Computer Proofs of Hypergeometric Identities

Etienne Rassart
Massachusetts Institute of Technology

Primera Conferencia Iberoamericana de
Matemática Computacional

Slides available from

`http://www-math.mit.edu/~rassart/pub/`

Plan

Part I

1. Hypergeometric sums
   - What they are
   - The canonical notation ${}_pF_q$
   - Applications
2. Proving hypergeometric identities automatically
   - Hypergeometric identities
   - The standard form
   - Verifying an identity given a proof certificate
   - Famous identities

Plan

Part II

1. The theory (based on the book $A = B$ by P-W-Z)
   - Sister Celine Fasenmyer's algorithm
   - Gosper's algorithm
   - Zeilberger's algorithm
   - Petkovšek's algorithm
2. Recent developments
   - Parallel with differential topology
   - Elimination theory (non-commutative Gröbner bases, Ore algebras)

Geometric sums

Hypergeometric sums were first studied by Euler, and then by Gauss, Riemann, Kummer, and many others. They are a natural generalization of geometric sums.

Definition A sum

$$\sum_k t_k$$

is *geometric* if the ratio of two consecutive terms t_{k+1}/t_k is a constant with respect to k, i.e.

$$\frac{t_{k+1}}{t_k} = c$$

for some (complex) number c not depending on k.

Hypergeometric sums

Definition A sum

$$\sum_k t_k$$

is *hypergeometric* if the ratio of two consecutive terms t_{k+1}/t_k is a rational function of the summation index k, i.e. if we have

$$\frac{t_{k+1}}{t_k} = \frac{P(k)}{Q(k)}$$

for polynomials $P, Q \in \mathbb{C}[k]$ $(Q \neq 0)$.

The canonical notation ${}_pF_q$

Suppose $t_0 = 1$ and

$$\frac{t_{k+1}}{t_k} = \frac{P(k)}{Q(k)} = \frac{(k+a_1)(k+a_2)\dots(k+a_p)}{(k+b_1)(k+b_2)\dots(k+b_q)(k+1)}x$$

where x is a constant.

We denote the sum $\sum_{k\geq 0} t_k$ by

$${}_pF_q\left[\begin{matrix} a_1 & a_2 & \dots & a_p \\ b_1 & b_2 & \dots & b_q \end{matrix}\, ;\; x\right].$$

Remark The extra $(k+1)$ factor in the denominator of the factorized ratio of p and Q above is there for historical reasons mostly, but also because people in the past have written their series in exponential form $\sum_k \frac{t_k}{k!}$ instead of the ordinary form $\sum_k t_k$.

EXAMPLE The exponential function is a hypergeometric series:

$$e^x = \sum_{k \geq 0} \frac{x^k}{k!}$$

so that, with the notation above, $t_k = x^k/k!$ and $t_0 = 1/0! = 1$.
Then $\dfrac{t_{k+1}}{t_k} = \dfrac{x^{k+1}/(k+1)!}{x^k/k!} = \dfrac{1}{k+1}x$. Therefore

$$e^x = {}_0F_0 \left[\begin{matrix} - \\ - \end{matrix} ; \; x \right] .$$

EXAMPLE Consider the sum $\sum_k (-1)^k \binom{n}{k}^2 \binom{3n+k}{2n}$. The first non zero term of this sum occurs at $k = 0$ with $t_0 = \binom{3n}{2n}$.

$$\frac{t_{k+1}}{t_k} = \frac{(-1)^{k+1}\binom{n}{k+1}^2\binom{3n+k+1}{2n}}{(-1)^k\binom{n}{k}^2\binom{3n+k}{2n}} = (-1)\frac{(k-n)(k-n)(k+3n+1)}{(k+n+1)(k+1)(k+1)}.$$

Hence

$$\sum_k (-1)^k \binom{n}{k}^2 \binom{3n+k}{2n} = \binom{3n}{2n} {}_3F_2\left[\begin{matrix} -n & -n & 3n+1 \\ & 1 & n+1 \end{matrix} ; -1\right].$$

Applications

Combinatorics Combinatorics is an endless source of hypergeometric sums because the question "How many structures of this type (trees, graphs, etc) are there on n points?" often is a hypergeometric sum, as are many generating series.

Probability Calculating expectations and moment generating functions, among other things, sometimes involve hypergeometric sums.

Physics Hypergeometric sums appear as partition functions of systems of particles in statistical physics.

Number Theory Many of the fundamental constants have series expressions that are hypergeometric. Analytic number theory is full of hypergeometric sums. Also WZ theory provides ways to accelerate the convergence of certain series, for example

$$\zeta(3) = \sum_{n=0}^{\infty} (-1)^n \frac{n!^{10}(205n^2 + 250n + 77)}{64(2n+1)!^5} .$$

WZ theory also provides a way to systematize the proof of the irrationality of certain numbers, like $\zeta(3)$ for instance.

