An Implementation of Karr's Summation Algorithm in Mathematica*

Carsten.Schneider@risc.uni-linz.ac.at

Abstract

Implementations of the celebrated Gosper algorithm (1978) for indefinite summation are available on almost any computer algebra platform. We report here about an implementation of an algorithm by Karr, the most general indefinite summation algorithm known. Karr's algorithm is, in a sense, the summation counterpart of Risch's algorithm for indefinite integration. This is the first implementation of this algorithm in a major computer algebra system. Our version contains new extensions to handle also definite summation problems. In addition we provide a feature to find automatically appropriate difference field extensions in which a closed form for the summation problem exists. These new aspects are illustrated by a variety of examples.

1 Introduction

Karr developed an algorithm for **indefinite** summation [Kar81, Kar85] based on the theory of difference fields [Coh65]. He introduced so called $\pi\sigma$-fields, in which first order linear difference equations can be solved in full generality. We implemented this algorithm in the computer algebra system Mathematica and developed a user interface that dispenses the user from working explicitly with difference fields. Instead, the user can handle all summation problems in terms of sums and products.

This algorithm cannot only deal with series of hypergeometric terms, like Gosper's algorithm [Gos78, PS95], series with q-hypergeometric terms, like [PR97], or holonomic series, like Chyzak's algorithm [CS98], but with series of terms where for example the harmonic numbers can appear in the denominator (see section 2.4).

In some cases appropriate difference field extensions are necessary in order to find a closed form to a summation problem. In many cases our implementation is able to find such extensions automatically. Therefore one does not have to deal with problems concerning difference field extensions. This feature to find automatic extensions will be demonstrated in section 3.

Finally, we extended Karr's algorithm to handle **definite** summation problems. The algorithm is generalized so that linear difference equations of any order can be solved. It is also possible to consider field extensions in form of algebraic relations, like $((-1)^k)^2 = 1$. A rather complex example will illustrate how definite summation problems can be solved in our Mathematica implementation.

*This work is supported by the SFB grant F1305 of the Austrian "Fonds zur Förderung der wissenschaftlichen Forschung" and by the DAAD grant "Doktorandenstipendium im Rahmen des gemeinsamen Hochschulsonderprogramms III von Bund und Ländern".

This article is based on transparencies [Sch99] we used for a presentation at the *Séminaire Lotharingien de Combinatoire 43*. Some further results have been added.

The Mathematica package is available in an encoded form by email request to `Carsten.Schneider@risc.uni-linz.ac.at`. The Mathematica code of the package has not been published yet because it is still under construction.

2 Karr's indefinite summation algorithm

2.1 A first example

In the book Concrete Mathematics [GKP94, exercise 6.69] the task of finding a closed form representation of

$$\sum_{k=1}^{n} k^2 \mathrm{H}_{n+k},$$

where $\mathrm{H}_n := \sum_{k=1}^{n} \frac{1}{k}$ is the n-th harmonic number, is posed as a bonus problem. Knuth's solution to this problem is

$$\frac{1}{3} n \left(n + \frac{1}{2}\right)(n+1)\left(2\mathrm{H}_{2n} - \mathrm{H}_n\right) - \frac{1}{36} n \left(10n^2 + 9n - 1\right)$$

where he remarks

> **"It would be nice to automate the derivation of formulas such as this."**

The closed form of this bonus problem can be computed by using Karr's algorithm. The implementation is available in form of a Mathematica package, in which functions are provided to define a given summation problem in the Mathematica environment.

$\mathit{In}[1] := \mathbf{Problem69} = \mathbf{DefineSum}[\mathbf{k\hat{\ }2DefineHNumber}[\mathbf{n}+\mathbf{k}], \{\mathbf{k}, \mathbf{1}, \mathbf{n}\}]$

$$\mathit{Out}[1] = \sum_{\mathtt{k}=1}^{\mathtt{n}} \left(\mathtt{k}^2\ \mathrm{H}_{\mathtt{k}+\mathtt{n}}\right)$$

The functions `DefineSum` and `DefineProduct` are used to define sums and products. There are several other functions available, like `DefineHNumber`, `DefineBinomial` or `DefinePower` to define harmonic numbers, binomials or powers. Additionally, various functions are provided to introduce new objects.

