = \frac{t}{1-t}\right] = [t^m]\frac{1}{\sqrt{1-t^2}}.$$

This quantity is clearly 0 when m is odd and equals $\binom{m}{m/2}/2^m$ when m is even.

4 COMBINATORIAL SUMS.

As shown in the examples at the end of the previous section, the method of coefficients can be used to evaluate combinatorial sums by passing through generating functions. Nowadays, general techniques have been studied to perform definite and indefinite summation for large classes of expressions. Gosper's method [6] and the Petkovšek-Wilf-Zeilberger approach [19] are the most conspicuous examples, and every system for computer algebra embodies the corresponding algorithms. However, these methods are not convenient for hand calculation. Moreover, although they furnish the sums of very complicated expressions, something unsatisfactory remains in the results: you have proved what you wished to prove and you are now convinced that your theorem is true, but in reality you do not know "why" or "how" it is true. For this reason, we think that the method of coefficients might be interesting in the following situations:

(1) to obtain a constructive and "human" proof of sums for which it could be important to show the steps used to find the closed form (Often, mechanical proofs give the same unpleasant feeling that proofs by verification do. Think, for example, of proofs by mathematical induction.)

(2) to deal with different classes of expressions (The Gosper and Petkovšek-Wilf-Zeilberger's methods consider only sums of hypergeometric terms. In this way, they leave out expressions containing, for example, harmonic or Stirling numbers. The method of coefficients can also be used for these quantities.)

(3) to obtain asymptotic estimates (Another drawback of the aforementioned methods is that if a sum does not have a closed form, the user is informed thereof, but no hint is furnished for obtaining an asymptotic approximation, which can be almost as useful as a closed form. The method of coefficients usually arrives at a generating function and, even if extracting a coefficient is too difficult, asymptotic techniques can be used to obtain an estimate of the sum. A recent example can be found in H. K. Hwang [12].)

An interesting method for evaluating combinatorial sums has surfaced in the *Riordan arrays* concept (see [23], [24]). This concept is just a generalization of two theorems discussed earlier: the partial sum theorem (Theorem 2.1) and the Euler's transformation (Theorem 3.1). As a matter of fact, Riordan arrays correspond to a special application of the method of coefficients that allows one to compute (see Theorem 4.1) a vast number of combinatorial sums in a uniform and often very simple way.

Formally, a *Riordan array* is a pair of formal power series $D = (d(t), h(t))$. If both $d(t)$ and $h(t)$ are in $\mathcal{F}_0$, then the Riordan array is called *proper*. Here we always assume our Riordan arrays to be proper. A Riordan array D can be identified with the infinite, lower triangular array (or *triangle*) $(d_{n,k})_{n,k\geq 0}$ defined by

$$d_{n,k} = [t^n]d(t)(th(t))^k. \tag{4.1}$$

In fact, we are mainly interested in the sequence of functions $(d_k(t))$ defined recursively as follows:

$$\begin{cases} d_0(t) &= d(t), \\ d_k(t) &= d_{k-1}(t)th(t) = d(t)(th(t))^k \ (k = 1, 2, \ldots). \end{cases}$$

These functions are the column generating functions of the triangle. A common example of a Riordan array is the Pascal triangle, for which we have $d(t) = h(t) = 1/(1-t)$. Indeed, by (4.1)

$$[t^n]\frac{1}{1-t}\left(\frac{t}{1-t}\right)^k = [t^{n-k}](1-t)^{-k-1} = \binom{-k-1}{n-k}(-1)^{n-k} =$$

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

The most important algebraic property of Riordan arrays lies in the fact that the usual row-by-column product of two Riordan arrays is likewise a Riordan array. This is proved by considering two Riordan arrays $(d(t), h(t))$ and $(a(t), b(t))$ and computing the product. Its generic element is $\sum_j d_{n,j} f_{j,k}$, where $d_{n,j}$ is the generic element in $(d(t), h(t))$ and $f_{j,k}$ is the generic element in $(a(t), b(t))$. In fact

$$\sum_{j=0}^{\infty} d_{n,j} f_{j,k} = \sum_{j=0}^{\infty} [t^n]d(t)(th(t))^j [y^j]a(y)(yb(y))^k =$$

$$= [t^n]d(t)\sum_{j=0}^{\infty}(th(t))^j[y^j]a(y)(yb(y))^k = [t^n]d(t)a(th(t))(th(t)b(th(t)))^k.$$

We conclude that

$$(d(t), h(t)) \cdot (a(t), b(t)) = (d(t)a(th(t)), h(t)b(th(t))). \tag{4.2}$$

This expression is particularly important and is the basis for many developments in the Riordan array theory. It is now easy to show that the set of (proper) Riordan arrays is a group in which $(1, 1)$ acts as the identity.