Proving hypergeometric identities automatically

The theory developed by Wilf and Zeilberger, building on work by Sister Celine Fasenmeyer and Gosper, provides a way to make computers prove a large class of hypergeometric identities.

Definition A *hypergeometric identity* is an identity of the form

$$\sum_k f(n,k) = h(n)$$

where the left hand side of the equation is a hypergeometric sum (n is considered a parameter).

Standard form for identities

Consider the hypergeometric identity $\sum_k f(n,k) = h(n)$.

- Assume that $f(n,k)$ is zero for all k outside of a finite interval.
- Assume also that the sum is taken over all k in that interval.
- Let

$$F(n,k) = \begin{cases} \dfrac{f(n,k)}{h(n)} & \text{if } h(n) \neq 0 \\ f(n,k) & \text{if } h(n) = 0. \end{cases}$$

- Checking the initial identity is the same as checking that $\sum_k F(n,k) = 1$ if $h(n) \neq 0$, or checking that $\sum_k F(n,k) = 0$ if $h(n) = 0$.

The crucial part

- Imagine there exists a $G(n,k)$ such that

$$F(n+1,k) - F(n,k) = G(n,k+1) - G(n,k)\,,$$

and suppose it vanishes outside a k-interval like $F(n,k)$.

- Then if we sum this equation over all k, the $G(n,k)$ cancel each other out and the left hand side telescopes to 0:

$$\sum_k \big(F(n+1,k) - F(n,k)\big) = \sum_k \big(G(n,k+1) - G(n,k)\big) \equiv 0\,.$$

- This says that

$$\sum_k F(n+1,k) = \sum_k F(n,k)\,.$$

- But this means that $\sum_k F(n,k)$ is independent of n.
- So it is a constant.
- We evaluate the value of the constant by plugging any value of n (usually 0) in the sum.
- If $h(n) \neq 0$, the constant should be 1, and if $h(n) = 0$, it should be 0.

The certificate

- An important part of a good computer proof is being able to verify it.
- Zeilberger's algorithm (described later) gives a way of computing the "magical" function $G(n,k)$, if $F(n,k)$ satisfies certain conditions.
- The great thing about this is that although $F(n,k)$ and $G(n,k)$ might be very complicated hypergeometric terms, they will be related by

$$G(n,k) = R(n,k)F(n,k)$$

where $R(n,k)$ is a rational function called the *certificate*.

EXAMPLE Let's consider the identity $\sum_k \binom{n}{k} = 2^n$.

Using algorithms that will be presented later, the computer can prove this identity, and outputs the certificate

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

Let's verify by hand, using the certificate, that the computer's proof is correct.

Step 0

- In our case, $h(n) = 2^n \neq 0$, so we set

$$F(n,k) = \frac{\binom{n}{k}}{2^n} = \binom{n}{k} 2^{-n}\,.$$

Step 1

- We set $G(n,k) = R(n,k)F(n,k)$:

$$\begin{aligned} G(n,k) &= \frac{k}{2(k-n-1)}\binom{n}{k}2^{-n} \\ &= -\frac{kn!2^{-n}}{2(n+1-k)k!(n-k)!} = -\binom{n}{k-1}2^{-n-1} \end{aligned}$$

Step 2

- We have to verify that

$$F(n+1,k) - F(n,k) = G(n,k+1) - G(n,k)\,.$$

$$\begin{aligned} F(n+1,k) - F(n,k) - G(n,k+1) + G(n,k) \\ &= \binom{n+1}{k}2^{-n-1} - \binom{n}{k}2^{-n} + \binom{n}{k}2^{-n-1} - \binom{n}{k-1}2^{-n-1} \\ &= 2^{-n-1}\left(\binom{n+1}{k} - \binom{n}{k} - \binom{n}{k-1}\right) \\ &= 0 \qquad \text{(Pascal's triangle)} \end{aligned}$$

- Hence

$$\sum_k \binom{n+1}{k}2^{-(n+1)} - \sum_k \binom{n}{k}2^{-n} \equiv 0$$

and so $\sum_k \binom{n}{k}2^{-n}$ is a constant independent of n.