Karr's algorithm is applied to the summation problem by calling the function `KReduce`. Here the solution of Karr's algorithm is simplified by using the Mathematica function `Simplify`.

$\mathit{In}[2] := \mathbf{KReduce}[\mathbf{Problem69}]//\mathbf{Simplify}$

$$\mathit{Out}[2] = -\frac{1}{36}\ \mathtt{n}\ (1+\mathtt{n})\ (-1 + 10\ \mathtt{n} + 6\ (1 + 2\ \mathtt{n})\ \mathrm{H}_{\mathtt{n}} - 12\ (1 + 2\ \mathtt{n})\ \mathrm{H}_{2\ \mathtt{n}})$$

2.2 Indefinite summation and first order linear difference equations

In this section we will give a rough outline of Karr's approach, which is based on the theory of difference fields. In the following a difference field is considered as a field together with a field automorphism[1].

A huge class of indefinite summation problems can be formalized by first order linear difference equations in difference field settings. Since Karr's algorithm can solve first order linear difference equations in full generality, Karr's algorithm enables to treat this type.

We will illustrate Karr's approach by the following elementary problem: find a closed form of

$$\sum_{k=0}^{n} k\,k!.$$

A difference field for the problem

Let t_1, t_2 be indeterminates and consider the field automorphism $\sigma : \mathbb{Q}(t_1, t_2) \to \mathbb{Q}(t_1, t_2)$ induced by

$$\begin{aligned} \sigma(t_1) &= t_1 + 1 \\ \sigma(t_2) &= (t_1 + 1)\,t_2. \end{aligned}$$

Note that the automorphism acts on t_1 and t_2 like the shift operator N on n and $n!$ via $Nn = n + 1$ and $Nn! = (n + 1)\,n!$.

A first order difference equation

The indefinite summation problem can be rephrased in terms of the difference field $\mathbb{Q}(t_1, t_2)$ as follows: find a solution $g \in \mathbb{Q}(t_1, t_2)$ of

$$\sigma(g) - g = t_1\,t_2.$$

Karr's algorithm computes the solution $g = t_2$ from which

$$(k + 1)! - k! = k\,k!$$

immediately follows.

The closed form

By the telescoping trick one obtains the closed form evaluation

$$\sum_{k=0}^{n} k\,k! = (n + 1)! - 1.$$

2.3 Difference field extensions

The goal is to simplify the triple sum (note $\mathrm{H}_i = \sum_{j=1}^{i} \frac{1}{j}$)

$$\sum_{k=1}^{n} \sum_{i=2}^{k} \frac{\mathrm{H}_i}{i^2 - 1},$$

[1] More precisely, we will consider only a subclass of difference fields, so called $\pi\sigma$-fields (see [Kar81]).

by eliminating the outermost sum. Applying Karr's algorithm on

$In[3] :=$ **tripleSum = DefineSum[**
DefineSum[DefineHNumber[i]/(i^2 − 1), {i, 2, k}],
{k, 1, n}]

$$Out[3] = \sum_{k=1}^{n}\Big(\sum_{i=2}^{k}\Big(\frac{H_i}{-1+i^2}\Big)\Big)$$

one gets

$In[4] :=$ **KReduce[tripleSum]//Simplify**

$$Out[4] = \frac{1}{4n(1+n)}\Big(2H_n - 2n(1+n)H_n^2 + (1+n)\Big(2 - n + 4n(1+n)\sum_{\iota_1=2}^{n}\Big(\frac{H_{\iota_1}}{-1+\iota_1^2}\Big)\Big)\Big)$$

This means, the triple sum can be expressed in terms of n, H_n and the double sum

$$\sum_{i=2}^{n}\frac{H_i}{i^2-1}. \tag{1}$$

Experiences with handling summations of harmonic numbers tell us that the sum (1) can be expressed by using the harmonic numbers of second order $H_n^{(2)} = \sum_{i=1}^{n}\frac{1}{i^2}$. Karr's algorithm can be forced to use $H_n^{(2)}$, which amounts algebraically to an extension of the underlying difference field, the solution space, by these elements $H_n^{(2)}$.