We now suppose that $(d(t), h(t))$ is a proper Riordan array. In view of formula (4.2), we can look for a proper Riordan array $(a(t), b(t))$ such that $(d(t), h(t)) \cdot (a(t), b(t)) = (1, 1)$. If this is the case, we should have $d(t)a(th(t)) = 1$ and $h(t)b(th(t)) = 1$. Setting $y = th(t)$ we require that

$$a(y) = \left[d(t)^{-1} \;\middle|\; t = yh(t)^{-1} \right], \qquad\qquad b(y) = \left[h(t)^{-1} \;\middle|\; t = yh(t)^{-1} \right].$$

As we shall see, this allows us to find the inverse of a given Riordan array by appealing to the Lagrange inversion formula, which is introduced in the next section.

From our point of view, one of the most important properties of Riordan arrays is that sums involving the rows of a Riordan array can be calculated by first applying a suitable transformation on a generating function and then extracting a coefficient from the resulting function. More precisely, we can prove the following theorem:

Theorem 4.1 *If $D = (d(t), h(t))$ is a Riordan array and $f(t)$ is the generating function of the sequence $(f_k)_{k \geq 0}$, then*

$$\sum_{k=0}^{n} d_{n,k} f_k = [t^n] d(t) f(th(t)). \tag{4.3}$$

Proof: The proof consists in a straightforward computation with the method of coefficients:

$$\sum_{k=0}^{n} d_{n,k} f_k = \sum_{k=0}^{\infty} [t^n] d(t)(th(t))^k f_k = [t^n] d(t) \sum_{k=0}^{\infty} f_k (th(t))^k = [t^n] d(t) f(th(t)),$$

as desired. ■

In the case of the Pascal triangle Theorem 4.1 reduces to the Euler transformation (Theorem 3.1). More general formulas are obtained if we consider *simple binomial coefficients* (i.e., binomial coefficients of the form $\binom{n+ak}{m+bk}$, where a and b are parameters and k is a nonnegative integer variable). If we consider n a variable and m a parameter, and vice versa, we get two different infinite arrays $(d_{n,k})$ and $(\widehat{d}_{m,k})$ whose elements depend on the parameters a, b, m and a, b, n, respectively. In either case, if certain conditions on a and b hold, we have Riordan arrays and therefore we can apply formula (4.3) to determine the value of many sums. In this way we arrive at the following formulas:

$$\sum_k \binom{n+ak}{m+bk} f_k = [t^n] \frac{t^m}{(1-t)^{m+1}} f\left(\frac{t^{b-a}}{(1-t)^b} \right) \qquad (b > a), \tag{4.4}$$

$$\sum_k \binom{n+ak}{m+bk} f_k = [t^m](1+t)^n f(t^{-b}(1+t)^a) \qquad (b < 0). \tag{4.5}$$

If m and n are independent of each other, these relations can also be stated as generating function identities.

As a simple example, we consider the following sum, in which we apply (4.5) with $n = 0, m = n, a = 1$, and $b = -1$:

$$\sum_{k=n/2}^{n} \binom{k}{n-k} \frac{1}{k} = [t^n] \left[\ln \frac{1}{1-y} \middle| y = t(1+t) \right] = [t^n] \ln \frac{1}{1-t-t^2} =$$

$$= \frac{1}{n} [t^{n-1}] \frac{1+2t}{1-t-t^2} = \frac{F_n + 2F_{n-1}}{n} = \frac{F_{n+1} + F_{n-1}}{n},$$

where $F_n = [t^n] t/(1-t-t^2)$ is the nth Fibonacci number. We also refer the reader to [17] for a recent application of formula (4.5) to prove the identity

$$(x+m+1) \sum_{k=0}^{m} (-1)^k \binom{x+y+k}{m-k} \binom{y+2k}{k} - \sum_{k=0}^{m} \binom{x+k}{m-k} (-4)^k = (x-m) \binom{x}{m}.$$

