Konrad-Zuse-Zentrum für Informationstechnik Berlin
Heilbronner Str. 10, D-10711 Berlin - Wilmersdorf

Wolfram Koepf

Algorithms for the Indefinite and Definite Summation

Preprint SC 94–33 (Dezember 1994)

Algorithms for the Indefinite and Definite Summation

Wolfram Koepf

koepf@zib-berlin.de

Abstract

The celebrated Zeilberger algorithm which finds holonomic recurrence equations for definite sums of hypergeometric terms $F(n,k)$ is extended to certain nonhypergeometric terms. An expression $F(n,k)$ is called hypergeometric term if both $F(n+1,k)/F(n,k)$ and $F(n,k+1)/F(n,k)$ are rational functions. Typical examples are ratios of products of exponentials, factorials, Γ function terms, binomial coefficients, and Pochhammer symbols that are integer-linear with respect to n and k in their arguments.

We consider the more general case of ratios of products of exponentials, factorials, Γ function terms, binomial coefficients, and Pochhammer symbols that are rational-linear with respect to n and k in their arguments, and present an extended version of Zeilberger's algorithm for this case, using an extended version of Gosper's algorithm for indefinite summation.

In a similar way the Wilf-Zeilberger method of rational function certification of integer-linear hypergeometric identities is extended to rational-linear hypergeometric identities.

The given algorithms on definite summation apply to many cases in the literature to which neither the Zeilberger approach nor the Wilf-Zeilberger method is applicable. Examples of this type are given by theorems of Watson and Whipple, and a large list of identities ("Strange evaluations of hypergeometric series") that were studied by Gessel and Stanton. It turns out that with our extended algorithms practically all hypergeometric identities in the literature can be verified.

Finally we show how the algorithms can be used to generate new identities.

REDUCE and MAPLE implementations of the given algorithms can be obtained from the author, many results of which are presented in the paper.

1 Hypergeometric identities

In this paper we deal with hypergeometric identities. As usual, the notation of the generalized hypergeometric function ${}_pF_q$ defined by

$$ {}_pF_q\left(\begin{array}{cccc} a_1 & a_2 & \cdots & a_p \\ b_1 & b_2 & \cdots & b_q \end{array} \middle|\, x\right) := \sum_{k=0}^{\infty} A_k\, x^k = \sum_{k=0}^{\infty} \frac{(a_1)_k\cdot(a_2)_k\cdots(a_p)_k}{(b_1)_k\cdot(b_2)_k\cdots(b_q)_k\,k!}x^k \qquad (1)$$

is used, $(a)_k = \frac{\Gamma(a+k)}{\Gamma(k)}$ denoting the *Pochhammer symbol* or *shifted factorial.* The numbers a_k are called the *upper*, and b_k the *lower parameters* of ${}_pF_q$.

The coefficients A_k of the generalized hypergeometric function have the rational term ratio

$$\frac{A_{k+1}}{A_k} = \frac{(k+a_1)\cdot(k+a_2)\cdots(k+a_p)}{(k+b_1)\cdot(k+b_2)\cdots(k+b_q)(k+1)} \quad (k\in\mathbb{N})\;.$$

If an expression A_k has a rational term ratio A_{k+1}/A_k, we call A_k a *hypergeometric term* or *closed form*. Note that any hypergeometric term essentially has a representation as the ratio of shifted factorials (over $\mathbb{C}$), and its generating function is connected with a generalized hypergeometric series.
The classical reference concerning generalized hypergeometric series is the book of Bailey (1935) containing a huge amount of relations between hypergeometric series some of which represent the value of certain hypergeometric functions at a special point (mostly $x = 1$ or $x = -1$) by a single hypergeometric term. We will be concerned with this type of identities, and Table 1 is a complete list of all such hypergeometric identities found in Bailey's book.
Here $n \in \mathbb{N}$ is assumed to represent a positive integer so that the hypergeometric series with upper parameter $-n$ are terminating. All other parameters involved represent arbitrary complex variables such that none of the lower parameters corresponds to a negative integer.
With a method due to Wilf and Zeilberger, and with an algorithm of Zeilberger, many of these hypergeometric identities can be checked. It turns out, however, that for some of these identities both methods fail. We give extensions of both the Wilf-Zeilberger approach, and the (fast) Zeilberger algorithm with which all above identities can be handled as well as a large list of identities that were studied by Gessel and Stanton (1982).
Our extensions therefore unify the verification of hypergeometric identities.

2 The Gosper Algorithm

In this section we recall the celebrated Gosper algorithm (Gosper, 1978), see also Graham, Knuth and Patashnik (1994).
The Gosper algorithm deals with the question to find an antidifference s_k for given a_k, i. e. a sequence s_k for which

$$a_k = s_k - s_{k-1} \tag{2}$$

in the particular case that s_k is a hypergeometric term, therefore

$$\frac{s_k}{s_{k-1}} \quad \text{is a rational function with respect to } k \;, \tag{3}$$

i. e. $s_k/s_{k-1} \in \mathbb{Q}(k)$. We call this *indefinite summation*.
Note that if a hypergeometric term antidifference s_k exists, we call the input function a_k *Gosper-summable* which then itself is a hypergeometric term since by (2) and (3)

$$\frac{a_k}{a_{k-1}} = \frac{s_k - s_{k-1}}{s_{k-1} - s_{k-2}} = \frac{\frac{s_k}{s_{k-1}} - 1}{1 - \frac{s_{k-2}}{s_{k-1}}} = \frac{u_k}{v_k} \in \mathbb{Q}(k)$$

is rational, i. e. $u_k, v_k \in \mathbb{Q}[k]$ are polynomials.
Now, Gosper uses a representation lemma for rational functions to express a_k/a_{k-1} in terms of polynomials.
The idea behind this step comes from the following observation: If we calculate a_k from $s_k = (2k)!/k!$, e.g., we get

$$a_k = s_k - s_{k-1} = \frac{(2k)!}{k!} - \frac{(2k-2)!}{(k-1)!} = ((2k)(2k-1) - k) \cdot \frac{(2k-2)!}{k!} = k(4k-3) \cdot \frac{(2k-2)!}{k!},$$

i. e. a product of a polynomial $p_k = k(4k-3)$ and a factorial term $b_k = \frac{(2k-2)!}{k!}$ for which $b_k/b_{k-1} = q_k/r_k$ is rational, and therefore q_k and r_k can be assumed to be polynomials.

Table 1: Bailey's hypergeometric database

page	Theorem	Identity
2–3	Vandermonde Gauß	${}_2F_1\left(\begin{matrix} a\,,b \\ c\end{matrix}\middle\vert\, 1\right) = \frac{(c-b)_{-a}}{(c)_{-a}} = \frac{\Gamma(c)\Gamma(c-a-b)}{\Gamma(c-a)\Gamma(c-b)}$
9	Saalschütz	${}_3F_2\left(\begin{matrix} a\,,b\,,-n \\ c\,,1+a+b-c-n\end{matrix}\middle\vert\, 1\right) = \frac{(c-a)_n\,(c-b)_n}{(c)_n\,(c-a-b)_n}$
9	Kummer	${}_2F_1\left(\begin{matrix} a\,,b \\ 1+a-b\end{matrix}\middle\vert\, -1\right) = \frac{(1+a)_{-b}}{(1+a/2)_{-b}} = \frac{\Gamma(1+a-b)\Gamma(1+a/2)}{\Gamma(1+a)\Gamma(1+a/2-b)}$
11	Gauß	${}_2F_1\left(\begin{matrix} a\,,b \\ (a+b+1)/2\end{matrix}\middle\vert\, \frac{1}{2}\right) = \frac{\Gamma(1/2)\Gamma((a+b+1)/2)}{\Gamma((a+1)/2)\Gamma((b+1)/2)}$
11	Bailey	${}_2F_1\left(\begin{matrix} a\,,1-a \\ c\end{matrix}\middle\vert\, \frac{1}{2}\right) = \frac{\Gamma(c/2)\Gamma((c+1)/2)}{\Gamma((a+c)/2)\Gamma((1-a+c)/2)}$
13	Dixon	${}_3F_2\left(\begin{matrix} a\,,b\,,c \\ 1+a-b\,,1+a-c\end{matrix}\middle\vert\, 1\right) = \frac{(1+a)_{-c}\,(1+a/2-b)_{-c}}{(1+a/2)_{-c}\,(1+a-b)_{-c}} = \frac{\Gamma(1+a/2)\Gamma(1+a-b)\Gamma(1+a-c)\Gamma(1+a/2-b-c)}{\Gamma(1+a)\Gamma(1+a/2-b)\Gamma(1+a/2-c)\Gamma(1+a-b-c)}$
16	Watson Whipple	${}_3F_2\left(\begin{matrix} a\,,b\,,c \\ (a+b+1)/2\,,2c\end{matrix}\middle\vert\, 1\right) = \frac{\Gamma(\frac{1}{2})\Gamma(\frac{1+2c}{2})\Gamma(\frac{1+a+b}{2})\Gamma(\frac{1-a-b+2c}{2})}{\Gamma(\frac{1+a}{2})\Gamma(\frac{1+b}{2})\Gamma(\frac{1-a+2c}{2})\Gamma(\frac{1-b+2c}{2})}$
16	Whipple	${}_3F_2\left(\begin{matrix} a, 1-a, c \\ e, 1+2c-e\end{matrix}\middle\vert\, 1\right) = \frac{\pi 2^{1-2c}\Gamma(e)\Gamma(1+2c-e)}{\Gamma(\frac{a+e}{2})\Gamma(\frac{a+1+2c-e}{2})\Gamma(\frac{1-a+e}{2})\Gamma(\frac{2+2c-a-e}{2})}$
26	Dougall's Theorem	${}_7F_6\left(\begin{matrix} a\,,1+a/2\,,b\,,c\,,d\,,1+2a-b-c-d+n\,,-n \\ a/2\,,1+a-b\,,1+a-c\,,1+a-d\,,b+c+d-a-n\,,1+a+n\end{matrix}\middle\vert\, 1\right) = \frac{(1+a)_n\,(1+a-b-c)_n\,(1+a-b-d)_n\,(1+a-c-d)_n}{(1+a-b)_n(1+a-c)_n\,(1+a-d)_n\,(1+a-b-c-d)_n}$
25/27	Dougall	${}_5F_4\left(\begin{matrix} a, 1+a/2, c, d, e \\ a/2, 1+a-c, 1+a-d, 1+a-e\end{matrix}\middle\vert\, 1\right) = \frac{(1+a)_{-e}(1+a-c-d)_{-e}}{(1+a-c)_{-e}(1+a-d)_{-e}} = \frac{\Gamma(1+a-c)\Gamma(1+a-d)\Gamma(1+a-e)\Gamma(1+a-c-d-e)}{\Gamma(1+a)\Gamma(1+a-d-e)\Gamma(1+a-c-e)\Gamma(1+a-c-d)}$
28	Whipple	${}_4F_3\left(\begin{matrix} a, 1+a/2, d, e \\ a/2, 1+a-d, 1+a-e\end{matrix}\middle\vert\, -1\right) = \frac{(1+a)_{-e}}{(1+a-d)_{-e}} = \frac{\Gamma(1+a-d)\Gamma(1+a-e)}{\Gamma(1+a)\Gamma(1+a-d-e)}$
30	Bailey	${}_3F_2\left(\begin{matrix} a\,,1+a/2\,,-n \\ a/2\,,w\end{matrix}\middle\vert\, 1\right) = \frac{(w-a-1-n)(w-a)_{n-1}}{(w)_n}$
30	Bailey	${}_3F_2\left(\begin{matrix} a\,,b\,,-n \\ 1+a-b\,,1+2b-n\end{matrix}\middle\vert\, 1\right) = \frac{(a-2b)_n(1+a/2-b)_n(-b)_n}{(1+a-b)_n(a/2-b)_n(-2b)_n}$
30	Bailey	${}_4F_3\left(\begin{matrix} a\,,1+a/2\,,b\,,-n \\ a/2\,,1+a-b\,,1+2b-n\end{matrix}\middle\vert\, 1\right) = \frac{(a-2b)_n(-b)_n}{(1+a-b)_n(-2b)_n}$
30	Bailey	${}_4F_3\left(\begin{matrix} a, 1+a/2, b, -n \\ a/2, 1+a-b, 2+2b-n\end{matrix}\middle\vert\, 1\right) = \frac{(a-2b-1)_n(1/2+a/2-b)_n(-b-1)_n}{(1+a-b)_n(a/2-b-1/2)_n(-2b-1)_n}$

