FINE-TUNING ZEILBERGER'S ALGORITHM

The Methods of Automatic Filtering and Creative Substituting

Axel Riese*
Research Institute for Symbolic Computation
Johannes Kepler University Linz
A–4040 Linz, Austria
Axel.Riese@risc.uni-linz.ac.at

Abstract It is shown how the performance of Zeilberger's algorithm and its q-version for proving (q-)hypergeometric summation identities can be dramatically improved by a frequently missed optimization on the programming level and by applying certain kinds of substitutions to the summand. These methods lead to computer proofs of identities for which all existing programs have failed so far.

Keywords: Zeilberger's algorithm, summation, hypergeometric series, computer algebra.

1. INTRODUCTION

With Zeilberger's [20, 21] algorithm — also known as the method of *creative telescoping* — the process of proving and finding definite hypergeometric summation identities has become a task that to a large amount can be executed by computers. In recent years several implementations have been developed mainly for the computer algebra systems Maple and Mathematica. Nevertheless we are still faced with the situation that all those packages rather quickly exceed the systems' memory-capacities if applied to intricate examples. The object of this paper is to present two methods, one purely on the programming level and one on the user level, for improving the performance of the algorithm in general and for certain types of applications, respectively.

*Supported by SFB-grant F1305 of the Austrian FWF.

The paper is organized as follows. In Section 2 we briefly outline the theoretical background of Zeilberger's algorithm and its q-version. In Section 3 we present the method of automatic filtering, a simple but efficient programming trick to speed-up the algorithm. In Section 4 we describe the method of creative substituting. In other words, we show how a clever substitution of parameters in the summand or a shifting of the summation interval can reduce the run-time of the algorithm substantially.

Several examples of computer proofs will illustrate the power of both techniques. To indicate the magnitude of the achieved speed-up we include the following table which summarizes the run-times for the proofs listed in this paper with applying none, one, or both of the optimizations. The timings refer to tests on an SGI Octane, "o.o.m." is used as an abbreviation for "out of memory".

Table 1. Timings

Id.	*none*	*A. Filt.*	*C. Subst.*	*both*
(5)	o.o.m.	1820 s	2 s	1 s
(6)	1506 s	414 s	14 s	2 s
(7)	o.o.m.	o.o.m.	o.o.m.	123 s
(8)	o.o.m.	o.o.m.	o.o.m.	11 s
(9)	o.o.m.	o.o.m.	516 s	475 s
(10)	213 s	27 s	38 s	6 s

2. ZEILBERGER'S ALGORITHM

Zeilberger's algorithm takes as input a terminating hypergeometric sum and computes a linear recurrence with polynomial coefficients that is satisfied by this sum. Additionally it delivers a rational function, the so-called *certificate*, which contains all information necessary to validate the result independently.

More precisely, let $f_{n,k}$ be a double-indexed sequence over some suitable domain $\mathbb{F}$ (for computability, usually the field of rational numbers extended by some transcendental indeterminates), where n ranges over the nonnegative integers and k over all integers. We call $f_{n,k}$ *hypergeometric* in both parameters if both quotients

$$\frac{f_{n+1,k}}{f_{n,k}} \quad \text{and} \quad \frac{f_{n,k+1}}{f_{n,k}}$$

are rational functions in n and k over $\mathbb{F}$ (disregarding singularities). For example, the sequence $f_{n,k} := \binom{n}{k}$ is hypergeometric in n and k.

It was shown by Wilf and Zeilberger [18] that any hypergeometric sequence fulfilling some extra conditions, i.e. any so-called *proper hypergeometric* sequence, satisfies a linear recurrence of the form

$$\sigma_0(n)\, f_{n,k} + \sigma_1(n)\, f_{n+1,k} + \cdots + \sigma_d(n)\, f_{n+d,k} = g_{n,k+1} - g_{n,k}, \tag{1}$$

where the σ_i are polynomials in n over $\mathbb{F}$ not depending on k, and $g_{n,k}$ is a rational function multiple of $f_{n,k}$ and therefore a hypergeometric sequence, too.

