Annals of Combinatorics 11 (2007) 241-247
0218-0006/07/020241-7
DOI 10.1007/s00026-007-0317-y

Annals of Combinatorics

The Holonomic Ansatz II. Automatic Discovery(!) And Proof(!!) of Holonomic Determinant Evaluations

Doron Zeilberger*

Department of Mathematics, Rutgers University (New Brunswick), Hill Center-Busch Campus, 110 Frelinghuysen Rd., Piscataway, NJ 08854-8019, USA
zeilberg@math.rutgers.edu

Received February 21, 2006

AMS Subject Classification: 05A, 33F10

Abstract. Many explicit determinant evaluations can be automatically conjectured, and then rigorously automatically proved, once we suspect that they belong to the Holonomic Ansatz.

Keywords: computer-generated combinatorics, explicit evaluations of determinants

Prerequisites

I assume that readers have read [Z1].

Experimental Mathematics: From Computer-Assisted to Computer-Generated Mathematics

In a wonderful essay [W] on Experimental Mathematics, Herb Wilf outlines the four steps of doing Experimental Mathematics, in the way it is usually practiced today.

1. Wondering, by a human, what a "particular situation looks like in detail".
2. Some computer experimentation to show the structure of that situation for a selection of small values of the parameters.
3. The [human] mathematician *gazes* at the computer output, attempting to see or to codify some pattern, that hopefully leads him or her to formulate a *conjecture*.
4. Human-made proof of the human-made conjecture (that was computer-inspired).

* http://www.math.rutgers.edu/~zeilberg/.
Accompanied by the Maple package DET available from
`http://www.math.rutgers.edu/~zeilberg/tokhniot/DET`.
Supported in part by the NSF.

Under this scheme, only step **2** employs the computer. In the present series of articles, I illustrate, **by example**, how computers can be used, **without any human intervention**, to also do steps **3** and **4**. As for step **1**, the wondering, this can also be done by machinekind — it is not too hard to teach the computer how to wonder. All that we, humans, ultimately would have to do is *meta-wonder*. In other words, make up new **ansatzes** and write **once and for all** computer programs teaching the computer how to *wonder* in these ansatzes, **then** *gaze* at the pattern, **then** *formulate a conjecture* (within the given ansatz) and **then**, finally, *prove* the conjecture, **all by itself**, without any human intervention! No longer just **computer-assisted** but fully **computer-generated**.

The Art of Determinant Evaluations

To find out about the state of the art in contemporary explicit determinant evaluations, by *homo sapiens*, the reader should consult Christian Krattenthaler's beautiful surveys [K1] and [K2].

Shalosh B. Ekhad vs. Some Great Human Mathematicians

Consider the determinant evaluation

$$\det\left(\binom{\mu+i+j}{2i-j}\right)_{0\le i,j\le n-1} = \prod_{i=1}^{n-1} Nice(\mu,n,i), \qquad (MRR)$$

where *Nice* is some explicit expression whose exact form I omit right now in order not to distract from the general ideas. This determinant-evaluation, discovered and first proved by William Mills, David Robbins, and Howard Rumsey [MRR1], was so attractive that other great mathematicians, notably George Andrews and Dennis Stanton [AS], Marko Petkovšek and Herb Wilf [PW], and Christian Krathenthaler [K1] took the trouble to find other proofs.

But if you have Maple, and downloaded the Maple package `DET` into your computer, and gotten into Maple by typing `maple`, and typed `read DET:`, then typing:

`RproofP(binomial(m+n+p,2*m-n+1),m,n,N,30,R,p,40,60):` ,

and waiting 256 seconds of CPU time will, completely *ab initio*, conjecture (MRR) and immediately proceed to prove it *fully rigorously*.

Once written, `DET` can *discover(!)* and *prove(!!)* countless other determinant evaluations, provided they belong to the right *ansatz*, the holonomic ansatz in our case. You are welcome to look at the webpage of this article

`http://www.math.rutgers.edu/~zeilberg/mamarim/mamarimhtml/ansatzII.html`

for numerous examples of input and output, in addition to the Mills-Robbins-Rumsey evaluation mentioned above. If you have Maple, you can generate many more examples on your own.

Caveat

Unlike WZ theory, where the computer is guaranteed to give an output (time and space permitting), this is not the case here. It is not known, and is probably false, that it is always the case that if $a(i, j)$ is holonomic, and setting $A(n) := det(a(i, j))_{0 \le i, j \le n-1}$, then $B(n) := A(n+1)/A(n)$ is holonomic in n. But it happens often enough to justify asking the computer to give it a try.

