A Macsyma[*]Implementation of Zeilberger's Fast Algorithm[†]

Fabrizio Caruso

November 15, 2000

Abstract

We present the first implementation within the Macsyma computer algebra system of Zeilberger's fast algorithm for the definite summation problem for a very large class of sequences; i.e. given a hypergeometric sequence $F(n,k)$, we want to represent $f(n)=\sum_{k=0}^{n} F(n,k)$ in a "simpler" form. We do this by finding a linear recurrence for the summand $F(n,k)$, from which we can obtain a homogeneous $k-$free recurrence for $f(n)$. The solution of this recurrence is left as a post-processing, and it will give the "simpler" form we were looking for.

Zeilberger's fast algorithm exploits a specialized version of Gosper's algorithm for the indefinite summation problem; i.e. given a hypergeometric sequence $t(k)$, the problem of finding another sequence $T(k)$ such that $t(k)=\Delta_k T(k)=T(k+1)-T(k)$. The implementation of this algorithm has also been carried out in Macsyma, and its details are also briefly described in this paper.

1 Introduction

We present the first implementation within the Macsyma computer algebra system of Zeilberger's fast algorithm [13], [14] for the definite summation problem for the large class of proper hypergeometric sequences [12]. This means that given a double-indexed sequence $F(n,k)$, we want to rewrite the definite sum $f(n)=\sum_{k=0}^{n} F(n,k)$ in a form free of quantifier $\sum$. We do this by finding a special $k-$free linear recurrence with polynomial coefficients for the summand $F(n,k)$, which, under the condition of natural boundary for $F(n,k)$, can be extended into a $k-$free homogeneous linear recurrence with polynomial coefficients. The desired linear recurrence for the summand has the form: $\sum_{i=0}^{m} a_i(n)F(n+$

[*]Comp. algebra system dev. by Macsyma Inc., version 419

[†]Supported by the SFB grant F1305 of the Austrian FWF

$i, k) = G(n, k+1) - G(n, k)$, where the $a_i(n)$ are polynomials in n free of k. The search for this recurrence is done by guessing the order m of the left hand side of this recurrence and then trying to find the corresponding $G(n, k)$ of the right hand side by means of a specialized Gosper algorithm [5] (We note that a priori upper bounds for m can be given; however they turn out to be too large and therefore not useful from the practical point of view).

Gosper's algorithm for the indefinite summation problem solves the hypergeometric telescoping problem, i.e. given a hypergeometric sequence $t(k)$, it decides whether there is a hypergeometric sequence $T(k)$ such that $t(k) = \Delta_k T(k) = T(k+1) - T(k)$, and if it exists, it finds it by rewriting this problem as a linear recurrence in which the unknown is a polynomial which determines $T(k)$. For a simple and thorough description of these algorithms see [6], [10]. For alternative approaches and a detailed explanation of the connection between these strategies see, for instance, chapter 23 in [4] or [7], [1].

However the standard Gosper's algorithm, which is already implemented in Macsyma, cannot be used in Zeilberger's algorithm because it does not take into account that there are some polynomial parameters. Therefore a specialized parameterized version of this algorithm, in which unknown parameters a_i's are taken into account, has been implemented and it is used by our implementation of Zeilberger's algorithm to compute the right hand side of the desired recurrence.

This package [2] can be downloaded from the home page `http://www.risc.uni-linz.ac.at/research/combinat/risc/` and will also be soon available on the macsyma home page `http://www.macsyma.com`.

At the same home page a Mathematica implementation of this algorithm, which considers the more general q-case [9], as well as the Paule-Schorn implementation [8] for the ordinary case, can be found.

2 The Implementation

In this section we report on some of the details of the implementation.

The implementation of this algorithm has been done in the internal LISP-like language of the Macsyma computer algebra system.

It has been implemented in Macsyma in a straightforward way and no significant changes have been made in the classical algorithms.

This version of Zeilberger's algorithm works in the proper hypergeometric case. We remark that there are implementations that could work at least in principle in a slightly larger class of sequences (holonomic hypergeometric sequences, see [3]). Our choice makes the implementation simpler and allows us to use our implementation of Gosper's algorithm for the proper hypergeometric case.

2.0.1 Timings

We present some timings obtained by testing the function `parGosper` on increasing powers of the binomial coefficients in which no loop on the order of the recurrence is run (we know it in advance). In these tests the Paule-Schorn Mathematica implementation [8] and our Macsyma implementation are compared. These implementations have been run on an SGI Octane with 2 gigabytes of ram and two 250 Mhz RISC processors (although much less memory was necessary for running the program on these examples).