Now suppose that $f_{n,k}$ has *finite support*, i.e., for each nonnegative integer n there exists a finite integer interval I_n such that $f_{n,k}$ vanishes for $k \notin I_n$. Then $S_n := \sum_k f_{n,k}$, where k runs through all integers, actually denotes a finite sum, for which a recurrence can be easily deduced from (1), namely

$$\sigma_0(n)\, S_n + \sigma_1(n)\, S_{n+1} + \cdots + \sigma_d(n)\, S_{n+d} = 0. \tag{2}$$

Zeilberger [20] made the crucial observation that a slight variation of Gosper's [7] algorithm applied to

$$f_{n,k} \cdot \Big(\sigma_0(n) + \sigma_1(n)\, \frac{f_{n+1,k}}{f_{n,k}} + \cdots + \sigma_d(n)\, \frac{f_{n+d,k}}{f_{n,k}}\Big), \tag{3}$$

a rational function multiple of the original summand $f_{n,k}$ with undetermined σ_i, can be used to compute both the polynomials σ_i and the sequence $g_{n,k}$ from (1).

Several implementations of Zeilberger's algorithm have been carried out; the most prominent ones are due to Koepf [8], Koornwinder [9], Paule and Schorn [12], and Zeilberger (see Petkovšek, Wilf and Zeilberger [14]).

Since our methods will be shown to work also in the q-hypergeometric universe, we briefly comment on the underlying theory. A sequence $f_{n,k}$ with values in $\mathbb{F}(q)$ is called q-hypergeometric if the quotients

$$\frac{f_{n+1,k}}{f_{n,k}} \quad \text{and} \quad \frac{f_{n,k+1}}{f_{n,k}}$$

are rational functions in q^n and q^k over $\mathbb{F}(q)$. Recall the standard definition of the *q-shifted factorial*,

$$(a;q)_0 := 1, \quad \text{and} \quad (a;q)_k := (1-a)(1-aq)\cdots(1-aq^{k-1}) \text{ for } k > 0,$$

with the common abbreviation

$$(a_1, \ldots, a_m; q)_k := (a_1;q)_k \cdots (a_m;q)_k.$$

Then the sequence of *Gaussian polynomials* (or *q-binomial coefficients*)

$$\begin{bmatrix} n \\ k \end{bmatrix}_q := \begin{cases} \dfrac{(q;q)_n}{(q;q)_k\,(q;q)_{n-k}}, & \text{if } 0 \le k \le n, \\ 0, & \text{otherwise} \end{cases}$$

is q-hypergeometric in n and k.

Wilf and Zeilberger [18] first showed that Zeilberger's algorithm can be carried over to the q-case. It computes for a terminating q-hypergeometric sum a recurrence of form (2) with the only difference that the σ_i are polynomials in q^n over $\mathbb{F}(q)$. Implementations have been developed by Böing and Koepf [4], Koornwinder [9], the author (see Paule and Riese [11]), and Zeilberger (see Petkovšek, Wilf and Zeilberger [14]).

The efficiency of Zeilberger's algorithm suffers from the fact that the most expensive part of Gosper's algorithm consists in solving a homogeneous system of linear equations with coefficients being polynomials in several variables, which is known to be a rather time- and especially memory-consuming task. Furthermore, Zeilberger's algorithm does not always find a recurrence of minimal order. While Paule's [10] method of *creative symmetrizing* overcomes this problem in many instances and, as a side-effect, reduces the run-time of the algorithm for certain types of summands, we shall present different optimizations in the following.

3. THE METHOD OF AUTOMATIC FILTERING

The first improvement is based on an observation we made in the process of fine-tuning our implementation of the q-Zeilberger algorithm (see Paule and Riese [11]) and has been described originally in the author's PhD thesis [16, p. 89]. Although the idea is straightforward and simple, it has been obviously overlooked by all programmers. The method can be easily integrated into Zeilberger's algorithm and does not require any creativity from the user.