$In[5] :=$ **KReduce[tripleSum, Tower− > {$\mathbf{H_n^{(2)}}$}]//Simplify**

$$Out[5] = \frac{1}{4(1+n)}\Big(-2(3+2n)H_n - 2(1+n)H_n^{(2)} + (1+n)\big(3n + 2(1+n)H_n^{(2)}\big)\Big)$$

This non-automatic extension was carried out by setting the option `Tower−` > $\{H_n^{(2)}\}$.

2.4 The power of Karr's algorithm

Since Karr's algorithm is based on difference fields, arbitrary rational compositions of sums and products can be treated. As another example we consider

$In[6] :=$ **powerExample = DefineSum[1/DefineSum[1/i, {i, 1, k}]/**
(1 − k ∗ DefineSum[1/i, {i, 1, k}]),
{k, 2, n}]

$$Out[6] = \sum_{k=2}^{n}\Big(\frac{1}{\big(\sum_{i=1}^{k}\big(\frac{1}{i}\big)\big)\big(1 - k\sum_{i=1}^{k}\big(\frac{1}{i}\big)\big)}\Big)$$

Applying Karr's algorithm leads to the simplified expression

$In[7] :=$ **KReduce[powerExample]**

$$Out[7] = -1 + \frac{1}{\sum_{\iota_1=1}^{n}\big(\frac{1}{\iota_1}\big)}$$

3 Automatic difference field extensions

In section 2.3 it was shown, how a simpler closed form can be found by extending the difference field in which one expects to find the solution for the telescoping problem.

In general, one has to know in advance in which difference field extension a simple closed form exists. To make work easier, the implementation provides the possibility that appropriate extensions are searched for automatically.

The following two examples show how the automatic extension feature can be applied to summation problems.

Finding a manual extension

Applying Karr's algorithm on the double sum (1)

$In[8] := \mathbf{doubleSum} =$
$\qquad \mathbf{DefineSum[DefineHNumber[k]/(k\hat{\ }2-1), \{k, 2, n\}]}$

$$Out[8] = \sum_{\mathtt{k}=2}^{\mathtt{n}} \left(\frac{\mathtt{H_k}}{-1+\mathtt{k}^2} \right)$$

the following result is found with the automatic extension feature:

$In[9] := \mathbf{KReduce[doubleSum, TowerSuggestion- > True]//Simplify}$

$$Out[9] = \frac{-2\left(1+4\mathtt{n}+2\mathtt{n}^2\right)\mathtt{H_n}+\mathtt{n}(1+\mathtt{n})\left(7+4\sum_{\iota_1=2}^{\mathtt{n}}\left(\frac{-1+2\iota_1^2}{2(-1+\iota_1)\iota_1^2}\right)\right)}{4\mathtt{n}(1+\mathtt{n})}$$

Looking closer at the suggested extension

$$\sum_{i=2}^{n} \frac{2i-1}{2(i-1)i^2} = \frac{1}{2}\left(\sum_{i=2}^{n}\frac{1}{i-1} + \sum_{i=2}^{n}\frac{1}{i^2} - \sum_{i=2}^{n}\frac{1}{i}\right)$$

from the partial fraction decomposition one sees immediately that this sum can be expressed by the harmonic numbers of first and second order. This observation justifies the use of the manual extension $\mathrm{H}_n^{(2)}$, as it was done in section 2.3.

Automatic extension in two steps

The following expression consisting of four nested sums will be simplified by using Karr's algorithm twice with the automatic extension feature.