These examples of combinatorial sums derived from formulas (4.4) and (4.5) illustrate the situation described in point (1) at the beginning of this section. As to point (2), very important examples of Riordan arrays are given by the Stirling numbers of both kinds. A *Stirling number of the first kind*, usually denoted by $\left[{n \atop k} \right]$, counts the number of permutations of n elements containing exactly k cycles. A *Stirling number of the second kind*, usually denoted by $\left\{ {n \atop k} \right\}$, is the number of partitions of a set with n elements into k nonempty parts (see, for example, the book of Graham, Knuth, and Patashnik [8, p. 243]). If we set

$$s_{n,k} = \frac{k!}{n!} \left[{n \atop k} \right], \qquad S_{n,k} = \frac{k!}{n!} \left\{ {n \atop k} \right\}$$

the corresponding Riordan arrays are:

$$(s_{n,k}) = \left(1, \frac{1}{t} \ln \frac{1}{1-t} \right), \qquad (S_{n,k}) = \left(1, \frac{e^t - 1}{t} \right).$$

In [24] the second author establishes a number of identities on Stirling numbers by invoking these Riordan arrays. Here we limit ourselves to demonstrating the orthogonality property of the two arrays:

$$\sum_k \left[{n \atop k} \right] \left\{ {k \atop m} \right\} (-1)^{n-k} = (-1)^n \frac{n!}{m!} \sum_k \frac{k!}{n!} \left[{n \atop k} \right] \frac{m!}{k!} \left\{ {k \atop m} \right\} (-1)^k =$$

$$= (-1)^n \frac{n!}{m!} [t^n] \left[(e^{-y} - 1)^m \middle| y = \ln \frac{1}{1-t} \right] = (-1)^n \frac{n!}{m!} [t^n] (-t)^m = \delta_{n,m}.$$

Another orthogonality property is proved in an analogous way. Namely, the reader is invited to prove the following identity:

$$\sum_k \left[{n \atop k} \right] \left\{ {k \atop m} \right\} = \frac{n!}{m!} \binom{n-1}{m-1}.$$

Finally, concerning point (3) and issues relating to asymptotics, we consider the following sum, whose generating function is immediately found by the method of coefficients:

$$\sum_{k=0}^{n} \binom{n}{k} \binom{2k}{k} = [t^n] \frac{1}{1-t} \left[\frac{1}{\sqrt{1-4y}} \middle| y = \frac{t}{1-t} \right] = [t^n] \frac{1}{\sqrt{(1-t)(1-5t)}}.$$

That there is no closed form for this sum is confirmed by the Petkovšek algorithm (see [19]). However, by Bender's approach (see [1, Theorem 2, p. 496]) we find immediately that $t = 1/5$ is the dominant singularity and therefore

$$\sum_{k=0}^{n} \binom{n}{k}\binom{2k}{k} \approx \sqrt{\frac{5}{\pi n}} \cdot \frac{5^n}{2},$$

which is easily checked.

5 INVERSION.

As already mentioned, the compositional inverse $\overline{f}(t)$ of a formal power series $f(t)$ is defined by the relation $f(\overline{f}(t)) = \overline{f}(f(t)) = t$. The form of $\overline{f}(t)$ is best determined by the Lagrange inversion formula, which is our rule $(K6)$. There exist many proofs of the Lagrange inversion formula in terms of formal Laurent series, for which the reader is referred to [2], [10], [11], [13], [14], [21], or [27]. (Further references can be found in Stanley [27, p. 67].) The method of coefficients can be used to prove the Lagrange inversion formula, via the following steps:

1. consider the proper Riordan array $F = (1, f(t)/t)$ whose columns are the successive powers of $f(t)$;
2. observe that the inverse Riordan array F^{-1} is just $(1, \overline{f}(t)/t)$, whose columns are the successive powers of $\overline{f}(t)$;
3. verify directly that the array D defined by
$$d_{n,k} = \frac{k}{n}[t^{n-k}]\left(\frac{t}{f(t)}\right)^n$$
is also the inverse of F;
4. conclude that $D = F^{-1}$ and thereby deduce that formula $(K6)$ is correct for every n and every k.