Suppose that the summand $f_{n,k}$ contains factors that do not depend on k. As an example think of the numerator factor $n!$ in the common definition of the binomial coefficient $\binom{n}{k} := n!/(k!\,(n-k)!)$ for $0 \le k \le n$. Moreover, such factors may be hidden as one can see for instance from the relation $(a+k)_{n-k} = (a)_n/(a)_k$, $0 \le k \le n$, where $(a)_k$ denotes the *rising factorial* given by

$$(a)_0 := 1, \quad \text{and} \quad (a)_k := a\,(a+1)\cdots(a+k-1) \text{ for } k > 0.$$

If we now set up the input to Gosper's algorithm as in (3), then also the quotients $f_{n+i,k}/f_{n,k}$ contain factors free of k. Consequently, (3) can be written as

$$f_{n,k} \cdot \big(\sigma_0(n) + \sigma_1(n)\,r_1(n)\,s_1(n,k) + \cdots + \sigma_d(n)\,r_d(n)\,s_d(n,k)\big), \qquad (4)$$

where the r_i are rational functions in n and the s_i are rational functions in both n and k. Of course we do not have to enter Gosper's algorithm also with the r_i, because if we find a solution corresponding to the "filtered" summand

$$f_{n,k} \cdot \big(\hat{\sigma}_0(n) + \hat{\sigma}_1(n)\, s_1(n,k) + \cdots + \hat{\sigma}_d(n)\, s_d(n,k)\big),$$

then the solution corresponding to the full summand (4) is given by

$$\sigma_0 = \hat{\sigma}_0,\ \sigma_1 = \hat{\sigma}_1/r_1,\ \ldots,\ \sigma_d = \hat{\sigma}_d/r_d.$$

To transform the new rational solution into polynomials we only have to multiply by the least common multiple of the numerators of the r_i, which is admissible because the equations are homogeneous.

To summarize, by filtering out all factors not depending on k the coefficients of the equation system in Gosper's algorithm become smaller and the run-time decreases significantly. In particular this holds true for all recurrences computed in the following section; see also the timings in Table 1.

4. THE METHOD OF CREATIVE SUBSTITUTING

Our second optimization utilizes the fact that the complexity of the quotients $f_{n+i,k}/f_{n,k}$ involved in (3),

$$f_{n,k} \cdot \Big(\sigma_0(n) + \sigma_1(n)\, \frac{f_{n+1,k}}{f_{n,k}} + \cdots + \sigma_d(n)\, \frac{f_{n+d,k}}{f_{n,k}}\Big),$$

sometimes can be reduced simply by the following two actions:

The Method of Creative Substituting.

- Find a clever substitution for free parameters in $f_{n,k}$ in order to reduce the dependence of $f_{n,k}$ on the recurrence variable n.
- Shift the finite summation interval, for instance, by substituting $k+n$, $k-n$, etc. for k in $f_{n,k}$.

Although these suggestions do not look very spectacular, the effect on the run-time again might be dramatic. With both techniques combined we are able to compute certain recurrences for which we ran out of memory before within a few minutes now.

Note that the task of creative substituting cannot be performed automatically, since it has to rely heavily — in contrast to automatic filtering — on the trained eye of the user.

4.1. PARAMETER SUBSTITUTION

To explain the applicability of creative substituting, we need the notion of *hypergeometric series* which we define as usual by

$$ {}_rF_s\left[\begin{matrix} a_1, a_2, \ldots, a_r \\ b_1, \ldots, b_s \end{matrix}; z\right] := \sum_{k=0}^{\infty} \frac{(a_1)_k\,(a_2)_k \cdots (a_r)_k}{k!\,(b_1)_k \cdots (b_s)_k}\, z^k. $$

As a first example let us consider a problem that has been communicated to us recently by J. Wimp [19], who being interested in closed form aspects asked for a recurrence representation w.r.t. n of

$$ {}_4F_3\left[\begin{matrix} n-j, n+j+2a, n+a, n+b+1/2 \\ 2n+b+1, n+2a, n+a+1/2 \end{matrix}; 1\right]. \tag{5} $$

It is known that such a closed form evaluation exists for $a = b$ or $b = 2a - 1/2$ by Saalschütz's theorem (see, for instance, Slater [17]).