Step 3

- We compute the constant by evaluating the sum for one value of n:

$$\sum_k \binom{0}{k} 2^{-0} = \binom{0}{0} = 1\,.$$

- Therefore, the WZ algorithm proves that $\sum_k \binom{n}{k} 2^{-n} = 1$ for all n (positive integer) and thus

$$\sum_k \binom{n}{k} = 2^n\,.$$

Famous identities

Definition For a nonnegative integer n, we denote by $(a)_n$ the rising factorial:

$$(a)_n = a(a+1)(a+2)\ldots(a+n-1)\,.$$

Gauss's $_2F_1$ identity. If b is a nonpositive integer or $c-a-b$ has positive real part, then

$${}_2F_1\left[\begin{matrix} a & b \\ c & \end{matrix}\;;\;1\right] = \frac{\Gamma(c-a-b)\Gamma(c)}{\Gamma(c-a)\Gamma(c-b)}\,,$$

where Γ is Euler's gamma function.

Saalschütz's ${}_3F_2$ identity. If $d+e=a+b+c+1$ and c is a negative integer, then

$$ {}_3F_2\left[\begin{matrix} a & b & c \\ d & e & \end{matrix} ; 1\right] = \frac{(d-a)_{|c|}(d-b)_{|c|}}{(d)_{|c|}(d-a-b)_{|c|}} . $$

Clausen's ${}_4F_3$ identity. If d is a nonpositive integer and $a+b+c-d=\frac{1}{2}$, and $e=a+b+\frac{1}{2}$, and $a+f=d+1=b+g$, then

$$ {}_4F_3\left[\begin{matrix} a & b & c & d \\ e & f & g & \end{matrix} ; 1\right] = \frac{(2a)_{|d|}(a+b)_{|d|}(2b)_{|d|}}{(2a+2b)_{|d|}(a)_{|d|}(b)_{|d|}} . $$

All these identities and many more can be proved by the computer using WZ theory and the method described above.

WZ Theory

- Sister Celine Fasenmyer's algorithm (1945)
- Gosper's algorithm (1978)
- Zeilberger's algorithm (1990)
- Petkovšek's algorithm (1991)

Sister Celine Fasenmyer's algorithm

Given a hypergeometric sum $f(n) = \sum_k F(n,k)$, a good first step towards finding a nice formula for $f(n)$ is computing a recurrence that it satisfies.

Sister Celine's algorithm does that by finding a linear recurrence satisfied by the $F(n,k)$, whose coefficients don't involve the summation index k.

- For the algorithm to work, $F(n,k)$ should be *doubly hypergeometric*, meaning that

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

  are both rational functions of n and k.

- The algorithm is guaranteed to stop if $F(n,k)$ is a *proper hypergeometric term*.

Definition $F(n,k)$ is a *proper hypergeometric term* if it can be written in the form

$$F(n,k) = P(n,k)\frac{\prod_{i=1}^{M_1}(a_in+b_ik+c_i)!}{\prod_{i=1}^{M_2}(u_in+v_ik+w_i)!}x^k$$

where

- x is an indeterminate,
- P is a polynomial,
- the a_i, b_i, u_i and v_i, are specific integers, i.e. not depending on n, k or other parameters,
- M_1 and M_2 are specific nonnegative integers.

How it works

We want to find a recurrence satisfied by $f(n)$, where $f(n) = \sum_k F(n,k)$.

- Given a doubly hypergeometric term $F(n,k)$, the algorithm tries to find a recurrence of the form

$$\sum_{i=0}^{I} \sum_{j=0}^{J} a_{i,j}(n) F(n-j, k-i) = 0\,.$$

- It is very important that the $a_{i,j}$ depend only on n and not on k, because if the algorithm succeeds, then we can find a recurrence for the sum over all k:

$$f(n-j) = \sum_k F(n-j, k-i)$$

and therefore

$$\sum_{i=0}^{I} \sum_{j=0}^{J} a_{i,j}(n) f(n-j) = 0 \,,$$

which we can rewrite as

$$\sum_{j=0}^{J} \left(\sum_{i=0}^{I} a_{i,j}(n) \right) f(n-j) = 0 \,.$$

This recurrence is linear in the $f(n-j)$, and it turns out that when the algorithm succeeds, the $a_{i,j}$ are rational functions of n. So we can clear out the denominators and get a linear recurrence with polynomial coefficients.

We can then try to solve this recurrence with Petkovšek's algorithm or other techniques.

The algorithm

1. Fix initial values of I and J (usually $I = J = 1$).

2. Set up the recurrence

$$\sum_{i=0}^{I} \sum_{j=0}^{J} a_{i,j}(n) F(n-j, k-i) = 0$$

   where the $a_{i,j}(n)$ are undetermined (variables to be solved for).

3. Divide the whole recurrence by $F(n,k)$. Because $F(n,k)$ is doubly hypergeometric, $F(n-j,k-i)/F(n,k)$ will be rational in n and k. So we are left with rational functions only.

4. Put the expression over a common denominator. The numerator is a polynomial in n and k, which we can write as a polynomial in k (over $\mathbb{C}[n]$).