Results *Note:Timings are in seconds*

power	**order**	*Paule/Schorn Mathematica*	*Macsyma*	*diff. ratio*
3	2	$0.36s$	$3.31s$	9.19
4	2	$0.92s$	$3.95s$	4.29
5	3	$4.47s$	$37.11s$	8.30
6	3	$16.98s$	$121.64s$	7.16
Average ratio :				7.23

2.0.2 What can be done to improve this implementation

An analysis on the distribution of the computation time shows that in the "heavy cases" (binomial coefficient to the fifth and sixth powers) the bottle neck is the computation of the solution of the linear systems of equations that is required to solve the recurrence equation (*Gosper's equation*).

Therefore future optimized versions should use a special purpose linear solver that takes advantage of the specific structure of the system.

3 Manual

3.1 Loading the files

The entire package can be downloaded from the RISC Combinatorics home page at the following u.r.l. http://www.risc.uni-linz.ac.at/research/combinat/risc/
The user will find the following files:

```
algUtil.macsyma
shiftQuotient.macsyma
poly2quint.macsyma
makeGosperForm.macsyma
GosperEq.macsyma
Gosper.macsyma
```

```
Zeilberger.macsyma
LOADZeilberger.macsyma
testZeilberger.macsyma
testGosper.macsyma
```

The entire package can be loaded into memory by simply loading the file `LOAD-Zeilberger.macsyma`; this file will take care of loading the other components except the files containing some examples on which the system has been tested, namely (`testZeilberger.macsyma`, `testGosper.macsyma`).

3.2 The Commands

Zeilberger's algorithm (`Zeilberger`) as well as a parametrized version of Gosper's algorithm (`parGosper`) have been implemented in two versions: a verbose version, which allows the user to choose different levels of verbosity, and a non-verbose version [2], which should provide a bit of better performance.

3.2.1 Verbosity Levels

The levels of verbosity are selected by the user just adding a suffix to the command name (`parGosper`, `Zeilberger`) or by passing a parameter in the generic verbose version of the algorithm.

These are the levels of verbosity that have been implemented: `Summary`, `Verbose`, `VeryVerbose`, `Debugging`, `LinSys`.

Examples

`GosperVerbose(f,k)` invokes Gosper's algorithm in verbose mode

`ZeilbergerVeryVerbose(f,k,n)` invokes Zeilberger's algorithm in the very verbose mode

3.2.2 Functions' Calls

- `Zeilberger`(F, k, n)

  Given a double-indexed proper hypergeometric sequence $F(n,k)$, it computes by Zeilberger's algorithm a recurrence equation for F of the form: $\sum_{i=0}^{m} a_i(n)F(n+i,k) = \Delta_k(Cert(n,k)F(n,k))$, where the a_i's are polynomials free of k and $Cert$ ("rational certificate") is a rational function in n and k. The output will be a print-out of the recurrence and the explicit expressions for the polynomial parameters a_i and the "rational certificate" $Cert(n,k)$. $Cert(n,k)$.

- `ZeilbergerVerboseOpt`(F, k, n, verbosity)

  As `Zeilberger` but the level of verbosity is passed as a parameter.

- `parGosper`(F, k, n, ord)

  Given a double-indexed proper hypergeometric sequence, it computes, when it exists, a recurrence equation of order *ord* for F of the form: $\sum_{i=0}^{ord} a_i(n)F(n+i,k) = \Delta_k(R(n,k)F(n,k))$, where a_i's are polynomials and $Cert(n,k)$ is a rational function (the *certificate*), and it yields a sequence $[Cert(n,k), [a_0, \ldots, a_{ord}]]$; if no such recurrence exists them it yields $[0, [\text{dummy-value}, \ldots, \text{dummy-value}]]$.

- `parGosperVerboseOpt`(F, k, n, ord, verbosity)

  As `parGosper` but the level of verbosity is passed as a parameter.

3.2.3 Settings

No much settings and fine-tuning is necessary to use this package. The only setting is done through the environment variables `MAX_ORD` that sets an a priori bound on the order of the recurrence that Zeilberger's algorithm iteratively tries to find by applying `parGosper` with increasing order. The default value of `MAX_ORD` is 3.

3.3 Examples

Let us take a look at some examples, that can be found in the file `testZeilberger.macsyma`.