In Wimp's case it turns out that the Paule/Schorn[1] Mathematica implementation of Zeilberger's algorithm does not find a first-order recurrence and needs half an hour to compute a recurrence of order 2. Note that without filtering we run out of memory immediately. This does not come as a complete surprise, because actually *seven* parameters in the ${}_4F_3$-summand depend on n, one of them (even worse!) on $2n$, which blows up the rational function in (3) enormously. On the other hand, if we denote the summand by $f_{n,k}(a, b, j)$ and apply the first idea of creative substituting stated above, we find that in $f_{n,k}(a - n, b - n, j + n)$ only *two* parameters depend on n. The corresponding recurrence can be found in one second:

```
In[1]:= <<zb.m

        Fast Zeilberger by Peter Paule and Markus Schorn. (V 2.52 beta)
        Systembreaker = ENullspace

In[2]:= (* the shifted factorial *)
        shfac[a_, b_] := (a+b-1)! / (a-1)!

In[3]:= Zb[shfac[n-j,k] shfac[n+j+2a,k] shfac[n+a,k] shfac[n+b+1/2,k] /
          (k! shfac[2n+b+1,k] shfac[n+2a,k] shfac[n+a+1/2,k]) /.
           {a->a-n,b->b-n,j->j+n}, {k,0,Infinity}, n, 2]

Out[3]=
```

[1] available at: *http://www.risc.uni-linz.ac.at/research/combinat/risc/software/PauleSchorn/*

$$
\begin{aligned}
&\{(-3+4\,a-2\,b-2\,n)\,(-4+2\,a-b-2\,n)\,(-1+a-n)\,(-1+2\,a+j-n)\\
&\quad(1+b+n)\,(2+b+n)\,(1+j+n)\,\mathrm{SUM}(n)+\\
&(-3+2\,a-b-2\,n)\,(-1+2\,a-n)\,(2+b+n)\,\big(-20+54\,a-46\,a^2+12\,a^3-25\,b+\\
&\quad 47\,a\,b-22\,a^2\,b-8\,b^2+8\,a\,b^2+16\,a\,j-24\,a^2\,j+8\,a^3\,j+14\,a\,b\,j-12\,a^2\,b\,j+\\
&\quad 4\,a\,b^2\,j+8\,j^2-12\,a\,j^2+4\,a^2\,j^2+7\,b\,j^2-6\,a\,b\,j^2+2\,b^2\,j^2-54\,n+108\,a\,n-\\
&\quad 60\,a^2\,n+8\,a^3\,n-51\,b\,n+64\,a\,b\,n-16\,a^2\,b\,n-11\,b^2\,n+6\,a\,b^2\,n+24\,a\,j\,n-\\
&\quad 16\,a^2\,j\,n+8\,a\,b\,j\,n+12\,j^2\,n-8\,a\,j^2\,n+4\,b\,j^2\,n-54\,n^2+72\,a\,n^2-20\,a^2\,n^2-\\
&\quad 35\,b\,n^2+22\,a\,b\,n^2-4\,b^2\,n^2+8\,a\,j\,n^2+4\,j^2\,n^2-24\,n^3+16\,a\,n^3-8\,b\,n^3-4\,n^4\big)\\
&\quad \mathrm{SUM}(1+n)-\\
&(-2+2\,a-b-2\,n)\,(-2+2\,a-n)\,(-1+2\,a-n)\,(-2+a-b-n)\\
&\quad(-2+2\,a-b+j-n)\,(2+b+j+n)\,(3+2\,n)\,\mathrm{SUM}(2+n)=0\}
\end{aligned}
$$

After doing the inverse substitutions for a, b and j, and applying Petkovšek's [13, 14] algorithm `Hyper`, we indeed see that this recurrence has no hypergeometric solution.