Gosper shows then that such a representation with the property

$$\gcd\,(q_k, r_{k+j}) = 1 \qquad \text{for all } j \in \mathbb{N}_0 \tag{4}$$

generally can be found and gives an algorithm to generate it.
Therefore we have for a_k the relation

$$\frac{a_k}{a_{k-1}} = \frac{p_k}{p_{k-1}}\,\frac{q_k}{r_k}\;, \tag{5}$$

p_k corresponding to the polynomial and (q_k, r_k) to the factorial part of a_k.
Gosper finally defines the function f_k by the equation

$$s_k = \frac{q_{k+1}}{p_k}\, f_k\, a_k \tag{6}$$

for which one sees immediately that

$$f_k = \frac{p_k}{q_{k+1}}\,\frac{s_k}{a_k} = \frac{p_k}{q_{k+1}}\,\frac{s_k}{s_k - s_{k-1}} = \frac{p_k}{q_{k+1}}\,\frac{\frac{s_k}{s_{k-1}}}{\frac{s_k}{s_{k-1}} - 1}$$

is rational. Using (4), Gosper proves the essential fact that f_k is a polynomial.
It follows from its defining equation that the polynomial f_k satisfies

$$a_k = s_k - s_{k-1} = \frac{q_{k+1}}{p_k}\, f_k\, a_k - \frac{q_k}{p_{k-1}}\, f_{k-1}\, a_{k-1}\;,$$

or multiplying by p_k/a_k, and using (5), one gets the recurrence equation

$$p_k = q_{k+1}\, f_k - q_k\, \frac{p_k}{p_{k-1}}\,\frac{a_{k-1}}{a_k}\, f_{k-1} = q_{k+1}\, f_k - r_k\, f_{k-1}\;. \tag{7}$$

Using (7), Gosper gives an upper bound for the degree of f in terms of the degrees of p_k, q_k, and r_k which yields a fast method to calculate f_k, so that we finally find s_k, given by (6). This shows in particular that whenever a_k possesses a closed form antidifference s_k then necessarily s_k is a rational multiple of a_k:

$$s_k = R_k\, a_k \qquad \text{with} \qquad R_k = \frac{q_{k+1}\, f_k}{p_k}\;.$$

If any of the steps to find the polynomial f_k fails, the algorithm proves that no hypergeometric term antidifference s_k of a_k exists.
Therefore the Gosper algorithm is a *decision procedure* which either returns "No closed form antidifference exists" or returns a closed form antidifference s_k of a_k, *provided one can decide the rationality of* a_k/a_{k-1}, i. e. one finds polynomials u_k, v_k such that $a_k/a_{k-1} = u_k/v_k$. In so far, the Gosper algorithm is an algorithm with input u_k and v_k rather than a_k.
Since without preprocessing, the user's input is a_k rather than the polynomials u_k and v_k, the success of an implementation depends heavily on an algorithm quickly and safely calculating (u_k, v_k) given a_k. In Algorithm 1, we present such a method. It turns out that none of the existing implementations of Gosper's algorithm uses such a method, examples of which we will consider later.

In case, the Gosper algorithm provides us with an antidifference s_k of a_k, any sum

$$\sum_{k=m}^{n} a_k = s_n - s_{m-1}$$

can be easily calculated by an evaluation of s_k at the boundary points like in the integration case. Note, however, that the sum

$$\sum_{k=0}^{n} \binom{n}{k} \tag{8}$$

e. g. is not of this type as the summand $\binom{n}{k}$ depends on the upper boundary point n explicitly. This is an example of a definite sum that we consider in § 3.

Gosper implemented his algorithm in the MACSYMA `nusum` command, an implementation of the algorithm is distributed with the `sum` command of the MAPLE system (to check its use set `infolevel[sum]:=5`), and one was delivered with MATHEMATICA Version 1.2 (in the package `Algebra/GosperSum.m`). Another MATHEMATICA implementation was given by Paule and Schorn (1994).

On the lines of Koornwinder (1993), together with Gregor Stölting I implemented the Gosper algorithm in REDUCE (Koepf, 1994) and MAPLE, using the simple decision procedure for rationality of hypergeometric terms described in Algorithm 1 below rather than internal simplification procedures (like MAPLE's `expand`). In § 3, we will show that this makes our implementations that can be obtained from the author much stronger than the previous ones.

It is almost trivial but decisive that the following is a decision procedure for the rationality of a_k/a_{k-1} for input a_k (at least) of a special type:

Algorithm 1 (`simplify_combinatorial`)
The following algorithm decides the rationality of a_k/a_{k-1}:

1. Input: a_k as ratio of products of rational functions, exponentials, factorials, Γ function terms, binomial coefficients, and Pochhammer symbols that are rational-linear in their arguments.

2. (`togamma`)
   Build a_k/a_{k-1}, and convert all occurrences of factorials, binomial coefficients, and Pochhammer symbols to Γ function terms.

3. (`simplify_gamma`)
   Recursively rewrite this expression according to the rule

   $$\Gamma(a+k) = (a)_k \cdot \Gamma(a)$$

   ($(a)_k := a(a+1)\cdots(a+k-1)$ denoting the Pochhammer symbol) whenever the arguments a and $a+k$ of two representing Γ function terms have positive integer difference k. Reduce the final fraction cancelling common Γ terms.

4. (`simplify_power`)
   Recursively rewrite the last expression according to the rule

   $$b^{a+k} = b^k\, b^a$$

   whenever the arguments a and $a+k$ of two representing exponential terms have positive integer difference k. Reduce the final fraction cancelling common exponential terms.

5. The expression a_k/a_{k-1} is rational if and only if the resulting expression in step 4 is rational u_k/v_k, $u_k, v_k \in \mathbb{Q}[k]$.

6. Output: (u_k, v_k).

Note that this result follows immediately from the given form of a_k and therefore of the expression a_k/a_{k-1} considered.
As an example, the rationality of a_k/a_{k-1} of

$$a_k = \frac{\Gamma(2k)}{4^k\,\Gamma(k)\,\Gamma(k+1/2)}$$

is recognized by the given procedure, and from the resulting information ($a_k/a_{k-1} = 1$), by induction $a_k = 1/(2\sqrt{\pi})$ (Abramowitz and Stegun, 1964, (6.1.18)).
Algorithm 1 does also apply to

$$a_k = \Gamma(2k) - \alpha\, 4^k\, \Gamma(k)\,\Gamma(k+1/2)\,,$$

and leads to

$$\frac{a_k}{a_{k-1}} = 2\,(2\,k-1)\,(k-1)$$

which is true whenever $\alpha \neq \frac{1}{2\sqrt{\pi}}$. If $\alpha = \frac{1}{2\sqrt{\pi}}$, however, $a_k \equiv 0$, and therefore a_k/a_{k-1} is not properly defined.
For this reason, Gosper's algorithm with input

$$a_k = \Gamma(2k) - \frac{1}{2\sqrt{\pi}}\, 4^k\, \Gamma(k)\,\Gamma(k+1/2)$$

fails to find the true antidifference $s_k = 1$.
We note, however, that in most cases also sums of ratios of the described form can be treated by the same method without using multiplication formulas of the Γ function explicitly. An important family of examples of this type will be considered in the next section.
We note, finally, that an implementation of Algorithm 1 in a computer algebra system allows the user to enter his input in the form in which it is found in the literature, and no preprocessing is necessary. The user can be sure that the rationality is decided correctly. Unfortunately, this is not so with any of the current implementations.
MAPLE's `expand` command which is used for that purpose in Zeilberger's (1990) and Koornwinder's (1993) implementations of Zeilberger's algorithm (see § 6), e. g., does not the job required. The same is valid for MATHEMATICA's `FactorialSimplify` procedure that comes in the package `DiscreteMath/RSolve.m`. Also, the Gosper implementation which comes with MAPLE's `sum` command, has the same failure. Paule and Schorn's MATHEMATICA implementation aborts in those cases with the message `input not interpretable`, whereas Gosper's `nusum` command gives the (wrong) error message `errexp1 NON-RATIONAL TERM RATIO TO NUSUM`.
Therefore in all these implementations, the fact that Gosper's algorithm is a decision procedure, unfortunately is completely lost. An example for that fact is given by the expression

$$a_k = \frac{1}{2^n}\binom{n}{k} - \frac{1}{2^{n-1}}\binom{n-1}{k}$$

with

$$\frac{a_k}{a_{k-1}} = \frac{(n-k+1)(n-2k)}{k(n-2k+2)} ,$$

for which our Gosper implementations succeed very quickly, whereas MAPLE's `sum` command as well as MATHEMATICA's `GosperSum`, Paule/Schorn's `Gosper`, and Gosper's `nusum` fail if the input is not preprocessed by the user.
This is a very simple example of an important type of examples considered next.