The General Idea

Of course, the seed of the method *did* originate from humans. It can be formulated in two equivalent ways. We have:

Problem: Given some explicit expression $m(i, j)$, define the $n \times n$ matrix M_n to be

$$M_n := (m(i, j))_{0 \le i, j \le n-1}.$$

Find an *explicit* (in some sense) evaluation (in n) for

$$A(n) := \det M_n.$$

George Andrews's Approach

George Andrews pulls out of a hat an *upper-triangular* matrix U_n whose entries are "nice", and whose diagonal entries are all 1's, and such that $L_n := M_n U_n$ is lower-triangular and has a "nice" diagonal (but the other entries are possibly ugly). Then since $\det(M_n) = \det(L_n)/\det(U_n)$, and $\det(U_n) = 1$, we can express $\det(M_n)$ as a product of nice things, and hence it is nice itself.

Of course, the reader is never told how U_n was conjectured, it is just pulled out of the blue. To prove the assertion one has to prove that L_n is indeed lower-triangular, i.e., the entries of $M_n U_n$ above the diagonal are all 0, which boils down to (usually) proving a hypergeometric identity. Next, one has to prove that the diagonal entries of L_n are as claimed, which involves another hypergeometric identity. These are sometimes proved by computer, using the Zeilberger algorithm, but still in a piecemeal, **human-centric**, way.

Dave Robbins's Approach

Although mathematically equivalent to George Andrews's LU approach, I find Dave Robbins's approach, described in [MRR2], more conducive for teaching a computer.

Consider the $n+1$ by $n+1$ matrix $(a(i, j))$, for $0 \le i, j \le n$. Let $A(n, j)$ $(0 \le j \le n)$, be the cofactor of the (n, j) entry. Then of course

$$\det(a(i, j))_{0 \le i, j \le n} = \sum_{j=0}^{n} A(n, j) a(n, j).$$

Now the $n+1$ unknowns $A(n, j)$, $j = 0, \ldots, n$ are uniquely determined, up to a normalization factor (that only depends on n), by the n linear homogeneous equations

$$\sum_{j=0}^{n} A(n, j) a(i, j) = 0, \quad i = 0, \ldots, n-1.$$

The normalized cofactors $A'(n,j)$ defined by $A'(n,j) = A(n,j)/A(n,n)$, are then determined uniquely, for each specific n, by the system of linear equations

$$\sum_{j=0}^{n} A'(n,j)a(i,j) = 0, \quad i = 0,\ldots,n-1, \tag{$Dave1$}$$

subject to

$$A'(n,n) = 1. \tag{$Dave2$}$$

Now the human can ask the computer to crank out $A'(n,j)$ for $0 \leq j,n \leq N$ for some N (say 100), *gaze* at the output, and conjecture some 'explicit' form for $A'(n,j)$ and then prove that they indeed satisfy $(Dave1)$ and $(Dave2)$.

Finally, if in luck,

$$B(n) := \sum_{j=0}^{n} A'(n,j)a(n,j), \tag{$Dave3$}$$

turns out to be 'explicit' , and the human is clever enough to prove it. Since $A(n,n) = \det(a(i,j))_{0\leq i,j\leq n-1}$, it follows that

$$\frac{\det(a(i,j))_{0\leq i,j\leq n}}{\det(a(i,j))_{0\leq i,j\leq n-1}} = B(n),$$

that finally entails that

$$\det(a(i,j))_{0\leq i,j\leq n} = \prod_{j=0}^{n} B(j),$$

which should be considered 'nice' if $B(j)$ is.

The Limit of Humans

If $A'(n,j)$ turns out to be *closed-form* then some clever humans can gaze at it and conjecture an expression for it. Other humans will 'cheat' and use a computer program (like Krattenthaler's `rate`) to do the guessing, but in an *interactive* way.

If nothing nice emerges, then these humans use some tricks of the trade, that only they know, and are unwilling (and usually also unable) to divulge, to express the normalized co-factor $A'(n,j)$ as a single- or double- hypergeometric sum. Then go through excruciating pain to verify $(Dave1)$ and evaluate $B(n)$, using $(Dave3)$. This is indeed the most challenging case, when $A'(n,j)$ does not happen to be a hypergeometric term, but turns out to be a hypergeometric sum (or even a multisum).

Enter Computers

When $A'(n,j)$ turns out to be closed-form, it is easy enough to teach the computer to guess it empirically (and later on prove it, see below), but trying to represent it as a hypergemoetric sum or multisum is an art rather than a science, and does not seem to be amenable to computerization. But we know, thanks to WZ theory, that hypergeometric sums and multisums belong to the Holonomic Ansatz (see [Z1]), so why not start there?