As a second example we investigate an identity due to Andrews [1, (4.2)] slightly rewritten as

$$H_0(n, n+1) = 0, \tag{6}$$

where

$$
\begin{aligned}
&H_0(n,m)\\
&\quad = {}_5F_4\left[\begin{matrix} -m-n, x+m+n+1, x-z+1/2, x+m, z+n+1 \\ (x+1)/2, x/2+1, 2z+m+n+1, 2x-2z+1 \end{matrix}; 1\right].
\end{aligned}
$$

While we do not intend to comment on the discussion about the role of computer proofs that was initiated by a lengthy and — due to a bug in the implementation — wrong automatic proof of this identity, it is worth noting that both opponents, Andrews and Zeilberger, developed different strategies for proving identities of this type (see Andrews [1, 2], and Ekhad and Zeilberger [6]).

First of all we observe that with the Paule/Schorn implementation and automatic filtering it takes about 7 minutes to compute a recurrence of order 2 for the sum, whereas without filtering we need 25 minutes. However, if we creatively substitute $x-n$ for x and $z-n$ for z, we reduce the number of n-dependent parameters in the ${}_5F_4$-series. In addition we make use of the rewriting rule

$$((x+1)/2)_k\,(x/2+1)_k = 2^{-2k}\,(x+1)_{2k},$$

since after substituting $x-n$ for x both factorial expressions on the left-hand side are no longer "Zeilberger admissible" (the coefficients of n must be integers). This problem does not occur on the right-hand side. Finally we obtain within 2 seconds:

```
In[4]:= H0[n_, m_] :=
          shfac[-m-n,k] shfac[x+m+n+1,k] shfac[x-z+1/2,k] *
          shfac[x+m,k] shfac[z+n+1,k] 4^k /
         (k! shfac[x+1,2k] shfac[2z+m+n+1,k] shfac[2x-2z+1,k])

In[5]:= Zb[H0[n,n+1] /. {x->x-n,z->z-n}, {k,0,Infinity}, n, 2]

Out[5]=
```

$$\{(1+n)\,(2+n)\,(3+2\,n)\,(1+n+x-2\,z)\,(2+n-x+2\,z)\,\mathrm{SUM}(n) - (2+n)\,(n-x)\,\big(42+57\,n+26\,n^2+4\,n^3+23\,x+19\,n\,x+4\,n^2\,x-3\,x^2 - 2\,n\,x^2-2\,x^3-12\,z-5\,n\,z+20\,x\,z+10\,n\,x\,z+8\,x^2\,z-24\,z^2 - 10\,n\,z^2-8\,x\,z^2\big)\,\mathrm{SUM}(1+n) + (n-x)\,(1+n-x)\,(3+n+x)\,(5+2\,n+2\,x-2\,z)\,(3+n+z)\,\mathrm{SUM}(2+n)=0\}$$

We want to emphasize that the positive effect of this substitution on the run-time has been already observed by Zeilberger[2]. Nevertheless, Zeilberger was not able to prove the other 19 identities of this type listed in Andrews' [1] paper with his implementation, and the problem of finding computer proofs for them remained open for almost 6 years. Only the combination of automatic filtering and creative substituting could finally close this gap, which means that now we are able to semi-automatically prove each of the identities within 3 minutes!

For instance, the proof of the next identity in Andrews' list [1, (4.3)],

$$H_0(n,n) = \frac{(x+n)\,(2z-x+2n)}{(x+2n)\,(2z-x+n)}\,P_n, \tag{7}$$

wherein

$$P_n = \frac{(1/2)_n\,(2z-x)_{2n}}{(x+1)_n\,(1+x-z)_n\,(z+n+1/2)_n}$$

reads as follows (note that we divide the summand by the closed form on the right-hand side of the identity and shorten the output which otherwise filled more than two pages):

```
In[6]:= P[n_] := shfac[1/2,n] shfac[2z-x,2n] /
                 (shfac[x+1,n] shfac[1+x-z,n] shfac[z+n+1/2,n])

In[7]:= Zb[H0[n,n] (x+2n) (2z-x+n) / ((x+n) (2z-x+2n) P[n]) /.
          {x->x-n,z->z-n}, {k,0,Infinity}, n, 3] // Short[#, 7]&
```