3 The Wilf-Zeilberger Method

Examples for an application of the Gosper algorithm in connection with Algorithm 1 are given by the Wilf-Zeilberger method on *definite summation* (Wilf and Zeilberger, 1990), see also Wilf (1993).
The Wilf-Zeilberger method is a direct application of Gosper's algorithm to prove identities of the form

$$s_n := \sum_{k\in\mathbb{Z}} F(n,k) = 1 \tag{9}$$

for which $F(n,k)$ is a hypergeometric term with respect to both n and k, i. e.

$$\frac{F(n,k)}{F(n-1,k)} \quad \text{and} \quad \frac{F(n,k)}{F(n,k-1)} \quad \text{are rational functions with respect to both } n \text{ and } k,$$

where n is assumed to be an integer, and the sum is to be taken over all integers $k \in \mathbb{Z}$.
To prove a statement of the form (9) by the WZ method[1], one applies Gosper's algorithm to the expression

$$a_k := F(n,k) - F(n-1,k)$$

with respect to the variable k. If successful, this generates $G(n,k)$ with

$$a_k = F(n,k) - F(n-1,k) = G(n,k) - G(n,k-1) , \tag{10}$$

and summing over all k leads to

$$s_n - s_{n-1} = \sum_{k\in\mathbb{Z}} \Big(F(n,k) - F(n-1,k)\Big) = \sum_{k\in\mathbb{Z}} \Big(G(n,k) - G(n,k-1)\Big) = 0$$

since the right hand side is telescoping. Therefore s_n is constant, $s_n = s_0$, and if we are able to prove $s_0 = 1$, we are done. Note that $s_0 = 1$ generally can be proved if the series considered is terminating in which case also no questions concering convergence arise.
Since the WZ method only works if n is an integer, we can try to prove the statements of Bailey's list in Table 1 anyway only if one of the upper parameters of the hypergeometric series involved is a negative integer. The extension to the general case is over the capabilities of the methods of this article, and must be handled by other means.
Note that the rationality of a_k/a_{k-1} for the WZ method is decided by Algorithm 1 since

$$\frac{a_k}{a_{k-1}} = \frac{F(n,k) - F(n-1,k)}{F(n,k-1) - F(n-1,k-1)} = \frac{F(n,k)}{F(n,k-1)} \cdot \frac{1 - \frac{F(n-1,k)}{F(n,k)}}{1 - \frac{F(n-1,k-1)}{F(n,k-1)}} .$$

[1] Note that Wilf and Zeilberger use forward differences rather that downward differences, whereas we decided to follow Gosper's original treatment. There is no theoretical difference between these two approaches, though.

Table 2: The WZ method

Theorem	n	$R(n,k)$
Vandermonde	$-a$	$-\frac{(b+k)(-n+k)}{n(c+n-1)}$
Saalschütz	n	$-\frac{(b+k)(-n+k)(a+k)}{n(c+n-1)(1+a+b-c-n+k)}$
Kummer	$-b$	$\frac{(a+k)(-n+k)}{n(a+2n)}$
Dixon	$-c$	$-\frac{(a+k)(-n+k)(b+k)}{n(a-b+n)(a+2n)}$
Watson Whipple	$-c$	$2\frac{(a+k)(-n+k)(b+k)}{(-1+a+b+2n)(-2n+1+k)(-2n+k)}$
Whipple	$-c$	$-\frac{(a+k)(a-1-k)(-n+k)}{n(2-2n-e+k)(1-2n-e+k)}$
Dougall	n	$\frac{(2a-b-c-d+2n)(a+k)(-n+k)(b+k)(c+k)(d+k)}{n(a+2k)(a-b-c+n-d-k)(a-d+n)(a-c+n)(a-b+n)}$
Dougall	$-e$	$-\frac{(a+k)(-n+k)(c+k)(d+k)}{n(a+2k)(a-c+n)(a-d+n)}$
Whipple	$-e$	$\frac{(d+k)(-n+k)(a+k)}{n(a+2k)(a-d+n)}$
Bailey	n	$-\frac{\left(a^2+2a-wa+na+2-2w-2kw+2ka+2k+2kn\right)(a+k)(-n+k)}{(-w+a+n)n(a+2k)(w+n-1)}$
Bailey	n	$-\frac{\left(-2b-2b^2+2nb+ab-1+n-k\right)(a+k)(-n+k)(b+k)}{nb(1+2b-n+k)(a-2b+2n-2)(a-b+n)}$
Bailey	n	$-\frac{(2b+ab+1-n+2kb+k)(b+k)(-n+k)(a+k)}{nb(a+2k)(1+2b-n+k)(a-b+n)}$
Bailey	n	$-\frac{(a+k)(-n+k)(b+k)}{nb(a+2k)(2+2b-n+k)(a-2b-3+2n)(a-b+n)} \cdot \left(-8b-4b^2+6nb-ab-2n^2+2nba-4+6n-2b^2a+a^2b-6k-8kb-4b^2k+4kn+4kbn+2kba-2k^2\right)$

Note moreover, that the application of Gosper's algorithm may be slow. But as soon as we have found the function $G(n,k)$, we easily can calculate the rational function

$$R(n,k) := \frac{G(n,k)}{F(n,k)}$$

by an application of Algorithm 1. $R(n,k)$ is rational since the proof of Gosper's algorithm shows that $G(n,k)$ is a rational multiple of $a_k = F(n,k) - F(n-1,k)$, $G(n,k) = r(n,k) \cdot (F(n,k) - F(n-1,k))$, say, so that

$$R(n,k) = \frac{G(n,k)}{F(n,k)} = r(n,k)\frac{F(n,k) - F(n-1,k)}{F(n,k)} = r(n,k)\left(1 - \frac{F(n-1,k)}{F(n,k)}\right)$$

is rational. $R(n,k)$ is called the *rational certificate* of $F(n,k)$. Once the rational certificate of a hypergeometric expression $F(n,k)$ is known, it is a matter of pure rational arithmetic (which is fast) to decide the validity of (9) since the only thing that one has to show is (10) which after division by $F(n,k)$ is equivalent (modulo an application of Algorithm 1) to the purely rational identity

$$1 - R(n,k) + R(n,k-1)\frac{F(n,k-1)}{F(n,k)} - \frac{F(n-1,k)}{F(n,k)} = 0 \; . \tag{11}$$

As an example, to prove the Binomial Theorem (compare (8)) in the form

$$s_n := \sum_{k=0}^{n} F(n,k) = \sum_{k=0}^{n} \frac{1}{2^n}\binom{n}{k} = 1 \tag{12}$$

by the WZ method, Algorithm 1 yields

$$\frac{a_k}{a_{k-1}} = \frac{F(n,k) - F(n-1,k)}{F(n,k-1) - F(n-1,k-1)} = \frac{(n-k+1)(n-2\,k)}{k\,(n-2\,k+2)}$$

so that Gosper's algorithm can be applied, and results in

$$G(n,k) = \frac{k}{2^n(2k-n)}\left(2\binom{n-1}{k} - \binom{n}{k}\right) \; .$$

This proves (12) since $s_0 = \sum\limits_{k=0}^{0} 1 = 1$.
The rational certificate function is

$$R(n,k) = \frac{G(n,k)}{F(n,k)} = \frac{k-n}{n} \; ,$$

and the verification of identity (12) is therefore reduced to simplify the rational expression

$$1 - R(n,k) + R(n,k-1)\frac{F(n,k-1)}{F(n,k)} - \frac{F(n-1,k)}{F(n,k)} = 1 - \frac{k-n}{n} + \frac{k-1-n}{n}\frac{k}{n+1-k} - \frac{2(n-k)}{n}$$

to zero.
Table 2 is a complete list of those identities of Bailey's list (Table 1) that can be treated by the given method together with their rational certificates with which the reader may verify them easily.

Note that neither the statements of Gauß and Bailey of argument $x = 1/2$ (p. 11) are accessible with respect to any of the parameters involved, nor can Watson's Theorem (p. 16) be proved by the WZ method with respect to Watson's original integer parameter a, nor can the method be applied to Whipple's Theorem (p. 16) concerning parameters a or b since in all these cases the term ratio a_k/a_{k-1} is not rational.
Note further that both our REDUCE and our MAPLE implementations generate the results of Table 2, and only the calculation of the rational certificate of Dougall's Theorem needs more than a few seconds. In the appendix, we will present some example results. On the other hand, MAPLE's `sum` command was not successful with a single example without preprocessing the input (entered in factorial or Γ notation), in most cases quickly responding with the incorrect statement `Gosper's algorithm fails`, and unsuccessfully trying other methods afterwards.
In § 4, we consider a generalization of the WZ method. To be able to consider the most general case, we present an extended version of Gosper's algorithm next.

4 An Extended Version of the Gosper Algorithm

Here we deal with the question, given a nonnegative integer m, to find a sequence s_k for given a_k satisfying

$$a_k = s_k - s_{k-m} \tag{13}$$

in the particular case that s_k is an *m-fold hypergeometric term*, i. e.

$$\frac{s_k}{s_{k-m}} \quad \text{is a rational function with respect to } k\ . \tag{14}$$

Note that in the given case the input function a_k itself is an m-fold hypergeometric term since by (13) and (14)

$$\frac{a_k}{a_{k-m}} = \frac{s_k - s_{k-m}}{s_{k-m} - s_{k-2m}} = \frac{\frac{s_k}{s_{k-m}} - 1}{1 - \frac{s_{k-2m}}{s_{k-m}}} = \frac{u_k}{v_k}$$

is rational, i. e., u_k and v_k can be chosen to be polynomials.
Assume first, given a_k, we have found s_k with $s_k - s_{k-m} = a_k$. Then we can easily construct an antidifference $\tilde{s}_k$ of a_k by

$$\tilde{s}_k := s_k + s_{k-1} + \cdots + s_{k-(m-1)} \tag{15}$$

since then

$$\tilde{s}_k - \tilde{s}_{k-1} = \left(s_k + \cdots + s_{k-(m-1)}\right) - \left(s_{k-1} + \cdots + s_{k-m}\right) = s_k - s_{k-m} = a_k\ .$$

Note, however, that in general, this antidifference is not a hypergeometric term, but is a finite sum of hypergeometric terms.
Assume next that given a_k there exists a hypergeometric term s_k with $s_k - s_{k-m} = a_k$. Then obviously also

$$\frac{s_k}{s_{k-m}} = \frac{s_k}{s_{k-1}} \cdot \frac{s_{k-1}}{s_{k-2}} \cdots \frac{s_{k-(m-1)}}{s_{k-m}}$$

is rational, and therefore our algorithm below will find s_k.
An m-fold antidifference always can be constructed by an application of Gosper's original algorithm in the following way:

Algorithm 2 (`extended_gosper`)

The following steps generate an m-fold antidifference:

1. Input: a_k, and $m \in \mathbb{N}$.
2. Define $b_k := a_{km}$.
3. Apply Gosper's algorithm to b_k with respect to k. Get the antidifference t_k of b_k, or the statement: "No hypergeometric term antidifference of b_k, and therefore no m-fold hypergeometric term antidifference of a_k exists."
4. The output $s_k := t_{k/m}$ is a solution of (13) with the property (14).