Computers do not play favorites. When instructed to operate within the *Holonomic Ansatz*, they have no particular fondness for hypergeometric terms. A discrete function of two variables being a hypergeometric term is just the *special case* of the defining recurrences being *first-order*. So staying within the *holonomic ansatz*, once the computer generated enough numerical data $A'(n, j)$ for $0 \leq n, j \leq N$ for a big enough N, instead of *gazing*, it keeps **guessing** linear recurrence equations with polynomial coefficients (see [Z1]) in the n direction and in the j direction, not necessarily of the first-order. It also tries to guess a linear recurrence equation with polynomial coefficients satisfied by $B(n)$, once enough specific values of $B(n)$, for, say, $n \leq 100$ (or whatever), have been generated by $(Dave3)$.

Once conjectured, the rest, i.e., proving that $(Dave1)$, $(Dave2)$, and $(Dave3)$ hold when the 'real' $A'(n, j)$ and $B(n)$ are replaced by the sequences defined by these conjectured recurrences (subject to the obvious initial conditions), is algorithmically decidable thanks to [Z2], at least in principle, but thanks to Frederic Chyzak's beautiful work [C], probably also in practice. Once this is completed, it follows (rigorously!), by the *uniqueness* of the solution to $(Dave1)$ and $(Dave2)$, that the conjectured recurrences for $A'(n, j)$ are indeed correct, and since $(Dave3)$ uniquely determines $B(n)$, that the conjectured recurrence for $B(n)$ is indeed true.

The Lucky Case

When the conjectured recurrences for the normalized cofactors $A'(n, j)$ turn out to be first-order, then they can be easily solved explicitly, and $A'(n, j)$ turns out to be (conjecturally for now) a *hypergeometric term* in (n, j). In other words it is *closed form*, and verifying that that conjectured closed-form is indeed correct boils down to proving that $(Dave1)$ and $(Dave2)$ with $A'(n, j)$ replaced by that conjectured expression indeed hold. Then one also needs to verify that the conjectured expression (or recurrence) for $B(n)$ is compatible with $(Dave3)$ with the $A'(n, j)$ replaced by the (now proven) expression for it.

All this can be done with the original 'fast' Zeilberger algorithm [Z3] and [Z4], implemented in my Maple package `EKHAD` (available from my website), as well as by the built-in Maple package `SumTools`. The relevant part of `EKHAD` has been included in the Maple package `DET`. The procedures that handle this lucky case are `Rproof` and `RproofP`. They give fully rigorous proofs of the conjectured expression or recurrence for $B(n)$ (even when that recurrence is of higher order, all we need is that the conjectured recurrences for $A'(n, j)$ be of first order in order to take advantage of `Rproof` and `RproofP`).

The Unlucky Case

When the conjectured recurrences for the normalized cofactors $A'(n, j)$ turn out to be of higher order, we need to use Frederic Chyzak's [C] package instead. Since I am not familiar enough with it, this case is not yet implemented, and currently `Rproof` returns `FAIL` in that case. Also, it is possible that even once Chyzak's program will get interfaced with `DET`, it will take too long, in which case read the next section.

What If It Takes too Long? Let's Settle for a Semi-Rigorous Proof

Since we *know* that if we had a big enough computer, and good enough software, we can prove that $(Dave1)$, $(Dave2)$, and $(Dave3)$ hold when $A'(n, j)$ and $B(n)$ are replaced by their conjectured 'explicit expressions' (in the sense of the Holonomic Ansatz, where 'explicit expressions' are given implicitly as solutions of linear recurrence equations with polynomial coefficients), is it really worth the trouble? We can prove these facts semi-rigorously as follows.

We can use the conjectured linear recurrences for $A'(n, j)$ and $B(n)$ to crank out many more conjectured values, much faster than by solving the system of equations $(Dave1)$, $(Dave2)$ and then computing the $B(n)$ by $(Dave3)$.

Let's temporarily call these new values $A''(n, j)$ and $B'(n)$ and once found, plug them into the analogs of $(Dave1)$, $(Dave2)$, and $(Dave3)$ obtained by replacing $A'(n, j)$ by $A''(n, j)$ and $B(n)$ by $B'(n)$. If they turn out to be right for, say, $n, j \leq 10000$, then it means that the conjectured recurrences are right at least for $n \leq 10000$, but most likely we have also proved them for *all* n, even though we don't have a rigorous proof that this is indeed a rigorous proof (but I bet it is!).