[2] see *http://www.math.temple.edu/~zeilberg/synd.html*

Out[7]//Short=

$$\{(1+n)\,(2+n)\,(1+n+x-z)\,(2+n+x-z)\,(1+2\,n+2\,z)\,(3+2\,n+2\,z)$$
$$\big(24+17\,n+3\,n^2+7\,x+8\,n\,x+2\,n^2\,x+9\,x^2+4\,n\,x^2+2\,x^3-26\,z-26\,n\,z-$$
$$6\,n^2\,z-16\,x\,z-8\,n\,x\,z-6\,x^2\,z-4\,z^2+4\,x\,z^2\big)\,\mathrm{SUM}(n)+\ll 2\gg-$$
$$(5+2\,n)\,(2+n+x)\,(3+n+x-2\,z)\,(5+2\,n+2\,x-2\,z)\,(3+n+z)$$
$$(3+n-x+2\,z)\,\big(10+11\,n+3\,n^2+x+4\,n\,x+2\,n^2\,x+5\,x^2+$$
$$4\,n\,x^2+2\,x^3-6\,z-14\,n\,z-6\,n^2\,z-8\,x\,z-8\,n\,x\,z-6\,x^2\,z-4\,z^2+4\,x\,z^2\big)$$
$$\mathrm{SUM}(3+n)=0\}$$

Now we prove that this recurrence is indeed satisfied by 1:

```
In[8]:= ExpandAll[% /. SUM[_] -> 1]
```

```
Out[8]= True
```

Checking that the sum evaluates to 1 for $n \in \{0,1,2\}$ completes the proof.

4.2. SHIFTING THE SUMMATION INTERVAL

Let us now turn to the second idea of creative substituting. For this we examine the innocent-looking summand $f_{n,k} = \binom{2n}{k}$. If we applied Zeilberger's algorithm directly with order 1, say, the quotient $f_{n+1,k}/f_{n,k}$ in (3) were found to be

$$\frac{f_{n+1,k}}{f_{n,k}} = \frac{(2n+2)\,(2n+1)}{(2n-k+2)\,(2n-k+1)} = \frac{(2n+2)\,(2n+1)}{(2n+2)\,(2n+1)-k(4n+3)+k^2}.$$

On the other hand, since

$$\sum_{k=0}^{2n}\binom{2n}{k} = \sum_{k=-n}^{n}\binom{2n}{n+k}$$

we could also run the algorithm on $\bar{f}_{n,k} := f_{n,n+k}$. In this case the quotient seen as a function in k reduces to the much simpler expression

$$\frac{\bar{f}_{n+1,k}}{\bar{f}_{n,k}} = \frac{(2n+2)\,(2n+1)}{(n+k+1)\,(n-k+1)} = \frac{(2n+2)\,(2n+1)}{(n+1)^2-k^2}.$$

It hardly needs to be pointed out that such a substitution improves the performance of the algorithm for more involved examples dramatically provided that other factors of the shifted summand $\bar{f}_{n,k}$ do not produce considerably more complicated quotients.

To illustrate the power of this method we look at an identity due to Carlitz [5],

$$\sum_{k=0}^{2n}(-1)^k \begin{bmatrix}2n\\k\end{bmatrix}_q (b,c,d,e;q)_k\,(b,c,d,e;q)_{2n-k}\,q^{k(6n-3k+1)/2}$$
$$= (-1)^n\,(b,c,d,e;q)_n\,(q^{n+1},q^n bc,q^n cd,q^n bd;q)_n\,e^n\,q^{n(3n-1)/2}, \quad (8)$$

where $bcde = q^{1-3n}$. In this case it emerges that no existing implementation of the q-Zeilberger algorithm is able to compute a recurrence of order 1 even with the help of creative symmetrizing *and* filtering.