There is another way to apply the Lagrange inversion formula. Suppose that we have a functional equation $w = t\phi(w)$, where $\phi(t)$ belongs to $\mathcal{F}_0$, and that we wish to find the formal power series $w = w(t)$ satisfying this functional equation. Clearly $w(t)$ lies in $\mathcal{F}_1$ and if we set $f(y) = y/\phi(y)$, we also have $f(t)$ in $\mathcal{F}_1$. However, the functional equation can be written $f(w(t)) = t$, which shows that $w(t)$ is the compositional inverse of $f(t)$. We therefore know that $w(t)$ is uniquely determined, and we learn from the Lagrange inversion formula that

$$[t^n]w(t) = \frac{1}{n}[t^{n-1}]\left(\frac{t}{f(t)}\right)^n = \frac{1}{n}[t^{n-1}]\phi(t)^n.$$

The Lagrange inversion formula can also give us the coefficients of the powers $w(t)^k$, but we can obtain an even more general result. Let $F(t)$ be a member of $\mathcal{F}$ and consider the composition $F(w(t))$, where $w = w(t)$ is, as before, the solution to the functional equation $w = t\phi(w)$ for given $\phi(w)$ in $\mathcal{F}_0$. For the coefficient of t^n in $F(w(t))$ we have

$$[t^n]F(w(t)) = [t^n]\sum_{k=0}^{\infty} F_k w(t)^k = \sum_{k=0}^{\infty} F_k [t^n]w(t)^k = \sum_{k=0}^{\infty} F_k \frac{k}{n}[t^{n-k}]\phi(t)^n =$$

$$= \frac{1}{n}[t^{n-1}]\left(\sum_{k=0}^{\infty} kF_k t^{k-1}\right)\phi(t)^n = \frac{1}{n}[t^{n-1}]F'(t)\phi(t)^n. \tag{5.1}$$

Note that $[t^0]F(w(t)) = F_0$.

We are now in a position to prove the following important theorem on *diagonalization*:

Theorem 5.1 *If $F(t)$ and $\phi(t)$ belong to $\mathcal{F}$, then*

$$[t^n]F(t)\phi(t)^n = [t^n]\left[\frac{F(w)}{1-t\phi'(w)} \;\Big|\; w = t\phi(w)\right].$$

Proof: We first observe that

$$[t^n]F(t)\phi(t)^n = [t^{n-1}]\frac{F(t)}{t}\phi(t)^n = n[t^n]\left[\int \frac{F(y)}{y}dy \;\Big|\; y = w(t)\right]$$

(to get the last expression we applied formula (5.1) backwards, so $w = w(t)$ in $\mathcal{F}_1$ is the unique solution of the functional equation $w = t\phi(w)$). By now applying the rule of differentiation for the "coefficient of" operator, we can proceed to obtain

$$[t^n]F(t)\phi(t)^n = [t^{n-1}]\frac{d}{dt}\left[\int \frac{F(y)}{y}dy \;\Big|\; y = w(t)\right] = [t^{n-1}]\left[\frac{F(w)}{w} \;\Big|\; w = t\phi(w)\right]\frac{dw}{dt},$$

in which we have applied the chain rule for differentiation. From $w = t\phi(w)$ we have

$$\frac{dw}{dt} = \phi(w) + t\left[\frac{d\phi}{dw} \;\Big|\; w = t\phi(w)\right]\frac{dw}{dt}$$

or

$$\frac{dw}{dt} = \left[\frac{\phi(w)}{1-t\phi'(w)} \;\Big|\; w = t\phi(w)\right],$$

where $\phi'(w)$ denotes the derivative of $\phi(w)$ with respect to w. We can substitute this expression into the previous formula and observe that $w/\phi(w) = t$ can be taken outside of the substitution parentheses, leading to

$$[t^n]F(t)\phi(t)^n = [t^{n-1}]\left[\frac{F(w)}{w}\frac{\phi(w)}{1-t\phi'(w)} \;\Big|\; w = t\phi(w)\right] =$$