Proof: This is valid since $a_k = s_k - s_{k-m}$ is equivalent to

$$b_k = a_{km} = s_{km} - s_{km-m} = t_k - t_{k-1}$$

and since

$$\frac{t_k}{t_{k-1}} = \frac{s_{km}}{s_{km-m}}$$

describes the transformation between t_k and s_k. $\square$

As an example, we consider $a_k := k\left(\frac{k}{2}\right)!$, and $m = 2$. Then $b_k = a_{2k} = 2k\,k!$, and Gosper's algorithm yields $t_k = 2(k+1)k!$. Therefore $s_k = t_{k/2} = (k+2)\left(\frac{k}{2}\right)!$ has the property that

$$s_k - s_{k-2} = a_k \ .$$

By (15), we moreover find the antidifference

$$\tilde{s}_k = s_k + s_{k-1} = (k+2)\left(\frac{k}{2}\right)! + (k+1)\left(\frac{k-1}{2}\right)!$$

of a_k.

We consider two other examples: If $a_k = \binom{k/3}{n}$ then our algorithm generates the antidifference

$$s_k = \frac{1}{3(n+1)}\left((k+3)\binom{\frac{k}{3}}{n} + (k+2)\binom{\frac{k-1}{3}}{n} + (k+1)\binom{\frac{k-2}{3}}{n}\right) \ ,$$

and if $a_k = \binom{n}{k/2} - \binom{n}{k/2-1}$ then

$$\begin{aligned} s_k \ &= \ \frac{(2n+3-k)(n+1-k)}{2(n+2-k)(n+1-k)}\left(\binom{n}{\frac{k-1}{2}} - \binom{n}{\frac{k-3}{2}}\right) \\ &\quad + \frac{(n+2-k)(2n+2-k)}{2(n+2-k)(n+1-k)}\left(\binom{n}{k/2} - \binom{n}{k/2-1}\right) \ . \end{aligned}$$

Note, however, that we will use m-fold hypergeometric antidifferences rather than non-hypergeometric antidifferences in the later chapters.

Now, we give an algorithm that finds an appropriate nonnegative integer m for an arbitrary input function a_k given as ratio of products of rational functions, exponentials, factorials, Γ function terms, binomial coefficients, and Pochhammer symbols that are rational-linear in their arguments:

Algorithm 3 (`find_mfold`)

The following is an algorithm generating a successful choice for m for an application of Algorithm 2.

1. Input: a_k as ratio of products of rational functions, exponentials, factorials, Γ function terms, binomial coefficients, and Pochhammer symbols that are rational-linear in their arguments.

2. Build the list of all arguments. They are of the form $p_j/q_j\,k+\alpha_j$ with integer p_j and q_j, p_j/q_j in lowest terms, q_j positive.

3. Calculate $m := \mathrm{lcm}\{q_j\}$.

Proof: It is clear that the procedure generates a representation for $b_k = a_{km}$ with the given choice of m which is integer-linear in the arguments involved. Since in this case b_k/b_{k-1} is rational, Algorithm 2 is applicable. $\square$

We mention that in our example cases above, the given procedure yields the desired values $m = 2$ for $a_k := k\left(\frac{k}{2}\right)!$, $m = 3$ for $a_k = \binom{k/3}{n}$, and $m = 2$ for $a_k = \binom{n}{k/2} - \binom{n}{k/2-1}$.

5 Extension of the WZ method

In this section we will give an extended version of the WZ method which resolves the questions that remained open in § 3 so that finally Bailey's complete list (Table 1) can be settled using a unifying approach.

Assume that for a hypergeometric identity the WZ method fails. This may happen either because a_k/a_{k-1} is not rational, or because there is no single formula for the result like in Andrews' statement

$$ {}_3F_2\left(\left.\begin{array}{c} -n\ , n+3a\ , a \\ 3a/2\ , (3a+1)/2 \end{array}\right| \frac{3}{4}\right) = \begin{cases} 0 & \text{if } n \not\equiv 0 \pmod 3 \\ \dfrac{n!\,(a+1)_{n/3}}{(n/3)!\,(3a+1)_n} & \text{otherwise} \end{cases} \tag{16} $$

which—together with many similar statements—can be found in a paper of Gessel and Stanton (1982), Equation (1.1).

In such cases, we proceed as follows: To prove an identity of the form

$$ s_n := \sum_{k\in\mathbb{Z}} F(n,k) = \text{constant} \qquad (n \bmod m \text{ constant})\ , \tag{17} $$

m denoting a certain positive integer, $F(n,k)$ being an (m,l)*-fold hypergeometric term* with respect to (n,k), i. e.

$$ \frac{F(n,k)}{F(n-m,k)} \quad \text{and} \quad \frac{F(n,k)}{F(n,k-l)} \quad \text{are rational functions with respect to both } n \text{ and } k, $$

and n assuming to be an integer, we apply our extended version of Gosper's algorithm to find an l-fold antidifference of the expression

$$ a_k := F(n,k) - F(n-m,k) $$

Table 3: Gessel and Stanton's hypergeometric identities

Eq.	Identity
(1.1)	${}_3F_2\left(\begin{matrix} -n\,,n+3a\,,a \\ 3a/2\,,(3a+1)/2 \end{matrix}\,\middle|\, \frac{3}{4}\right) = \begin{cases} 0 & \text{if } n \not\equiv 0 \pmod 3 \\ \dfrac{n!\,(a+1)_{n/3}}{(n/3)!\,(3a+1)_n} & \text{otherwise} \end{cases}$
(1.2)	${}_5F_4\left(\begin{matrix} 2a\,,2b\,,1-2b\,,1+2a/3\,,-n \\ a-b+1\,,a+b+1/2\,,2a/3\,,1+2a+2n \end{matrix}\,\middle|\, \frac{1}{4}\right) = \dfrac{(a+1/2)_n\,(a+1)_n}{(a+b+1/2)_n\,(a-b+1)_n}$
(1.3)	${}_5F_4\left(\begin{matrix} a, b, a+1/2-b, 1+2a/3, -n \\ 2a+1-2b, 2b, 2a/3, 1+a+n/2 \end{matrix}\,\middle|\, 4\right) = \begin{cases} 0 & \text{if } n \text{ odd} \\ \dfrac{n!\,(a+1)_{n/2}\,2^{-n}}{(\frac{n}{2})!(a-b+1)_{n/2}(b+\frac{1}{2})_{n/2}} & \text{otherwise} \end{cases}$
(1.4)	${}_3F_2\left(\begin{matrix} 1/2+3a\,,1/2-3a\,,-n \\ 1/2\,,-3n \end{matrix}\,\middle|\, \frac{3}{4}\right) = \dfrac{(1/2-a)_n\,(1/2+a)_n}{(1/3)_n\,(2/3)_n}$
(1.5)	${}_3F_2\left(\begin{matrix} 1+3a\,,1-3a\,,-n \\ 3/2\,,-1-3n \end{matrix}\,\middle|\, \frac{3}{4}\right) = \dfrac{(1+a)_n\,(1-a)_n}{(2/3)_n\,(4/3)_n}$
(1.6)	${}_3F_2\left(\begin{matrix} 2a\,,1-a\,,-n \\ 2a+2\,,-a-1/2-3n/2 \end{matrix}\,\middle|\, 1\right) = \dfrac{((n+3)/2)_n\,(n+1)(2a+1)}{(1+(n+2a+1)/2)_n\,(2a+n+1)}$
(1.7)	${}_7F_6\left(\begin{matrix} 2a\,,2b\,,1-2b\,,1+2a/3\,,a+d+n+1/2\,,a-d\,,-n \\ a-b+1\,,a+b+1/2\,,2a/3\,,-2d-2n\,,2d+1\,,1+2a+2n \end{matrix}\,\middle|\, 1\right) =$ $\dfrac{(2a+1)_{2n}\,(b+d+1/2)_n\,(d-b+1)_n}{(2d+1)_{2n}\,(a+b+1/2)_n\,(a-b+1)_n} = \dfrac{(a+1/2)_n\,(a+1)_n\,(b+d+1/2)_n\,(d-b+1)_n}{(a+b+1/2)_n\,(a-b+1)_n\,(d+1/2)_n\,(d+1)_n}$
(1.8)	${}_7F_6\left(\begin{matrix} a\,,b\,,a+1/2-b\,,1+2a/3\,,1-2d\,,2a+2d+n\,,-n \\ 2a-2b+1\,,2b\,,2a/3\,,a+d+1/2\,,1-d-n/2\,,1+a+n/2 \end{matrix}\,\middle|\, 1\right) =$ $\begin{cases} 0 & \text{if } n \text{ odd} \\ \dfrac{(b+d)_{n/2}\,(d-b+a+1/2)_{n/2}\,n!\,(a+1)_{n/2}\,2^{-n}}{(b+1/2)_{n/2}\,(a+d+1/2)_{n/2}\,(d)_{n/2}\,(n/2)!\,(a-b+1)_{n/2}} & \text{otherwise} \end{cases}$
(3.7)	${}_2F_1\left(\begin{matrix} -n\,,-2n-2/3 \\ 4/3 \end{matrix}\,\middle|\, -8\right) = \dfrac{(5/6)_n}{(3/2)_n}\,(-27)^n$
(5.21)	${}_3F_2\left(\begin{matrix} 3a+1/2\,,3a+1\,,-n \\ 6a+1\,,-n/3+2a+1 \end{matrix}\,\middle|\, \frac{4}{3}\right) = \begin{cases} 0 & \text{if } n \not\equiv 0 \pmod 3 \\ \dfrac{(1/3)_{n/3}\,(2/3)_{n/3}}{(1+2a)_{n/3}\,(-2a)_{n/3}} & \text{otherwise} \end{cases}$
(5.22)	${}_2F_1\left(\begin{matrix} -n\,,1/2 \\ 2n+3/2 \end{matrix}\,\middle|\, \frac{1}{4}\right) = \dfrac{(1/2)_n}{(2n+3/2)_n}\left(\dfrac{27}{4}\right)^n$
(5.23)	${}_2F_1\left(\begin{matrix} -n\,,-1/3-2n \\ 2/3 \end{matrix}\,\middle|\, -8\right) = (-27)^n$
(5.24)	${}_2F_1\left(\begin{matrix} -n\,,n/2+1 \\ 4/3 \end{matrix}\,\middle|\, \frac{8}{9}\right) = \begin{cases} 0 & \text{if } n \text{ odd} \\ \dfrac{(1/2)_{n/2}}{(7/6)_{n/2}}\,(-3)^{-(n/2)} & \text{otherwise} \end{cases}$
(5.25)	${}_2F_1\left(\begin{matrix} -n\,,1/2 \\ (n+3)/2 \end{matrix}\,\middle|\, 4\right) = \begin{cases} 0 & \text{if } n \text{ odd} \\ \dfrac{(1/2)_{n/2}\,(3/2)_{n/2}}{(5/6)_{n/2}\,(7/6)_{n/2}} & \text{otherwise} \end{cases}$
(5.27)	${}_4F_3\left(\begin{matrix} 1/3-n\,,-n/2\,,(1-n)/2\,,22/21-3n/7 \\ 5/6\,,4/3\,,1/21-3n/7 \end{matrix}\,\middle|\, -27\right) = \dfrac{(-8)^n}{1-9n}$