However, shifting the summation interval via substituting $k+n$ for k solves the problem in less than 2 minutes. This happens because from the sequences

$$f^{(1)}_{n,k} := \begin{bmatrix}2n\\k\end{bmatrix}_q, \quad f^{(2)}_{n,k} := (a;q)_k\,(a;q)_{2n-k}, \quad \text{and} \quad f^{(3)}_{n,k} := q^{k(6n-3k+1)/2}$$

we obtain much more involved quotients $f^{(i)}_{n+1,k}/f^{(i)}_{n,k}$ than the quotients $\bar{f}^{(i)}_{n+1,k}/\bar{f}^{(i)}_{n,k}$, where $\bar{f}^{(i)}_{n,k} := f^{(i)}_{n,k+n}$ ($i \in \{1,2,3\}$). If we additionally apply the first idea of creative substituting, i.e., if we replace c by $c\,q^{-n}$ and d by $d\,q^{-n}$, the run-time decreases once more to 11 seconds.

For instance, using the author's[3] Mathematica implementation we get:

```
In[1]:= <<qZeil.m

Out[1]= Axel Riese's q-Zeilberger implementation version 1.9 loaded

In[2]:= b = q^(1-3n) / (c d e);

In[3]:= qZeil[(-1)^k qBinomial[2n,k,q] qfac[b,q,k] qfac[c,q,k] *
          qfac[d,q,k] qfac[e,q,k] qfac[b,q,2n-k] qfac[c,q,2n-k] *
          qfac[d,q,2n-k] qfac[e,q,2n-k] q^(k(6n-3k+1)/2) /
         ((-1)^n qfac[b,q,n] qfac[c,q,n] qfac[d,q,n] qfac[e,q,n] *
          qfac[q^(n+1),q,n] qfac[q^n b c,q,n] qfac[q^n c d,q,n] *
          qfac[q^n b d,q,n] e^n q^(n(3n-1)/2)) /.
          {k->k+n,c->c q^(-n),d->d q^(-n)}, {k,-n,n}, n, 1,
          MagicFactor->-k]

Out[3]= SUM(n) = 1
```

In a paper of Berkovich and McCoy [3] the sum

$$\sum_{k=-n}^{n} \begin{bmatrix}3n+k\\n-k\end{bmatrix}_q q^{k^2} y^{-k} \quad (9)$$

[3]available at *http://www.risc.uni-linz.ac.at/research/combinat/risc/software/qZeil/*

came to our attention. Again, we succeeded to compute a recurrence of order 4 *only* after shifting the summation interval via substituting $k-n$ for k within a few minutes. Otherwise we ran out of memory rather quickly.

In our final example we show that also reversing the order of summation might be of advantage sometimes. Let us consider a special case of an identity due to Rahman [15],

$$\sum_{k=0}^{n} \frac{(1-q^{3k-2n})\,(q^{-2n}, d, q^{1-2n}/d; q^2)_k\,(b, c, q^{1-2n}/bc; q)_k}{(q^{2-2n}/b, q^{2-2n}/c, bcq; q^2)_k\,(q, q^{1-2n}/d, d; q)_k}\, q^k = 0. \qquad (10)$$

If we reverse the order of summation, i.e., we replace k by $n-k$, the run-time decreases from 27 seconds to 6 seconds. Note that this happens for a different reason now. With this substitution it turns out that the degree of the solution polynomial in Gosper's algorithm, which is needed for computing the sequence $g_{n,k}$ in (1), drops from 2 to 1. Hence we end up with a smaller system of equations again.

```
In[4]:= Clear[b]

In[5]:= qZeil[(1-q^(3k-2n)) qfac[q^(-2n),q^2,k] qfac[d,q^2,k] *
           qfac[q^(1-2n)/d,q^2,k] qfac[b,q,k] qfac[c,q,k] *
           qfac[q^(1-2n)/(b c),q,k] q^k /
          (qfac[q^(2-2n)/b,q^2,k] qfac[q^(2-2n)/c,q^2,k] *
           qfac[b c q,q^2,k] qfac[q,q,k] qfac[q^(1-2n)/d,q,k] *
           qfac[d,q,k]) /. k->n-k, {k,0,n}, n, 1]

Out[5]=
```