$$= [t^{n-1}]\frac{1}{t}\left[\frac{F(w)}{1-t\phi'(w)} \;\Big|\; w = t\phi(w)\right] = [t^n]\left[\frac{F(w)}{1-t\phi'(w)} \;\Big|\; w = t\phi(w)\right],$$

as desired. ■

The name "diagonalization" stems from the fact that if we imagine the coefficients of $F(t)\phi(t)^n$ as constituting row n in an infinite matrix, then the $[t^n]F(t)\phi(t)^n$ are just the elements in the main diagonal of this array. One of its simplest applications is determining the generating function of the central binomial coefficients. Since plainly

$$\binom{2n}{n} = [t^n](1+t)^{2n},$$

Theorem 5.1 can be applied with $F(t) = 1$ and $\phi(t) = (1+t)^2$. In this case, the function $w = w(t)$ is easily determined by solving the functional equation $w = t(1+w)^2$. Expanding, we find that $tw^2 - (1-2t)w + t = 0$ or

$$w = w(t) = \frac{1 - 2t \pm \sqrt{1-4t}}{2t}.$$

Since $w = w(t)$ should belong to $\mathcal{F}_1$, we must eliminate the solution with the + sign. Consequently, using $\mathcal{G}$ to signify the *generating function of a sequence*, we have the following well-known result:

$$\mathcal{G}\left\{\binom{2n}{n}\right\} = \left[\frac{1}{1-2t(1+w)} \;\middle|\; w = \frac{1-2t-\sqrt{1-4t}}{2t}\right] = \frac{1}{\sqrt{1-4t}}.$$

As a more complex example, we prove the famous *Abel identity* (see [20]):

$$S_n = \sum_{k=0}^{n} \binom{n}{k} a(a+k)^{k-1}(b+n-k)^{n-k} = (a+b+n)^n.$$

This identity has application, for example, in the analysis of linear probing in hashing, a fast method to find an element inside a table (see Sedgewick and Flajolet [22, p. 450]). The left-hand term can be written

$$n! \sum_{k=0}^{n} \frac{a(a+k)^{k-1}}{k!} \cdot \frac{(b+n-k)^{n-k}}{(n-k)!},$$

revealing that the sum is actually a convolution. Now we have

$$\frac{(b+k)^k}{k!} = [t^k]e^{(b+k)t} = [t^k]e^{bt}(e^t)^k$$

and can apply the diagonalization rule:

$$\mathcal{G}\left\{\frac{(b+k)^k}{k!}\right\} = \left[\frac{e^{bw}}{1-te^w} \;\middle|\; w = te^w\right] = \left[\frac{e^{bw}}{1-w} \;\middle|\; w = te^w\right]. \tag{5.2}$$

In a similar way we obtain

$$\frac{a(a+k)^{k-1}}{k!} = \frac{(a+k)^k}{k!} - \frac{(a+k)^{k-1}}{(k-1)!} = [t^k]e^{(a+k)t} - [t^{k-1}]e^{(a+k)t} = [t^k](1-t)e^{(a+k)t},$$

and proceeding as before we discover that

$$\mathcal{G}\left\{\frac{a(a+k)^{k-1}}{k!}\right\} = \left[e^{aw} \;\middle|\; w = te^w\right].$$

Finally, we convolve the two expressions, which yields

$$S_n = [t^n]\left[\frac{e^{(a+b)w}}{1-w} \;\middle|\; w = te^w\right] = \frac{(a+b+n)^n}{n!}$$

after applying the diagonalization rule backwards (since $1 - w = 1 - te^w$, as already seen in (5.2)). A systematic approach of this type has been used in [25] to prove a number of identities related to Abel's and Gould's formulas.