Table 4: The extended WZ method

Bailey p.	n	m	$R(n,k)$
11, Gauß	$-a$	2	$-\frac{(b+k)(n-k)}{(-b+n-1-2k)n}$
11, Bailey	$-a$	2	$\frac{(2n-1)(n-k)}{(c+n-1)(n+k)}$
16, Watson	$-a$	2	$-2\frac{(c+k)(b+k)(n-k)}{(-1+n+2c)(-b+n-1-2k)n}$
16, Whipple	$-a$	2	$2\frac{(2n-1)(n-k)(c+k)}{(2c-e+n)(-1+n+e)(n+k)}$
G.-S. Eq.	m	$R(n,k)$	
(1.1)	3	$3\frac{(a+k)(n-k)(3a+2n-3)}{(n+3a+k-2)(n+3a+k-1)n}$	
(1.2)	1	$-\frac{(2a+k)(n-k)(2b-1-k)(2b+k)}{2n(3k+2a)(2a+2b-1+2n)(a-b+n)}$	
(1.3)	2	$4\frac{(b+k)(a+k)(2a-2b+1+2k)(n-k)}{n(3k+2a)(2b-1+n)(2a-2b+n)}$	
(1.4)	1	$3\frac{(n-k)(6a-1-2k)(6a+1+2k)}{(12n-4k)(3n-1-k)(3n-2-k)}$	
(1.5)	1	$3\frac{(n-k)(3a-k-1)(3a+1+k)}{(3n-1-k)(3n-k)(3n-k+1)}$	
(1.6)	2	$\frac{\left(-4a+4an+18n^2-20n+2-16nk\right)(n-k)(a-k-1)(2a+k)}{n(2a+1+3n-2k)(2a-1+3n-2k)(2a-3+3n-2k)(n-1)}$	
(1.7)	1	$\frac{(2a-1+4n+2d)(a-d+k)(2a+k)(n-k)(2b-1-k)(2b+k)}{n(2a+3k)(2d+2n-k)(2d+2n-1-k)(2a+2b-1+2n)(a-b+n)}$	
(1.8)	2	$8\frac{(2d-1-k)(b+k)(n-k)(a+k)(2a-2b+2k+1)(a+n+d-1)}{n(2a+3k)(-2+2d+n-2k)(2b-1+n)(2a-2b+n)(2a+2d+n+k-1)}$	
(3.7)	1	$4\frac{(n-k)(6n+2-3k)(7n-1-3k)}{(3n+1)(1+2n)n}$	
(5.21)	3	$2\frac{(3a+1+k)(6a+2k+1)(n-k)}{n(6a+n)(-n+6a+3+3k)}$	
(5.22)	1	$\frac{(5+6k)(1+2k)(n-k)}{(24n+4)(6n-1)n}$	
(5.23)	1	$4\frac{(21n-7-9k)(6n+1-3k)(n-k)}{(6n+1)(3n-1)n}$	
(5.24)	2	$4\frac{n-k}{1+3n}$	
(5.25)	2	$\frac{(4n-4k)(1+2k)(2+3k)}{n(3n-1)(1+3n)}$	
(5.27)	1	$81\frac{(n-1-2k)(n-2k)(-1+3n-3k)}{n(3n-1)(-1+9n-21k)}$	

with respect to the variable k. (In most cases $l = 1$, so that Gosper's original algorithm is applied.) If successful, this generates $G(n, k)$ with

$$a_k = F(n,k) - F(n-m,k) = G(n,k) - G(n,k-l)\,, \tag{18}$$

and summing over all k leads to

$$s_n - s_{n-m} = \sum_{k\in\mathbb{Z}} \Big(F(n,k) - F(n-m,k)\Big) = \sum_{k\in\mathbb{Z}} \Big(G(n,k) - G(n,k-l)\Big) = 0$$

since the right hand side is telescoping. Therefore s_n is constant for constant n mod m, and these constants can be calculated using suitable initial values. This can be accomplished if the series considered is terminating. Note, that again, the function

$$R(n,k) = \frac{G(n,k)}{F(n,k)} \tag{19}$$

acts as a rational certificate function. Once the rational certificate is known, it is a matter of pure rational arithmetic to decide the validity of (17) since the only thing that one has to show is (18) which after division by $F(n,k)$ is equivalent to the purely rational identity

$$1 - R(n,k) + R(n,k-l)\frac{F(n,k-l)}{F(n,k)} - \frac{F(n-m,k)}{F(n,k)} = 0\,.$$

As an example, we prove (16): In the given case, we set $m := 3$, $l := 1$, further

$$F(n,k) := \frac{(-n)_k\,(n+3a)_k\,(a)_k}{k!\,(3a/2)_k\,((3a+1)/2)_k}\frac{(n/3)!\,(3a+1)_n}{n!\,(a+1)_{n/3}}\left(\frac{3}{4}\right)^k\,,$$

and notice that

$$\frac{F(n,k)}{F(n,k-1)} \quad \text{and} \quad \frac{F(n,k)}{F(n-3,k)}$$

are (complicated) rational functions (Algorithms 2 and 1). An application of Gosper's algorithm is successful, and leads to the rational certificate

$$R(n,k) = 3\frac{(a+k)\,(n-k)\,(3\,a+2\,n-3)}{(n+3\,a+k-2)\,(n+3\,a+k-1)\,n}\,.$$

Therefore

$$\sum_{k\in\mathbb{Z}} F(n,k) = \sum_{k=0}^{n} F(n,k) = \text{constant} \qquad (n \bmod 3 \text{ constant})\,,$$

and statement (16) follows using three trivial initial values.

Table 3 lists the hypergeometric identities of the Gessel-Stanton paper (note the misprint in Equation (1.4)), and Table 4 contains their rational certificates (19), calculated by our implementations, together with the certificates of Bailey's list (Table 1) to which the WZ method did not apply.

Note that in all cases considered, $l = 1$, so that the original Gosper algorithm is applied.

Note, moreover, that Gessel and Stanton were not able to present proofs for their statements (6.2), (6.3), (6.5), and (6.6): Table 5 contains proofs.

Table 5: Gessel and Stanton's open problems

Eq.	Identity
(6.2)	${}_7F_6\left(\begin{matrix} a+1/2\,,a\,,b\,,1-b\,,-n\,,(2a+1)/3+n\,,a/2+1 \\ 1/2\,,(2a-b+3)/3\,,(2a+b+2)/3\,,-3n\,,2a+1+3n\,,a/2 \end{matrix}\middle\vert\, 1\right) = \dfrac{((2a+2)/3)_n\,(2a/3+1)_n\,((1+b)/3)_n\,((2-b)/3)_n}{((2a-b)/3+1)_n\,((2a+b+2)/3)_n\,(2/3)_n\,(1/3)_n}$
(6.3)	${}_5F_4\left(\begin{matrix} a+1/2\,,a\,,-n\,,(2a+1)/3+n\,,a/2+1 \\ 1/2\,,-3n\,,2a+1+3n\,,a/2 \end{matrix}\middle\vert\, 9\right) = \dfrac{((2a+2)/3)_n\,(2a/3+1)_n}{(2/3)_n\,(1/3)_n}$
(6.5)	${}_2F_1\left(\begin{matrix} -n\,,-n+1/4 \\ 2n+5/4 \end{matrix}\middle\vert\, \frac{1}{9}\right) = \dfrac{(5/4)_{2n}}{(2/3)_n\,(13/12)_n}\left(\dfrac{2^6}{3^5}\right)^n$
(6.6)	${}_2F_1\left(\begin{matrix} -n\,,-n+1/4 \\ 2n+9/4 \end{matrix}\middle\vert\, \frac{1}{9}\right) = \dfrac{(9/4)_{2n}}{(4/3)_n\,(17/12)_n}\left(\dfrac{2^6}{3^5}\right)^n$

Rational certificates

Eq.	m	$R(n,k)$
(6.2)	1	$6\dfrac{(a-1+3\,n)\,(a+k)\,(2\,a+2\,k+1)\,(n-k)\,(b-1-k)\,(b+k)}{(a+2\,k)\,(3\,n-k)\,(3\,n-1-k)\,(3\,n-2-k)\,(2\,a-b+3\,n)\,(2\,a+b-1+3\,n)}$
(6.3)	1	$-\dfrac{(6\,a-6+18\,n)\,(n-k)\,(2\,a+2\,k+1)\,(a+k)}{(a+2\,k)\,(3\,n-k)\,(3\,n-1-k)\,(3\,n-2-k)}$
(6.5)	1	$-\dfrac{\left(52\,n^2-13\,n-21-56\,k+16\,nk-32\,k^2\right)(n-k)\,(4\,n-1-4\,k)}{(108\,n-27)\,(3\,n-1)\,(1+12\,n)\,n}$
(6.6)	1	$-\dfrac{\left(52\,n^2+39\,n-55-84\,k+16\,nk-32\,k^2\right)(4\,n-1-4\,k)\,(n-k)}{(108\,n-27)\,(1+3\,n)\,(5+12\,n)\,n}$