Example *(Binomial Theorem)*
To evaluate $\sum_{i=0}^{n} \binom{n}{k} x^k$ we can use `Zeilberger`, or `parGosper` and use the fact the we know that we expect a first order recurrence:
(prompt) `parGosper(binomial(n,k)x^k,k,n,1);`

output:

$$\left\{-\frac{k}{n-k+1}, \{-(x+1), 1\}\right\}$$

Where $-\frac{k}{-n-k+1}$ is the rational certificate $Cert(n,k)$, and $-(x+1)$ and 1 are respectivelly the polynomial coefficients a_0 and a_1 of the linear recurrence such that $a_0(n)\binom{n}{k} + a_1(n)\binom{n+1}{k} = \Delta_k(Cert(n,k)\binom{n}{k})$.

Example *(Special case of the Apéry-Schmidt-Strehl identity [1])*
(prompt) `Zeilberger(binomial(2*k,k)*binomial(n,k)^2,k,n);`
output:

$$a[0]f(n,k) + a[1]f(n+1,k) + a[2]f(n+2,k) = \Delta_k(Cert(n,k)f(n,k))$$

where

$$Cert(n,k) = -\frac{k^3(n+1)^2(4n-3k+8)}{(n-k+1)^2(n-k+2)^2}$$

and

$$a[0](n) = 9(n+1)^2$$
$$a[1](n) = -(10n^2+30n+23)$$
$$a[2](n) = (n+2)^2$$

Example *(Binomial coefficient to the fourth power)*
(prompt) `Zeilberger(binomial(n,k)^4,k,n);`
output:

$$a[0]f(n,k) + a[1]f(n+1,k) + a[2]f(n+2,k) = \Delta_k(Cert(n,k)f(n,k))$$

where

$$f(n,k) = \binom{n}{k}^4$$

and

$$Cert(n,k) = -\frac{\begin{array}{c} k^4(n+1)(74n^6 - 260kn^5 + 725n^5 + 374k^2n^4 - 2056kn^4 + 2885n^4 - \\ 276k^3n^3 - 6420kn^3 + 6045n^3 + 104k^4n^2 - 1244k^3n^2 + 5298k^2n^2 - \\ 9892kn^2 + 7030n^2 - 16k^5n + 298k^4n - 1884k^3 + 5322k^2n - 7520kn + \\ 4300n - 20k^5 + 210k^4 - 900k^3 + 1980k^2 - 2256k + 1080 \end{array}}{(n-k+1)^4(n-k+2)^4}$$

and

$$a[0](n) = -4(n+1)(4n+3)(4n+5)$$
$$a[1](n) = -2(2n+3)(3n^2+9n+7)$$
$$a[2](n) = (n+2)^3$$

[1] See [11] for more information on this identity

Acknowledgements I want to thank Professor Peter Paule for his useful suggestions.

References

[1] **Abramov, S.A.** *Rational solutions of linear differential and difference equations with polynomial coefficients*, Proc. ISSAC '95, ACM Press (1995), 285-289.

[2] **Caruso, F.** *A Macsyma Implementation of Zeilberger's Fast Algorithm*, SFB-Report No. 99-16, RISC, J.K. University, Linz, Austria (1999).

[3] **Chyzak, F.** *Fonctions holonomes en calcul formel*, Thèse universitaire no. TU 0531, INRIA. Defended on May 27 (1998).

[4] **Gerhard, J., von zur Gathen, J.** *Modern Computer Algebra* Cambridge University Press (1999), 617-622.

[5] **Gosper, R.W.** *Decision procedure for indefinite hypergeometric summation*, Proceedings of the National Academy of Sciences of USA, **75** (1978), 40-42.

[6] **Graham, R., Knuth, D. E., Patashnik, O.** *Concrete Mathematics - A Foundation for Computer Science*, Addison Wesley (1994).

[7] **Paule, P.** *Greatest factorial factorization and symbolic summation*, J. Symbolic Computation, **20** (1995), 235-268.

[8] **Paule, P., Schorn, M.** *A Mathematica Version of Zeilberger's Algorithm for Proving Binomial Coefficient Identities*, J. Symbolic Computation, **20** (1995), 673-698.

[9] **Paule, P., and Riese, A.** *A Mathematica q-Analogue of Zeilberger's Algorithm Based on an Algebraically Motivated Approach to q-Hypergeometric Telescoping*, Special Functions, q-Series and Related Topics, Fields Institute Communications, **14** (1997), 179-210.

[10] **Petkovšek, M., Wilf, H., Zeilberger, D.** *A=B*, A K Peters, MA (1996).

[11] **Strehl, V.** *Binomial identities - combinatorial and algorithmical aspects*, Discrete Mathematics, **136** (1994), 309-346.

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

[13] **Zeilberger, D.** *A fast algorithm for proving terminating hypergeometric identities*, Discrete Mathematics, **80** (1990), 207-211.

[14] **Zeilberger, D.** *The method of creative telescoping*, Journal of Symbolic Computation **11** (1991), 195-204.