We conclude this paper by returning to Riordan arrays in order to illustrate how that concept, together with the technique of diagonalization, allows one to solve *combinatorial inversions*. Let (a_n) be a sequence defined by an infinite set of equations

$$a_n = \sum_{k=0}^{n} f_{n,k} b_k,$$

where the sequence (b_n) is unknown. The problem of *inversion* is to find the values of b_n in terms of those of a_n, that is, to express b_n in the form

$$b_n = \sum_{k=0}^{n} g_{n,k} a_k.$$

As an illustrative example, we choose inversion (2.4.2) in Riordan's book [20]. Although it is not particularly important, it does provide an application of Riordan arrays that we have not considered in our previous examples. We consider

$$a_n = \sum_k \left(\binom{n}{k} - (c-1)\binom{n}{k-1} \right) b_{n-ck}.$$

Let $a(t)$ and $b(t)$ be the generating functions of the two sequences. The relevant Riordan array R is given by $b_{n-ck} = [t^n]b(t)(t^c)^k$, whence $R = (b(t), t^{c-1})$. The generating function of the other term is determined by

$$\binom{n}{k} - (c-1)\binom{n}{k-1} = [t^k](1+t)^n - (c-1)[t^{k-1}](1+t)^n = [t^k](1-(c-1)t)(1+t)^n.$$

From the summation formula (4.3) we infer that

$$a_n = [t^n]b(t)(1-(c-1)t^c)(1+t^c)^n.$$

We now invoke the diagonalization formula to get

$$a(t) = \left[\frac{b(w)(1-(c-1)w^c)}{1-tcw^{c-1}} \;\middle|\; w = t(1+w) \right] = \left[b(w)(1+w^c) \;\middle|\; t = \frac{w}{1+w^c} \right].$$

Instead of trying to compute $w = w(t)$, we substitute t into the left-hand term and change variables directly:

$$a\left(\frac{w}{1+w^c}\right) = b(w)(1+w^c)$$

or

$$b(w) = \frac{1}{1+w^c} a\left(\frac{w}{1+w^c}\right) \tag{5.3}$$

This, however, is the transformation related to the Riordan array $(1/(1+w^c), 1/(1+w^c))$, the generic element of which is

$$d_{n,k} = [w^n]\frac{1}{1+w^c} \cdot \frac{w^k}{(1+w^c)^k} = [w^{n-k}](1+w^c)^{-k-1} = [w^{n-k}]\sum_j \binom{-k-1}{j} w^{cj} =$$

$$= [w^{n-k}]\sum_j \binom{k+1+j+1}{j}(-1)^j w^{cj} = \binom{n-cj+j}{j}(-1)^j,$$

since the sum reduces to the case $n-k=cj$. If we now write k instead of j and pass to the coefficients, equation (5.3) becomes

$$b_n = \sum_k \binom{n-(c-1)k}{k}(-1)^k a_{n-ck}.$$

ACKNOWLEDGMENTS. We wish to thank the anonymous referee for his criticism and useful suggestions. We are also indebted to G. E. Andrews, I. Gessel, and E. K. Lloyd for their interesting historical comments, and to B. Richmond and L. W. Shapiro.

References

[1] E. A. Bender, Asymptotic methods in enumeration, *SIAM Review* **16** (1974) 485–515.

[2] T. J. I'A. Bromwich, *An Introduction to the Theory of Infinite Series*, Macmillan, London, 1931.

[3] G. P. Egorychev, *Integralnoe Predstavlenie i Vychislenie Kombinatornykh Summ*, Izdat. "Nauka" Sibirsk. Otdel., Novosibirsk, 1977.

[4] ———, *Integral Representation and the Computation of Combinatorial Sums (trans. H. H. McFadden)*, vol. 59, American Mathematical Society, Providence, 1984.

[5] ———, Algorithms of integral representation of combinatorial sums and their applications, *Formal Power Series and Algebraic Combinatorics (Moscow, 2000)*, Springer-Verlag, Berlin, 2000, pp. 15–29.

[6] R. W. Gosper, Decision procedure for indefinite hypergeometric summation, *Proc. Nat. Acad. Sciences U.S.A.* **75** (1978) 40–42.

[7] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration*, Wiley, New York, 1983.

[8] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, Addison-Wesley, New York, 1989.

[9] D. H. Greene and D. E. Knuth, *Mathematics for the Analysis of Algorithms*, Birkäuser, Boston, 1982.

[10] P. Henrici, An algebraic proof of the Lagrange-Bürmann formula, *J. Math. Anal. Appl.* **8** (1964) 218–224.

[11] ———, *Applied and Computational Complex Analysis, I*, Wiley, New York, 1988.