$In[10] := \mathbf{quadrupleSum = DefineSum[DefineSum[DefineSum[}$
$\qquad \mathbf{1/DefineHNumber[i], \{i, 1, k\}], \{k, 1, n\}],}$
$\qquad \mathbf{\{n, 2, m\}]}$

$$Out[10] = \sum_{\mathtt{n}=2}^{\mathtt{m}}\left(\sum_{\mathtt{k}=1}^{\mathtt{n}}\left(\sum_{\mathtt{i}=1}^{\mathtt{k}}\left(\frac{1}{\mathtt{H_i}}\right)\right)\right)$$

In the first step an expression is found consisting of terms with at most three nested sums.

$In[11] :=$ **tripleSum** $=$
KReduce[**quadrupleSum**, **TowerSuggestion**$->$ **True**]//
Simplify

$$Out[11] = (1+\mathtt{m})\sum_{\iota_1=1}^{\mathtt{m}}\left(\frac{1}{\mathtt{H}_{\iota_1}}\right) + (1+\mathtt{m})\sum_{\iota_1=1}^{\mathtt{m}}\left(-\frac{\iota_1}{\mathtt{H}_{\iota_1}}\right) + \sum_{\iota_1=2}^{\mathtt{m}}\left(\frac{\iota_1\left(-1+\iota_1+\mathtt{H}_{\iota_1}\sum_{\iota_2=1}^{\iota_1}\left(\frac{1}{\mathtt{H}_{\iota_2}}\right)\right)}{\mathtt{H}_{\iota_1}}\right)$$

In the second step Karr's algorithm returns a term with at most two nested sums.

$In[12] :=$ **KReduce**[**tripleSum**,
m, **TowerSuggestion**$->$ **True**, **Level**$->$ **2**]//
Simplify

$$Out[12] = \frac{1}{2\mathtt{H}_{\mathtt{m}}}\Bigg(2(-1+\mathtt{m})\mathtt{m} + \mathtt{H}_{\mathtt{m}}\Bigg(-2+(1+\mathtt{m})(2+\mathtt{m})\sum_{\iota_1=1}^{\mathtt{m}}\left(\frac{1}{\mathtt{H}_{\iota_1}}\right) + 2(1+\mathtt{m})\sum_{\iota_1=1}^{\mathtt{m}}\left(-\frac{\iota_1}{\mathtt{H}_{\iota_1}}\right) - \sum_{\iota_1=2}^{\mathtt{m}}\left(-\frac{(1+\mathtt{H}_{\iota_1}(-4+\iota_1))(-1+\iota_1)\iota_1}{\mathtt{H}_{\iota_1}(-1+\mathtt{H}_{\iota_1}\iota_1)}\right)\Bigg)\Bigg)$$

The higher the value of option `Level` is set, the more the chances are increased to find an appropriate extension. But this also increases the required time and space resources.

4 Definite summation

I have extended Karr's algorithm in order to deal also with definite summation. I will demonstrate some features of my Mathematica package with a concrete example. In [FK99] the following identity pops up:

$$\sum_{k=1}^{n}\frac{\mathrm{H}_k\ (3+k+n)!\ (-1)^k\ (-1)^{-1+n}}{(1+k)!\ (2+k)!\ (-k+n)!} + \frac{(n)!}{(3+n)!}\sum_{k=1}^{n}-\frac{(3+k+n)!\ (-1)^k\ (1-(2+n)\ (-1)^n)}{k\ (1+k)!^2\ (-k+n)!} = (2+n)(-1)^n-2. \quad (2)$$

With my package one not only can prove this identity automatically but even is able to *find* the closed form

$$(2+n)(-1)^n-2.$$

In the following the two sums on the left hand side of (2) are considered separately.