$$\mathrm{SUM}(n) = \frac{(1-q^{2\,n})\,(1-b\,c\,q^{-2+2\,n})\,(1-b\,d\,q^{-2+2\,n})\,(1-c\,d\,q^{-2+2\,n})\,\mathrm{SUM}(-1+n)}{q^2\,(1-b\,q^{-2+2\,n})\,(1-c\,q^{-2+2\,n})\,(1-d\,q^{-2+2\,n})\,(1-b\,c\,d\,q^{-2+2\,n})}$$

Acknowledgments

I wish to thank Peter Paule and Burkhard Zimmermann for helpful comments.

References

[1] G.E. Andrews, Pfaff's method (I): The Mills-Robbins-Rumsey determinant, *Discrete Math.* **193** (1998), 43–60.

[2] G.E. Andrews, Pfaff's method (III): Comparison with the WZ method, *Electr. J. Combin.* **3** (1996), #R21.

[3] A. Berkovich and B.M. McCoy, The universal chiral partition function for exclusion statistics, in "Statistical Physics at the Eve of the 21st Century" (M.T. Batchelor and L.T. Wille, eds.), *Ser. Adv. Statist. Mech.* Vol. 14, pp. 240–256, 1999.

[4] H. Böing and W. Koepf, Algorithms for q-hypergeometric summation in computer algebra, *J. Symbolic Comput.* **28** (1999), 777–799.

[5] L. Carlitz, Some formulas of F.H. Jackson, *Monatsh. Math.* **73** (1969), 193–198.

[6] S.B. Ekhad and D. Zeilberger, Curing the Andrews syndrome, *J. Differ. Equations Appl.* **4** (1998), 299–310.

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

[8] W. Koepf, *Hypergeometric Summation*, Vieweg, 1998.

[9] T.H. Koornwinder, On Zeilberger's algorithm and its q-analogue, *J. Comput. Appl. Math.* **48** (1992), 91–111.

[10] P. Paule, Short and easy computer proofs of the Rogers-Ramanujan identities and of identities of similar type, *Electr. J. Combin.* **1** (1994), #R10.

[11] P. Paule and A. Riese, A Mathematica q-analogue of Zeilberger's algorithm based on an algebraically motivated approach to q-hypergeometric telescoping, in "Special Functions, q-Series and Related Topics" (M.E.H. Ismail et al., eds.), *Fields Inst. Commun.* Vol. 14, pp. 179–210, 1997.

[12] P. Paule and M. Schorn, A Mathematica version of Zeilberger's algorithm for proving binomial coefficient identities, *J. Symbolic Comput.* **20** (1995), 673–698.

[13] M. Petkovšek, Hypergeometric solutions of linear recurrences with polynomial coefficients, *J. Symbolic Comput.* **14** (1992), 243–264.

[14] M. Petkovšek, H.S. Wilf and D. Zeilberger, $A = B$, A.K. Peters, 1996.

[15] M. Rahman, Some quadratic and cubic summation formulas for basic hypergeometric series, *Canad. J. Math.* **45** (1993), 394–411.

[16] A. Riese, *Contributions to symbolic q-hypergeometric summation*, PhD thesis, RISC, J. Kepler University Linz, Austria, 1997.

[17] L.J. Slater, *Generalized Hypergeometric Functions*, Cambridge University Press, 1966.

[18] H.S. Wilf and D. Zeilberger, An algorithmic proof theory for hypergeometric (ordinary and "q") multisum/integral identities, *Invent. Math.* **108** (1992), 575–633.

[19] J. Wimp, Personal communication, January 2000.

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

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