Similarly as the original WZ approach, our method is not capable, however, to prove Gessel-Stanton's (6.1), a non-terminating version of (6.2). Also, Gessel-Stanton's result (1.9)

$$ {}_3F_2\left(\left.\begin{array}{c} -sb+s+1\,,b-1\,,-n \\ b+1\,,s(-n-b)-n \end{array}\right| 1\right) = \frac{(1+s+sn)_n\,b\,(n+1)}{(1+s(b+n))_n\,(b+n)} $$

is over the capabililites of our method since in this case the summand is an (m,l)-fold hypergeometric term only for fixed (rational), but not for arbitrary s.
Note, that our method not only unifies the proof of hypergeometric identities in a stronger fashion than the original WZ approach but moreover our Reduce and Maple implementations do all the necessary computations completely automatically. We present some of the input and output in the appendix.
Finally, we give examples of an application for which $l \neq 1$. To prove the identity ($n \in \mathbb{N}$)

$$ -\sum_{k=0}^{n}(-2)^n \binom{n}{k}\cdot\binom{k/2}{n} = 1\,, \tag{20} $$

we apply our extended WZ method with $l=2$, and $m=1$, and get the rational certificate

$$ R(n,k) = \frac{(-k+n-1)\,(-k+n)}{(n-1)\,(-k+2\,n-2)}\,, $$

which proves (20). Similarly one proves the statement ($n \in \mathbb{N}_0$)

$$ \sum_{k=0}^{n}(-1)^k\,(-2)^n \binom{n}{k}\cdot\binom{k/2}{n} = 1\,. $$

6 The Zeilberger Algorithm

In this section, we recall the celebrated Zeilberger algorithm (Zeilberger, 1990–1991), see also Graham, Knuth and Patashnik (1994) with which one can not only verify hypergeometric identities but moreover definite sums can be calculated if they represent hypergeometic terms. Zeilberger's algorithm deals with the question to determine a *holonomic recurrence equation*

$$ \sum_{j=0}^{J} P_j(n)\,\Sigma(n-j) = 0 \tag{21} $$

with polynomials P_j in n, for sums

$$ \Sigma(n) := \sum_{k\in\mathbb{Z}} F(n,k) \tag{22} $$

for which $F(n,k)$ is a hypergeometric term with respect to both n and k.
Zeilberger's idea is to apply Gosper's algorithm in the following non-obvious way: Set

$$ a_k := F(n,k) + \sum_{j=1}^{J} \sigma_j(n)\,F(n-j,k) $$

with yet undetermined variables σ_j depending on n, but not depending on k. Then

$$\frac{a_k}{a_{k-1}} = \frac{F(n,k) + \sum\limits_{j=1}^{J} \sigma_j(n)\, F(n-j,k)}{F(n,k-1) + \sum\limits_{j=1}^{J} \sigma_j(n)\, F(n-j,k-1)} = \frac{F(n,k)}{F(n,k-1)} \cdot \frac{1 + \sum\limits_{j=1}^{J} \sigma_j(n)\, \frac{F(n-j,k)}{F(n,k)}}{1 + \sum\limits_{j=1}^{J} \sigma_j(n)\, \frac{F(n-j,k-1)}{F(n,k-1)}}$$

turns out to be rational with respect to k, so the Gosper algorithm may be applied.
If an application of Gosper's algorithm is successful it provides us with s_k depending on n, and a set of rational functions $\sigma_j(n)$ (the coefficients of f_k are determined together with the unknowns σ_j) such that

$$s_k - s_{k-1} = a_k = F(n,k) + \sum_{j=1}^{J} \sigma_j(n)\, F(n-j,k)\ ,$$

so that by summation

$$\begin{aligned} \sum_{k\in\mathbb{Z}} a_k &= \sum_{k\in\mathbb{Z}} \left(F(n,k) + \sum_{j=1}^{J} \sigma_j(n)\, F(n-j,k) \right) \\ &= \Sigma(n) + \sum_{j=1}^{J} \sigma_j(n)\, \Sigma(n-j) = \sum_{k\in\mathbb{Z}} \Big(s_k - s_{k-1} \Big) = 0 \end{aligned}$$

since the right hand side is a telescoping sum. After multiplication with the common denominator this establishes the recurrence equation (21) searched for.
Koornwinder (1993) gives a rigorous description of Zeilberger's algorithm in the (most common) case that the summation bounds are natural: $a_{-1} = a_{n+1} = 0$, i. e. the summation is for $k = 0 \ldots n$.
Like for the Wilf-Zeilberger method, the Zeilberger algorithm is accompanied by a rational certification mechanism.
Note that Zeilberger's algorithm can be applied to ratios of products of rational functions, exponentials, factorials, Γ function terms, binomial coefficients, and Pochhammer symbols that are integer-linear in their arguments with respect to both n and k.
In the next section we will present a modified version of Zeilberger's algorithm that is applicable if the arguments of such expressions are rational-linear with respect to n and k.
The application of Zeilberger's algorithm has the advantage over the WZ method that the right hand side of the hypergeometric identity does not have to be known in advance, but is *generated* by the algorithm (not to speak of the possibility to verify identities of other type). Therefore, Zeilberger's algorithm can be used to calculate definite sums rather than only verifying them. All identities mentioned in this article which could be verified with the WZ method, can be generated by Zeilberger's algorithm.
Implementations of the Zeilberger algorithm were given by Zeilberger (1990) and Koornwinder (1993) in MAPLE, and by Paule and Schorn (1994) in MATHEMATICA. On the lines of Koornwinder (1993), we implemented the Zeilberger algorithm in REDUCE (Koepf, 1994) and MAPLE, examples of which are given in the appendix.
Note the following side conditions of the previous implementations:

- Zeilberger: Here one must write the input into a file rather than on the command line. Supports only integer-linear input of a special form.

- Koornwinder: Supports only integer-linear input in hypergeometric notation.
- Paule-Schorn: Supports only ratios of rational functions, products of exponentials, factorials, and binomial coefficients.

Our implementations support the input in factorial-binomial-Gamma-Pochhammer as well as hypergeometric notation, and use Algorithm 1 for rationality decisions, and are therefore not bound to integer-linear input.

7 An Extended Version of Zeilberger's Algorithm

Our extended version of the Zeilberger algorithm deals with the question to determine a holonomic recurrence equation (21) for sums (22) for which $F(n,k)$ is an (m,l)-fold hypergeometric term with respect to (n,k), see § 5.
In particular, this applies to all cases when the input function $F(n,k)$ is given as a ratio of products of rational functions, exponentials, factorials, Γ function terms, binomial coefficients, and Pochhammer symbols that are rational-linear in their arguments with respect to both n, and k.
First of all we mention that Zeilberger's algorithm may be applicable even though this is safely the case only if the arguments are integer-linear. An example of that type is the function

$$\Sigma(n) := {}_2F_1\left(\left.\begin{array}{cc} -n/2\,, & -n/2+1/2 \\ & b+1/2 \end{array}\right| 1\right) = \sum_{k=0}^{\infty} \frac{(-n/2)_k\,(-n/2+1/2)_k}{k!\,(b+1/2)_k}\ ,$$

for which an application of Zeilberger's algorithm yields the recurrence equation

$$(2b+n-1)\,\Sigma(n) - 2(b+n-1)\,\Sigma(n-1) = 0\ ,$$

and therefore the explicit representation

$$\Sigma(n) = \frac{2^n\,(b)_n}{(2b)_n}\ .$$

Zeilberger's algorithm applies since $F(n,k)/F(n-1,k)$ and $F(n,k)/F(n,k-1)$ are rational even though the representing expression for $F(n,k)$ is not integer-linear in its arguments.
On the other hand, not for every $F(n,k)$ given with rational-linear Γ-arguments, the Zeilberger algorithm is applicable. An example for this situation is the left hand side of the Watson theorem with respect to variable a (see Table 1).
We present now an algorithm which can be applied for arbitrary rational-linear input.

Algorithm 4 (`extended_sumrecursion`)

The following steps perform an algorithm to determine a holonomic recurrence equation (21) for sums (22).

1. Input: $F(n,k)$, given as a ratio of products of rational functions, exponentials, factorials, Γ function terms, binomial coefficients, and Pochhammer symbols with rational-linear arguments in n and k.
2. Build the list of all arguments. They are of the form $p_j/q_j\,n + s_j/t_j\,k + \alpha_j$ with integer p_j, q_j, s_j, t_j, p_j/q_j and s_j/t_j in lowest terms, q_j and t_j positive.

3. Calculate $m := \mathrm{lcm}\{q_j\}$ and $l := \mathrm{lcm}\{t_j\}$.
4. Define $\tilde{F}(n,k) := F(mn, kl)$. Then $\tilde{F}(n,k)$ is integer-linear in the arguments.
5. Apply Zeilberger's algorithm to $\tilde{F}(n,k)$. Get the recurrence equation

$$\sum_{j=0}^{J} P_j(n)\, \tilde{\Sigma}(n-j) = 0$$

with polynomials P_j in n, for the sum

$$\tilde{\Sigma}(n) := \sum_{k\in\mathbb{Z}} \tilde{F}(n,k) \; .$$

6. The output is the recurrence equation

$$\sum_{j=0}^{J} P_j(n/m)\, \Sigma(n-jm) = 0$$

for the sum

$$\Sigma(n) := \sum_{k\in\mathbb{Z}} F(n,k) \; .$$

Proof: Obviously our construction provides us with $\tilde{F}(n,k)$ integer-linear in the arguments involved. Therefore Zeilberger's algorithm can be applied, and the result follows. □

As a first example, we apply our algorithm to the Watson function

$$\Sigma(n) = {}_3F_2\left(\begin{matrix} -n \,, b \,, c \\ (-n+b+1)/2 \,, 2c \end{matrix} \middle| 1\right)$$

with respect to the variable n to which Zeilberger's algorithm does not apply. In this case, the algorithm determines $m = 2$ and $l = 1$, and leads to the two-fold recurrence equation

$$(b-2c-n+1)\,(n-1)\,\Sigma(n-2) - (b-n+1)\,(2c+n-1)\,\Sigma(n) = 0$$

from which the explicit right hand representation listed in Table 1 can be deduced for integer n since for positive values of n the Watson sum is finite, and therefore

$$\Sigma(0) = 1 \; ,$$

and

$$\Sigma(1) = 1 + \frac{-1\, b\, c}{1\,(b/2)\,(2c)} = 0 \; .$$

It turns out that our method is applicable to all identities considered in this paper to which Zeilberger's original approach does not apply.
For example, we consider the three major identities of the paper of Gessel and Stanton (1982): The evaluation of (1.7)

$$\begin{aligned} \Sigma(n) \;&:=\; {}_7F_6\left(\begin{matrix} 2a \,, 2b \,, 1-2b \,, 1-2a/3 \,, a+d+n+1/2 \,, a-d \,, -n \\ a-b+1 \,, a+b+1/2 \,, 2a/3 \,, -2d-2n \,, 2d+1 \,, 1+2a+2n \end{matrix} \middle| 1\right) \\ &=\; \frac{(a+1/2)_n\,(a+1)_n\,(b+d+1/2)_n\,(d-b+1)_n}{(a+b+1/2)_n\,(a-b+1)_n\,(d+1/2)_n\,(d+1)_n} \end{aligned}$$

is found by a direct application of Zeilberger's algorithm with respect to n, leading to the equivalent recurrence equation

$$\begin{aligned} 0 &= (2a+2b+2n-1)(a-b+n)(2d+2n-1)(d+n)\Sigma(n) \\ &\quad +(2a+2n-1)(a+n)(2b+2d+2n-1)(b-d-n)\Sigma(n-1)\,. \end{aligned}$$

On the other hand, the evaluation of (1.8)

$$\begin{aligned} \Sigma(n) &:= {}_7F_6\left(\left.\begin{array}{c} a\,,b\,,a+1/2-b\,,1+2a/3\,,1-2d\,,2a+2d+n\,,-n \\ 2a-2b+1\,,2b\,,2a/3\,,a+d+1/2\,,1-d-n/2\,,1+a+n/2 \end{array}\right| 1\right) \\ &= \begin{cases} 0 & \text{if } n \text{ odd} \\ \dfrac{(b+d)_{n/2}\,(d-b+a+1/2)_{n/2}\,n!\,(a+1)_{n/2}}{(b+1/2)_{n/2}\,(a+d+1/2)_{n/2}\,(d)_{n/2}\,(n/2)!\,(a-b+1)_{n/2}} & \text{otherwise} \end{cases} \end{aligned}$$

cannot be handled with respect to n using Zeilberger's algorithm, but the extended version leads to the equivalent 2-fold recurrence equation

$$\begin{aligned} 0 &= (n-1+2d+2a)(2b-n-2a)(n-1+2b)(n-2+2d)\Sigma(n) \\ &\quad +(n-1+2d-2b+2a)(n-2+2d+2b)(2a+n)(n-1)\Sigma(n-2)\,. \end{aligned}$$