In[13] := **mySum1 = DefineSum[DefinePower[−1, k] DefinePower[−1, n − 1]**
DefineFactorial[n + k + 3]/(DefineFactorial[k + 1]
DefineFactorial[k + 2]DefineFactorial[n − k])
DefineHNumber[k], {k, 1, n}]

$$Out[13] = \sum_{k=1}^{n} \Big(\frac{H_k\ (3+k+n)!_{.}\ (-1)^{k_{.}}\ (-1)^{-1+n_{.}}}{(1+k)!_{.}\ (2+k)!_{.}\ (-k+n)!_{.}} \Big)$$

In[14] := **mySum2 = DefineSum[−DefinePower[−1, k]**
DefineFactorial[n + k + 3](1 − DefinePower[−1, n](n + 2))/
(k DefineFactorial[k + 1]ˆ2DefineFactorial[n − k]),
{k, 1, n}]

$$Out[14] = \sum_{k=1}^{n} \Big(-\frac{(3+k+n)!_{.}\ (-1)^{k_{.}}\ (1-(2+n)\ (-1)^{n_{.}})}{k\ (1+k)!_{.}^{\ 2}\ (-k+n)!_{.}} \Big)$$

Especially, in section 4.1 I will demonstrate the basic procedure to solve definite summation problems with my Mathematica package, whereas in section 4.2 I focus on some technical details one has to take into account to find closed forms.

4.1 A closed form of `mySum1`

Finding a recurrence

First a recurrence is found that is satisfied by `mySum1`.

In[15] := **rec1 = GenerateRecurrence[mySum1]//Simplify**

$$\begin{aligned} Out[15] = \big\{&n\ (1+n)\ (2+n)\ (3+n)\ (4+n)\ (-1+n)! \\ &\big(-(9+2\ n)\ \big(8+6\ n+n^2\big)\ \mathtt{SUM}[n]+ \\ &\quad (9+2\ n)\ \big(13+8\ n+n^2\big)\ \mathtt{SUM}[1+n]+ \\ &\quad \big(30+42\ n+17\ n^2+2\ n^3\big)\ \mathtt{SUM}[2+n]- \\ &\quad (3+n)\ \big(25+15\ n+2\ n^2\big)\ \mathtt{SUM}[3+n]\big) == \\ &2\ (-1)^n\ (9+2\ n)\ \big(35+24\ n+4\ n^2\big)\ (4+n)!\big\} \end{aligned}$$

The idea how to find a recurrence is based on Zeilberger's creative telescoping method [Zei90]. Although Karr's original summation algorithm was already capable to carry out creative telescoping, nobody has noticed this possibility until now.

Solving the recurrence

In the second step, solutions of the recurrence are computed. As `mySum1` depends on the harmonic numbers, it can be expected that the solutions for its recurrence also consist of terms of the harmonic numbers. Consequently the solution space is extended manually via `Tower−` $> \{H_n\}$.

In[16] := **recSol1 = SolveRecurrence[rec1, SUM[n], Tower− > {H$_\mathbf{n}$}]**

$$Out[16] = \left\{\{0,1\}, \left\{0, \frac{3 - n^2 + 4\ H_n + 6\ n\ H_n + 2\ n^2\ H_n}{(1+n)\ (2+n)}\right\}, \left\{0, \frac{1}{4}\ (2+n)\ (-1)^{n}\right\}, \left\{1, \frac{\left(16 - 13\ n^2 - 5\ n^3 + 32\ H_n + 64\ n\ H_n + 40\ n^2\ H_n + 8\ n^3\ H_n\right)\ (-1)^{n}}{4\ (1+n)\ (2+n)}\right\}\right\}$$

This has to be interpreted as follows: Karr's algorithm delivers three linear independent solutions for the homogeneous version of the recurrence, namely

$$1, \qquad \frac{3 - n^2 + 4\ \mathrm{H}_n + 6\ n\ \mathrm{H}_n + 2\ n^2\ \mathrm{H}_n}{(1+n)\ (2+n)}, \qquad \frac{1}{4}\ (2+n)\ (-1)^{n}$$

and one particular solution of the inhomogeneous recurrence itself:

$$\frac{\left(16 - 13\ n^2 - 5\ n^3 + 32\ \mathrm{H}_n + 64\ n\ \mathrm{H}_n + 40\ n^2\ \mathrm{H}_n + 8\ n^3\ \mathrm{H}_n\right)\ (-1)^{n}}{4\ (1+n)\ (2+n)}.$$