It is like proving that two polynomials are the same by plugging-in enough special values. In the case of polynomials of one variable, there is an easy parameter, the degree plus 1, that tells you how many special values you have to plug-in in order to rigorously prove a conjectured identity between two given polynomials. The problem now is that it is not so easy to find the analog of the *degree* for a holonomic function, but if we check them for $n \leq 10000$, this is most likely more than enough. The notion of *semi-rigorous* proof was introduced in [Z5].

A Note on Guessing

There now exist powerful guessing programs, for example `superseeker` in Neil Sloane's famous site, Bruno Salvy and Paul Zimmermann's [SZ] Maple packages `Gfun` and `Mgfun`, and Christian Krattenthaler's Mathematica program `rate`, that could have been used as subroutines in the present endeavor. But I found it easier to program everything *ab initio*, borrowing freely, of course, from these pioneering programs.

The Maple Package DET

Everything here is implemented in the Maple package `DET` available from

`http://www.math.rutgers.edu/~zeilberg/tokhniot/DET`.

As already mentioned, the webpage of this article

`http://www.math.rutgers.edu/~zeilberg/mamarim/mamarimhtml/ansatzII.html`

contains sample input and output.

The main functions are `Rproof, RproofP, SRproof, SRproofI`. To find out about them, in `DET`, type `ezra(FunctionName);`. For example, for help with `Rproof`, type `ezra(Rproof);`.

The 'engines' driving these functions are procedures `DaveH` and `DaveV`, that guess horizontal and vertical pure recurrences respectively for the normalized cofactors $A'(n, m)$. Full details can be gotten by reading the source code of `DET`. Let me just mention that `DaveH` works by conjecturing recurrences, in m, for many rows (i.e., for fixed n), and then using procedure `GR1`, that guesses rational functions, in order to 'interpolate' them, thereby guessing a unified recurrence, also featuring n symbolically.

Future Directions

`DET` can be further optimized, and translated into Matlab or C, thereby probably making it much faster. It would be also worthwhile to interface it with Frederic Chyzak's packages thereby turning the semi-rigorous proofs, that we presently have to contend with in the *unlucky* case mentioned above, into fully rigorous proofs.

Finally, it should be relatively easy to write the q-analog of `DET`, but the running times would be larger because of the extra symbol q.

References

[AS] G.E. Andrews and D.W. Stanton, Determinants in plane partition enumeration, Europ. J. Combin. **19** (1998) 273–282.

[C] F. Chyzak, An extension of Zeilberger's fast algorithm to general holonomic functions, Discrete Math. **217** (2000) 115–134.

[K1] C. Krattenthaler, Advanced determinant calculus, In: The Andrews Festschrift, D. Foata and G.-N. Han Eds., Sém. Lothar. Comb. **42** (1999) B42q.

[K2] C. Krattenthaler, Advanced determinant calculus: a complement, Linear Algebra Appl. 411 (2005) 68–166. Avaiable from: `http://arXiv.org/abs/math.Co/0503507`.

[MRR1] W.H. Mills, D.H. Robbins, and H. Rumsey, Enumeration of a Symmetry Class of Plane Partitions, Discrete Math. **67** (1987) 43–55.

[MRR2] W.H. Mills, D.H. Robbins, and H. Rumsey, Proof of the Macdonald conjecture, Invent. Math. **66** (1982) 73–87.

[PW] M. Petkovšek and H. Wilf, A high-tech proof of the Mills-Robbins-Rumsey determinant formula, Electron. J. Combin. **3** (2) (1996) #R19.

[SZ] B. Salvy and P. Zimmermann, Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable, ACM Trans. Math. Software **20** (2) (1994) 163–177.

[W] H.S. Wilf, Mathematics: an experimental science, to appear in "Princeton Companion to Mathematics", W.T. Gowers Ed., Princeton University Press. Available from: `http://www.math.upenn.edu/~wilf/reprints.html`.

[Z1] D. Zeilberger, Foundations The Holonomic ansatz I. foundations and applications to lattice walk counting, submitted. Available from: `http://www.math.rutgers.edu/~zeilberg/mamarim/mamarimhtml/ansatzI.html`.

[Z2] D. Zeilberger, A Holonomic systems approach to special functions identities, J. Comput. Appl. Math. **32** (3) (1990) 321–368.

[Z3] D. Zeilberger, A fast algorithm for proving terminating hypergeometric identities, Discrete Math. **80** (2) (1990) 207–211.

[Z4] D. Zeilberger, The method of creative telescoping, J. Symbolic Comput. **11** (3) (1991) 195–204.

[Z5] D. Zeilberger, Theorems for a price: tomorrow's semi-rigorous mathematical culture, Notices Amer. Math. Soc. **40** (8) 978–981; reprinted in Math. Intelligencer **16** (4) 11–14.