A direct application of Zeilberger's algorithm is possible, however, with respect to the other variables involved (even with respect to a).
Gessel-Stanton's open problem (6.2)

$$\begin{aligned} \Sigma(n) &:= {}_7F_6\left(\left.\begin{array}{c} a+1/2\,,a\,,b\,,1-b\,,-n\,,(2a+1)/3+n\,,a/2+1 \\ 1/2\,,(2a-b+3)/3\,,(2a+b+2)/3\,,-3n\,,2a+1+3n\,,a/2 \end{array}\right| 1\right) \\ &= \frac{((2a+2)/3)_n\,(2a/3+1)_n\,((1+b)/3)_n\,((2-b)/3)_n}{((2a-b)/3+1)_n\,((2a+b+2)/3)_n\,(2/3)_n\,(1/3)_n}\,, \end{aligned}$$

again, can be solved directly with Zeilberger's algorithm leading to the recurrence equation

$$\begin{aligned} 0 &= (2a+b+3n-1)\,(2a-b+3n)\,(3n-1)\,(3n-2)\,\Sigma(n) \\ &\quad +(2a+3n-1)\,(2a+3n)\,(b+3n-2)\,(b-3n+1)\,\Sigma(n-1)\,. \end{aligned}$$

Finally, as an example with $l \neq 1$, we consider (20), again. Our algorithm generates $m = 1$ and $l = 2$, and the recurrence equations

$$\Sigma(n) - \Sigma(n-1) = 0 \qquad \text{and} \qquad 2\Sigma(n) + \Sigma(n-1) = 0$$

for

$$\Sigma(n) := (-2)^n \binom{n}{k} \cdot \binom{k/2}{n}\,, \qquad \text{and} \qquad \Sigma(n) := \binom{n}{k} \cdot \binom{k/2}{n}\,,$$

respectively.

8 Deduction of hypergeometric identities

Finally, we mention that with a good implementation of Zeilberger's algorithm and our extension at hand, it is easy to *discover* new identities. Just for fun, we realized the pattern in

Andrews' statement (16), and tried to generate similar ones: It turns out that

$$ {}_3F_2\left(\begin{array}{c} -n\,,n+2a\,,a \\ 2a/2\,,(2a+1)/2 \end{array}\middle|\,\frac{2}{4}\right) = \begin{cases} 0 & \text{if } n \text{ odd} \\ \dfrac{(-1)^{n/2}\,(1/2)_{n/2}}{(1/2+a)_{n/2}} & \text{otherwise} \end{cases} $$

and

$$ {}_3F_2\left(\begin{array}{c} -n\,,n+4a\,,a \\ 4a/2\,,(4a+1)/2 \end{array}\middle|\,\frac{4}{4}\right) = \begin{cases} 0 & \text{if } n \text{ odd} \\ \dfrac{(1/2)_{n/2}}{(1/2+2a)_{n/2}} & \text{otherwise} \end{cases} . $$

Another example of a more deductive strategy is: Applying Zeilberger's algorithm to the general ${}_2F_1$ polynomial

$$ \Sigma(n) := {}_2F_1\left(\begin{array}{c} a\,,-n \\ b \end{array}\middle|\,x\right) , $$

e. g., leads to the recurrence equation

$$ (b-1+n)\,\Sigma(n) + (-2n+xn+xa-b+2-x)\,\Sigma(n-1) - (x-1)\,(n-1)\,\Sigma(n-2) = 0\,. $$

It is therefore hypergeometric only if the coefficient of $\Sigma(n-2)$ is identical zero, i. e. if $x=1$. This gives Vandermonde's identity. However, the coefficient of $\Sigma(n-1)$ can be made zero (equating coefficients), if we choose $x=2$, and $b=2a$, in which situation we get

$$ (n+2a-1)\,\Sigma(n) - (n-1)\,\Sigma(n-2) = 0\,. $$

Therefore we have deduced the identity

$$ {}_2F_1\left(\begin{array}{c} a\,,-n \\ 2a \end{array}\middle|\,2\right) = \begin{cases} 0 & \text{if } n \text{ odd} \\ \dfrac{(1/2)_{n/2}}{(1/2+a)_{n/2}} & \text{otherwise} \end{cases} . $$

We see that this method, to some extent, can be a substitute for the ingenuity of people like Dougall, Bailey, Andrews, Gessel or Stanton to find hypergeometric sums which can be represented by single hypergeometric terms.
We finally give a strange example to demonstrate that our method can be of great help to find new identities.
We try to find all hypergeometric functions of the form

$$ \Sigma(n) := {}_2F_1\left(\begin{array}{c} a\,,-n \\ n+b \end{array}\middle|\,x\right) $$

for which a, b and x are constants with respect to n, and for which a recurrence equation with only two terms $\Sigma(n-j)$ is valid.
The recurrence equation for $\Sigma(n)$ turns out to be

$$ \begin{aligned} 0 \;=\; & -(x-1)^2\,(n-1)\,(n-1+b)\,(n-2+b)\,(xn+n-xa-x+bx)\,\Sigma(n-2) \\ & +(n-1+b)\,P(n,a,b,x)\,\Sigma(n-1) \\ & +x\,(2\,n+b-1)\,(2\,n+b-2)\,(n-a-1+b)\,(xn+n-xa-2\,x-1+bx)\,\Sigma(n)\,, \end{aligned} $$

where $P(n,a,b,x)$ denotes a very complicated polynomial of degree 2 in n that does not have a rational factorization. To receive a recurrence equation for which only two terms $\Sigma(n-j)$

different from zero occur, we may set the coefficient lists with respect to n of any of the factors occurring zero, and try to solve for a, b and x. Note that since the resulting equations systems are polynomial systems, by Gröbner bases methods these can be solved algorithmically.
In our case, we receive either $x = 1$, i. e. the recurrence equation

$$\begin{aligned} 0 \;\; = \;\; & -(n-1+b)(2\,n-a+b-1)(b-a+2\,n-2)\,\Sigma(n-1) \\ & +(2\,n+b-1)(2\,n+b-2)(n-a-1+b)\,\Sigma(n)\,, \end{aligned}$$

or we are led to the Kummer identity, i. e. to the values $b = a + 1$ and $x = -1$ with the recurrence equation

$$-2\,(n+a)\,\Sigma(n-1) + (2\,n+a)\,\Sigma(n) = 0\,.$$

The only exception occurs when we set the coefficient list with respect to n of the factor $P(n, a, b, x)$ zero, leading to the Kummer case again, and to the second solution set

$$\{a = 1/2, b = 3/2, x^2 - 6x + 1 = 0\}\,.$$

For $x = 3 \pm 2\sqrt{2}$, we have the recurrence equation

$$-4\,(2\,n-1)(2\,n+1)\,\Sigma(n-2) + (4\,n-1)(4\,n+1)\,\Sigma(n) = 0$$

leading to the closed form representations

$$ {}_2F_1\left(\left.\begin{array}{c} 1/2\,,-n \\ n+3/2 \end{array}\right| 3+2\sqrt{2}\right) = \begin{cases} \dfrac{2\,(5/4)_{(n-1)/2}\,(7/4)_{(n-1)/2}}{5\,(11/8)_{(n-1)/2}\,(13/8)_{(n-1)/2}}\,(1-\sqrt{2}) & \text{if } n \text{ odd} \\ \dfrac{(3/4)_{n/2}\,(5/4)_{n/2}}{(7/8)_{n/2}\,(9/8)_{n/2}} & \text{otherwise} \end{cases} $$

and

$$ {}_2F_1\left(\left.\begin{array}{c} 1/2\,,-n \\ n+3/2 \end{array}\right| 3-2\sqrt{2}\right) = \begin{cases} \dfrac{2\,(5/4)_{(n-1)/2}\,(7/4)_{(n-1)/2}}{5\,(11/8)_{(n-1)/2}\,(13/8)_{(n-1)/2}}\,(1+\sqrt{2}) & \text{if } n \text{ odd} \\ \dfrac{(3/4)_{n/2}\,(5/4)_{n/2}}{(7/8)_{n/2}\,(9/8)_{n/2}} & \text{otherwise} \end{cases}\,, $$

in particular, for even n, the values at $x = 3 + 2\sqrt{2}$ and $x = 3 - 2\sqrt{2}$ are rational and equal:

$$ {}_2F_1\left(\left.\begin{array}{c} 1/2\,,-2n \\ 2n+3/2 \end{array}\right| 3\pm 2\sqrt{2}\right) = \sum_{k=0}^{2n} (-1)^k \frac{\binom{2n}{k}\binom{2n+k+1}{k}}{\binom{4n+2k+2}{2k}} \left(3 \pm 2\sqrt{2}\right)^k = \frac{(3/4)_n\,(5/4)_n}{(7/8)_n\,(9/8)_n}\,. $$

We will discuss the given method in greater detail in a forthcoming paper.

Acknowledgement

I like to thank Gregor Stölting for his work on the implementations, Tom Koornwinder for his wonderful implementation `zeilb` which was the starting point of our implementations, and Peter Deuflhard who initiated my studies on the given topic.

Appendix

In this appendix, we give a short description of a MAPLE implementation, which I implemented together with Gregor Stölting on the lines of Koornwinder (1993), incorporating Gosper's and Zeilberger's algorithms and the extensions of this article, and present some of its results. Our REDUCE implementation is described elsewhere (Koepf, 1994).
After loading our package, one can use the following MAPLE functions:

`gosper(f,k)` determines a closed form antidifference. If it does not return a closed form solution, then a closed form solution does not exist.

`gosper(f,k,m,n)` determines

$$\sum_{k=m}^{n} a_k$$

using Gosper's algorithm. This is only successful if Gosper's algorithm applies.

`extended_gosper(f,k,m)` determines an m-fold hypergeometric antidifference. If it does not return a solution, then such a solution does not exist.

`sumrecursion(f,k,n)` determines a holonomic recurrence equation for $\mathtt{summ}(n) = \sum\limits_{k=-\infty}^{\infty} f(n,k)$ with respect to n if $f(n,k)$ is hypergeometric with respect to both n and k. The resulting expression equals zero.

`sumrecursion(f,k,n,j)` searches only for a holonomic recurrence equation of order j.