[12] H. K. Hwang, Uniform asymptotics of some Abel sums arising in coding theory, *Theoret. Comput. Sci.* **263** (2001) 145–158.

[13] E. Jabotinsky, Sur la représentation de la composition de fonctions par un produit de matrices. Application à l'itération de e^x et de e^x-1, *C. R. Acad. Sci. Paris* **224** (1947), 323–324.

[14] C. G. J. Jacobi, De resolutione aequationum per series infinitas, *J. Reine Angew. Math.* **6** (1830) 257–286.

[15] D. E. Knuth, *The Art of Computer Programming*, vol. 1-3, Addison-Wesley, Reading, MA, 1973.

[16] ———, Bracket notation for the 'coefficient-of' operator, *A Classical Mind, Essays in Honour of C. A. R. Hoare*, edited by A. W. Roscoe, Prentice-Hall, UK, (1994) 247–258.

[17] D. Merlini and R. Sprugnoli, A Riordan array proof of a curious identity, *Integers* **2** (2002), A8.

[18] C. Ó'Dúnlaing and L. Ericson, Enumeration of tree properties by naïve methods, *Algorithms review* **1**, n. 3, (1990) 119–124.

[19] M. Petkovšek, H. S. Wilf, and D. Zeilberger, *A=B*, AK Peters, Natick, MA, 1996.

[20] J. Riordan, *An Introduction to Combinatorial Analysis*, Wiley, New York, 1958.

[21] I. Schur, Identities in the theory of power series, *Amer. J. Math.* **69** (1947) 14–26.

[22] R. Sedgewick and P. Flajolet, *An Introduction to the Analysis of Algorithms*, Addison-Wesley, Reading, MA, 1996.

[23] L. W. Shapiro, S. Getu, W.-J. Woan, and L. Woodson, The Riordan group, *Discrete Appl. Math.* **34** (1991) 229–239.

[24] R. Sprugnoli, Riordan arrays and combinatorial sums, *Discrete Math.* **132** (1994) 267–290.

[25] ———, Riordan arrays and the Abel-Gould identity, *Discrete Math.* **142** (1995) 213–233.

[26] R. P. Stanley, *Enumerative Combinatorics*, vol. 1, Wadsworth, Cambridge, 1986.

[27] ———, *Enumerative Combinatorics*, vol. 2, Cambridge University Press, Cambridge, 1999.

[28] H. S. Wilf, *Generatingfunctionology*, Academic Press, Boston, 1990.

DONATELLA MERLINI received her master degree (laurea) in mathematics in 1992 and her Ph.D. in computer science and automation engineering in 1996 from the University of Florence (Italy). In 1997 she became research associate at the Systems and Computer Science Department of the University of Florence, and she is presently an associate professor in the same department. Her research and teaching activities deal with the analysis of algorithms, data structures and algorithms, combinatorics, asymptotics, and symbolic computation.
Università di Firenze, Dipartimento di Sistemi e Informatica
viale Morgagni 65, 50134, Firenze, Italia
merlini@dsi.unifi.it

RENZO SPRUGNOLI obtained his master degree (laurea) in mathematics from the University of Pisa (Italy) in 1965. Up to 1985 he was a researcher at the Italian Research Council and taught algebra and data base management systems at Pisa University. In 1987 he became full professor of computer science. As such, he has taught at the Universities of Padua and Florence. Presently, teaches classes on algorithms, data structures, and databases at Florence. His interests focus mainly on the analysis of algorithms and combinatorics.
Università di Firenze, Dipartimento di Sistemi e Informatica
viale Morgagni 65, 50134, Firenze, Italia
sprugnoli@dsi.unifi.it

MARIA CECILIA VERRI received her master degree (laurea) in mathematics from the University of Florence (Italy) in 1988. Since 1992 she has been a research associate at the Systems and Computer Science Department of the University of Florence, and in 2000 she became an associate professor in that department. She presently teaches courses on algorithms and human computer interaction. Her research activities deal with the following subjects: analysis of algorithms, data structures and algorithms, combinatorics, asymptotics, algorithm animation.
Università di Firenze, Dipartimento di Sistemi e Informatica
viale Morgagni 65, 50134, Firenze, Italia
verri@dsi.unifi.it