Finding the linear combination

Finally, the closed form of `mySum1` is that linear combination of the homogeneous solutions plus the inhomogeneous solution which has exactly the same initial values as `mySum1`. This is also computed automatically:

In[17] := **solution1 = FindLinearCombination[recSol1, mySum1]**

$$Out[17] = -1 - \frac{3 - n^2 + 4\ H_n + 6\ n\ H_n + 2\ n^2\ H_n}{(1+n)\ (2+n)} + \frac{1}{4}\ (2+n)\ (-1)^{n} + \frac{\left(16 - 13\ n^2 - 5\ n^3 + 32\ H_n + 64\ n\ H_n + 40\ n^2\ H_n + 8\ n^3\ H_n\right)\ (-1)^{n}}{4\ (1+n)\ (2+n)}$$

4.2 A Closed form of `mySum2`

Finding a recurrence

Similar to `mySum1` a recurrence of order 2 for the second sum is computed.

In[18] := **rec2 = GenerateRecurrence[mySum2, RecOrder− > 2]**

$$\begin{aligned} Out[18] = \{ & -n\ (1+n)\ (3+n)\ (1 + 3\ (-1)^n + (-1)^n\ n) \\ & \quad (-1 + 4\ (-1)^n + (-1)^n\ n)\ \left(28 + 15\ n + 2\ n^2\right)\ (-1+n)!\ \mathrm{SUM}[n] + \\ & 6\ n\ (1+n)\ (3+n)^2\ (-1 + 2\ (-1)^n + (-1)^n\ n) \\ & \quad (-1 + 4\ (-1)^n + (-1)^n\ n)\ (-1+n)!\ \mathrm{SUM}[1+n] + \\ & n\ (1+n)\ (3+n)\ (-1 + 2\ (-1)^n + (-1)^n\ n) \\ & (1 + 3\ (-1)^n + (-1)^n\ n)\ \left(10 + 9\ n + 2\ n^2\right)\ (-1+n)!\ \mathrm{SUM}[2+n] == \\ & -2\ (-1 + 2\ (-1)^n + (-1)^n\ n)\ (1 + 3\ (-1)^n + (-1)^n\ n) \\ & (-1 + 4\ (-1)^n + (-1)^n\ n)\ \left(35 + 24\ n + 4\ n^2\right)\ (4+n)! \} \end{aligned}$$

Here the order of the recurrence we were looking for is specified by the option `RecOrder − > 2`. By default - as in the previous example for `mySum1` - the algorithm starts looking for a recurrence of order one and increases the order step by step if it does not succeed in finding a recurrence of the current order.

Solving the recurrence

In the second step, the following solutions for the recurrence are found.

$$In[19] := \mathbf{recSol2} = \mathbf{SolveRecurrence[rec2, SUM[n], Tower- > \{DefineHNumber[n]\}, plusBound- > 1, WithMinusPower- > True]}$$

$$Out[19] = \{\{0, 2 + n - (-1)^{n.}\}, \{0, 16 - 6\ n^2 - n^3 + (-1)^{n.} + 28\ n\ (-1)^{n.} + 23\ n^2\ (-1)^{n.} + 8\ n^3\ (-1)^{n.} + n^4\ (-1)^{n.}\}, \{1, \frac{1}{28}\ (260 - 150\ n^2 - 39\ n^3 + 336\ H_n + 616\ n\ H_n + 336\ n^2\ H_n + 56\ n^3\ H_n - 325\ (-1)^{n.} + 365\ n^2\ (-1)^{n.} + 228\ n^3\ (-1)^{n.} + 39\ n^4\ (-1)^{n.} - 672\ H_n\ (-1)^{n.} - 1568\ n\ H_n\ (-1)^{n.} - 1288\ n^2\ H_n\ (-1)^{n.} - 448\ n^3\ H_n\ (-1)^{n.} - 56\ n^4\ H_n\ (-1)^{n.})\}\}$$