`extended_sumrecursion(f,k,n,m,l)` determines a holonomic recurrence equation for $\mathtt{summ}(n) = \sum\limits_{k=-\infty}^{\infty} f(n,k)$ with respect to n if $f(n,k)$ is an (m,l)-fold hypergeometric term with respect to (n,k).

`hyperrecursion(upper,lower,x,n)` determines a holonomic recurrence equation with respect to n for ${}_pF_q\left(\begin{array}{cccc} a_1, & a_2, & \cdots, & a_p \\ b_1, & b_2, & \cdots, & b_q \end{array} \middle| \, x\right)$, where $\mathtt{upper} = \{a_1, a_2, \ldots, a_p\}$ is the list of upper parameters, and $\mathtt{lower} = \{b_1, b_2, \ldots, b_q\}$ is the list of lower parameters depending on n.

`hyperrecursion(upper,lower,x,n,j)` searches only for a holonomic recurrence equation of order j.

`hyperterm(upper,lower,x,k)` yields the hypergeometric term

$$\frac{(a_1)_k \cdot (a_2)_k \cdots (a_p)_k}{(b_1)_k \cdot (b_2)_k \cdots (b_q)_k \, k!} x^k$$

with upper parameters $\mathtt{upper} = \{a_1, a_2, \ldots, a_p\}$, and lower parameters $\mathtt{lower} = \{b_1, b_2, \ldots, b_q\}$

`simplify_gamma(f)` simplifies an expression `f` involving only rational functions, exponentials and Γ function terms according to a recursive application of the simplification rule $\Gamma(a+1) = a\,\Gamma(a)$ to the expression tree, see Algorithm 1.

simplify_combinatorial(f) simplifies an expression **f** involving exponentials, factorials, Γ function terms, binomial coefficients, and Pochhammer symbols by converting factorials, binomial coefficients, and Pochhammer symbols into Γ function terms, and applying **simplify_gamma** and **simplify_power** to its result. If the output is not rational, it is given in terms of Γ functions, see Algorithm 1.

The MAPLE function

```
WZ:=proc(summand,k,n,m)
local tmp,gos;
   tmp:=summand-subs(n=n-m,summand);
   gos:=extended_gosper(tmp,k,m);
   RETURN(simplify_combinatorial(gos/summand))
end:
```

therefore, calculates the (m, l)-fold rational certificate (19) of $F(n, k)$.
Here are some results of the implementation:

```
    |\^/|     Maple V Release 3 (FU-Berlin)
._|\|   |/|_. Copyright (c) 1981-1994 by Waterloo Maple Software and the
 \  MAPLE  /  University of Waterloo. All rights reserved. Maple and Maple V
 <____ ____>  are registered trademarks of Waterloo Maple Software.
      |       Type ? for help.
> read summation;

> # see (SIAM Review, 1994, Problem 94-2)

> gosper((-1)^(k+1)*(4*k+1)*(2*k)!/(k!*4^k*(2*k-1)*(k+1)!),k);
                                          (k + 1)
                                (2 k)! (-1)
                                ------------------
                                             k
                                  (k + 1)! 4  k!
> # Dougall

> WZ(hyperterm({a,1+a/2,b,c,d,1+2*a-b-c-d+n,-n},
  {a/2,1+a-b,1+a-c,1+a-d,1+a-(1+2*a-b-c-d+n),1+a+n},1,k)/
  hyperterm({1+a,1+a-b-c,1+a-b-d,1+a-c-d,1},
  {1+a-b,1+a-c,1+a-d,1+a-b-c-d},1,n),k,n,1);

       (2 a - d + 2 n - c - b) (a + k) (- k + n) (b + k) (c + k) (d + k)
   - ---------------------------------------------------------------------
      n (a + 2 k) (a - b - c - d + n - k) (a - b + n) (a - d + n) (a - c + n)

> sumrecursion(hyperterm({a,1+a/2,b,c,d,1+2*a-b-c-d+n,-n},
  {a/2,1+a-b,1+a-c,1+a-d,1+a-(1+2*a-b-c-d+n),1+a+n},1,k)/
  hyperterm({1+a,1+a-b-c,1+a-b-d,1+a-c-d,1},
  {1+a-b,1+a-c,1+a-d,1+a-b-c-d},1,n),k,n);

                              summ(n) - summ(n - 1)

> hyperrecursion({a,1+a/2,b,c,d,1+2*a-b-c-d+n,-n},
  {a/2,1+a-b,1+a-c,1+a-d,1+a-(1+2*a-b-c-d+n),1+a+n},1,n);
```

```
    - (a + n) (a - c - d + n) (a - b - d + n) (a - b - c + n) summm(n - 1)

         + summm(n) (a - d + n) (a - c + n) (a - b + n) (a - b - c - d + n)

> # Gessel-Stanton (6.2)

> WZ(hyperterm({a+1/2,a,b,1-b,-n,(2*a+1)/3+n,a/2+1},
  {1/2,(2*a-b+3)/3,(2*a+b+2)/3,-3*n,2*a+1+3*n,a/2},1,k)/
  hyperterm({(2*a+2)/3,2*a/3+1,(1+b)/3,(2-b)/3,1},
  {(2*a-b)/3+1,(2*a+b+2)/3,2/3,1/3},1,n),k,n);

   6 (a - 1 + 3 n) (a + k) (2 a + 2 k + 1) (- k + n) (b - 1 - k) (b + k)/(

       (a + 2 k) (3 n - k) (3 n - 1 - k) (3 n - 2 - k) (2 a - b + 3 n)

       (2 a + b - 1 + 3 n))

> sumrecursion(hyperterm({a+1/2,a,b,1-b,-n,(2*a+1)/3+n,a/2+1},
  {1/2,(2*a-b+3)/3,(2*a+b+2)/3,-3*n,2*a+1+3*n,a/2},1,k)/
  hyperterm({(2*a+2)/3,2*a/3+1,(1+b)/3,(2-b)/3,1},
  {(2*a-b)/3+1,(2*a+b+2)/3,2/3,1/3},1,n),k,n);

                            summm(n) - summm(n - 1)

> hyperrecursion({a+1/2,a,b,1-b,-n,(2*a+1)/3+n,a/2+1},
  {1/2,(2*a-b+3)/3,(2*a+b+2)/3,-3*n,2*a+1+3*n,a/2},1,n);

    - (3 n - 2 + b) (3 n - 1 - b) (2 a + 3 n) (2 a - 1 + 3 n) summm(n - 1)

         + summm(n) (3 n - 1) (3 n - 2) (2 a - b + 3 n) (2 a + b - 1 + 3 n)

> # The following two sums are identified to be equal, see Strehl (1993)

> sumrecursion(binomial(n,k)^3,k,n);

              2                    2                                  2
     - 8 (n - 1)  summm(n - 2) - (7 n  - 7 n + 2) summm(n - 1) + summm(n) n

> sumrecursion(binomial(n,k)^2*binomial(2*k,n),k,n);

              2                    2                                  2
     - 8 (n - 1)  summm(n - 2) - (7 n  - 7 n + 2) summm(n - 1) + summm(n) n

> simplify_combinatorial((binomial(n,k)-binomial(n-2,k))/
  (binomial(n-3,k)-binomial(n-6,k)));

               (n - 5) (n - 4) (n - 3) (n - 2) (- k + 2 n - 1)
   ----------------------------------------------------------------------------
       2                          2
   (3 n  - 24 n - 3 k n + 12 k + k  + 47) (n - 2 - k) (- k + n - 1) (- k + n)

> extended_gosper(binomial(k/2,n),k,2);
```

```
                    (1/2 k + 1) binomial(1/2 k, n)
                    ------------------------------
                                n + 1

> WZ(binomial(n,k)*binomial(k/2,n)*(-1)^k*(-2)^n,k,n,1,2);

                       (- k + n - 1) (- k + n)
                       -----------------------
                       (n - 1) (- k + 2 n - 2)

> extended_sumrecursion(binomial(n,k)*binomial(k/2,n)*(-1)^k*(-2)^n,k,n,1,2);

                        summ(n) - summ(n - 1)

> extended_sumrecursion(binomial(n,k)*binomial(k/2,n)*(-2)^n,k,n,1,2);

                        summ(n) - summ(n - 1)

> hyperrecursion({-n,n+2*a,a},{2/2*a,(2*a+1)/2},2/4,n);

              (n - 1) summ(n - 2) + (n + 2 a - 1) summ(n)

> hyperrecursion({-n,n+4*a,a},{4/2*a,(4*a+1)/2},4/4,n);

            - (n - 1) summ(n - 2) + (n + 4 a - 1) summ(n)
```

References

[1] Abramowitz, M., Stegun, I. A. (1964). *Handbook of Mathematical Functions.* Dover Publ., New York.

[2] Bailey, W. N. (1935). *Generalized hypergeometric series.* Cambridge University Press, England, reprinted 1964 by Stechert-Hafner Service Agency, New York–London.

[3] Gessel, I., Stanton, D. (1982). Strange evaluations of hypergeometric series. *Siam J. Math. Anal.* **13**, 295–308.

[4] Gosper Jr., R. W. (1978). Decision procedure for indefinite hypergeometric summation. *Proc. Natl. Acad. Sci. USA* **75**, 40–42.

[5] Graham, R. L., Knuth, D. E. and Patashnik, O. (1994). *Concrete Mathematics. A foundation for Computer Science.* Addison-Wesley, Reading, Massachussets, second edition.

[6] Koepf, W. (1994) Reduce package for the indefinite and definite summation. Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Technical Report TR 94-9, 1994.

[7] Koornwinder, T. H. (1993). On Zeilberger's algorithm and its q-analogue: a rigorous description. *J. of Comput. and Appl. Math.* **48**, 91–111.

[8] Paule, P., Schorn, M. (1994). A Mathematica version of Zeilberger's algorithm for proving binomial coefficient identities. *J. Symbolic Computation*, to appear.

[9] SIAM Review (1994). Problem 94–2, SIAM Review **36**.

[10] Strehl, V. (1993). Binomial sums and identities. *Maple Technical Newsletter* **10**, 37–49.

[11] Wilf, H. S., Zeilberger, D. (1990a). Rational functions certify combinatorial identities. *J. Amer. Math. Soc.* **3**, 147–158.

[12] Wilf, H. S., Zeilberger, D. (1990b). Towards computerized proofs of identities. *Bull. of the Amer. Math. Soc.* **23**, 77–83.

[13] Wilf, H. S. (1993). Identities and their computer proofs. "SPICE" Lecture Notes, 31. August–2. September 1993. Available as anonymous ftp file `pub/wilf/lecnotes.ps` on the server `ftp.cis.upenn.edu`.

[14] Zeilberger, D. (1990). A fast algorithm for proving terminating hypergeometric identities. *Discrete Math.* **80**, 207–211.

[15] Zeilberger, D. (1991). The method of creative telescoping. *J. Symbolic Computation* **11**, 195–204.