To handle this problem, I have generalized Karr's algorithm for solving linear difference equations of any order. For this generalization a denominator bounding is used which was developed by Bronstein [Bro99]. Unfortunately, there is still an unsolved problem concerning degree boundings of some solution parts. Nevertheless one can find all possible solutions by an incremental strategy, i.e., increasing step by step the degree boundings for each computation attempt. By increasing the value of the `plusBound` option these boundings are raised. Consequently the chances are higher to find more solutions. For this strategy, however, more time and space resources are required. In this example the value of `plusBound` is set at least as high as 1. By default - as in the previous example for `mySum1` - the value of `plusBound` is set to 1.

In addition, we have to consider the algebraic relation

$$((-1)^k)^2 = 1$$

to find all solutions for the recurrence. In order to take care of this, the option `WithMinusPower` is set to `True`.

Finding the linear combination of `mySum2`

Finally the closed form of `mySum2` can be found as before:

$$In[20] := \mathbf{solution2 = FindLinearCombination[recSol2, mySum2]}$$

$$Out[20] = -(3 + n)\ (-1 + 3\ n + 2\ n^2 - (-1 + 6\ n + 7\ n^2 + 2\ n^3)\ (-1)^{n.} + 2\ (2 + 3\ n + n^2)\ H_n\ (-1 + (2 + n)\ (-1)^{n.}))$$

4.3 The closed form of `mySum1`+`mySum2`

In the end, by combining the closed forms of `mySum1` and `mySum2` the closed form of the original summation problem (2) is computed.

$$In[21] := \mathbf{solution1 + solution2/((n + 1)(n + 2)(n + 3))//Simplify}$$

$$Out[21] = -2 + (2 + n)\ (-1)^{n.}$$

References

[Bro99] M. Bronstein. On solutions of linear ordinary difference equations in their coefficient field. Technical Report 3797, INRIA Sophia Antipolis, November 1999.

[Coh65] R. M. Cohn. *Difference Algebra*. Interscience Publishers, John Wiley & Sons, 1965.

[CS98] F. Chyzak and B. Salvy. Non-commutative elimination in ore algebras proves multivariate identities. *J. Symbolic Computation*, 26(2):187–227, 1998.

[FK99] M. Fulmek and C. Krattenthaler. The number of rhombus tilings of a symmetric hexagon which contains a fixed rhombus on the symmetric axis, II. *European Journal of Combinatorics*, 1999. to appear.

[GKP94] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*, chapter 6. Special Numbers, exercise 69, page 316. Addison Wesley Publishing Company, 2nd edition, 1994.

[Gos78] R. W. Gosper. Decision procedures for indefinite hypergeometric summation. *Proc. Nat. Acad. Sci. U.S.A.*, 75:40–42, 1978.

[Kar81] M. Karr. Summation in finite terms. *Journal of the ACM*, 28:305–350, 1981.

[Kar85] M. Karr. Theory of summation in finite terms. *J. Symbolic Computation*, 1:303–315, 1985.

[PR97] P. Paule and A. Riese. A Mathematica q-analogue of Zeilberger's algorithm based on an algebraically motivated aproach to q-hypergeometric telescoping. In M. Ismail and M. Rahman, editors, *Special Functions, q-Series and Related Topics*, volume 14, pages 179–210. Fields Institute Toronto, AMS, 1997.

[PS95] P. Paule and M. Schorn. A Mathematica version of Zeilberger's algorithm for proving binomial coefficient identities. *J. Symbolic Computation*, 20:673–698, 1995.

[Sch99] C. Schneider. An implementation of Karr's summation algorithm in Mathematica. Technical Report SFB 99-15, Johannes Kepler University Linz, August 1999.

[Zei90] D. Zeilberger. A fast algorithm for proving terminating hypergeometric identities. *Discr. Math.*, 80:207–211, 1990.