5. The only way for the rational expression to be zero is if its numerator is zero. The numerator will be zero for all n if and only if the coefficient of each power of k in it vanishes.

6. This gives a system of linear equations in the $a_{i,j}$ (n is considered a parameter). If the system has a solution, the algorithm has found the recurrence. Otherwise we increase one of I or J (or both) and restart.

EXAMPLE Consider $F(n,k) = k\binom{n}{k}$ and $f(n) = \sum_k F(n,k)$. The summand $F(n,k)$ is a proper hypergeometric term because we can write it as

$$F(n,k) = k\binom{n}{k} = k\frac{n!}{k!\,(n-k)!}\,.$$

Let's run Sister Celine's algorithm:

1. We set $I = J = 1$.
2. So we are looking for a recurrence of the form

$$a_{0,0}F(n,k)+a_{1,0}F(n,k-1)+a_{0,1}F(n-1,k)+a_{1,1}F(n-1,k-1) = 0\,.$$

3. We then divide by $F(n,k)$ throughout. After simplification, this gives

$$a_{0,0} + a_{1,0}\frac{k-1}{n-k+1} + a_{0,1}\frac{n-k}{n} + a_{1,1}\frac{k-1}{n} = 0\,.$$

4. We then write this expression over a common denominator and take the numerator as a polynomial in k:

$$\begin{array}{rc} & (a_{0,1} - a_{1,1})k^2 \\ + & (a_{1,0}n + a_{1,1}n - 2a_{0,1}n - a_{0,0}n + 2a_{1,1} - a_{0,1})k \\ + & (a_{0,0}n^2 + a_{0,1}n^2 + a_{0,0}n + a_{0,1}n - a_{1,0}n - a_{1,1}n - a_{1,1}) \end{array}$$

5. We have to set the coefficients of the powers of k equal to zero so that the numerator is zero:

$$\begin{array}{rcrcrcrcl} & & a_{0,1} & & & - & a_{1,1} & = & 0 \\ -na_{0,0} & - & (2n+1)a_{0,1} & + & na_{1,0} & + & (n+2)a_{1,1} & = & 0 \\ (n^2+n)a_{0,0} & + & (n^2+n)a_{0,1} & - & na_{1,0} & - & (n+1)a_{1,1} & = & 0 \end{array}$$

6. This system has the solutions

$$\begin{bmatrix} a_{0,0} \\ a_{0,1} \\ a_{1,0} \\ a_{1,1} \end{bmatrix} = a_{0,1} \begin{bmatrix} -\frac{n-1}{n} \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

If a recurrence is satisfied by $F(n,k)$, then every scalar multiple of it is also satisfied, so we could take $a_{0,1} = 1$.

However, since n is considered a parameter in this case, it is a scalar, and to avoid having rational coefficients, we can take $a_{0,1} = n$ and get polynomial coefficients.

Hence $F(n,k) = k\binom{n}{k}$ satisfies the recurrence

$$(n-1)F(n,k) = nF(n-1,k-1) + nF(n-1,k)\,.$$

We can then sum over all k to get that $f(n) = \sum_k k\binom{n}{k}$ satisfies the recurrence

$$(n-1)f(n) = 2nf(n-1) \qquad \text{or} \qquad f(n) = 2\frac{n}{n-1}f(n-1)\,,$$

which we can solve by iteration for

$$f(n) = 2^{n-1}nf(1) = 2^{n-1}n\,.$$

$$F(n,k) = P(n,k)\frac{\prod_{i=1}^{M_1}(a_i n + b_i k + c_i)!}{\prod_{i=1}^{M_2}(u_i n + v_i k + w_i)!}x^k$$

THEOREM ***(Sister Celine Fasenmyer, 1945)***
If $F(n,k)$ is a proper hypergeometric term, then there exist positive integers I, J, and polynomials $a_{i,j}(n)$ for $i = 0, \ldots, I$, $j = 0, \ldots, J$, not all zero, such that

$$\sum_{i=0}^{I}\sum_{j=0}^{J} a_{i,j}(n)F(n-j, k-i) = 0$$

where

$$\begin{aligned} J &= \sum_s |b_s| + \sum_s |v_s| \\ I &= 1 + \deg(P) + J\left(\left(\sum_s |a_s| + \sum_s |u_s|\right) - 1\right). \end{aligned}$$

Summary (Sister Celine's algorithm)

- Sister Celine's algorithm is useful, but it gets very slow as I and J grow large.
- It has the disadvantage of not really providing a nice formula for the hypergeometric sum, just a recurrence it satisfies.
- It also requires $F(n, k)$ to be summed up over the whole k-range.
- The good thing is that it is completely algorithmic and can be done by a computer.

Gosper's algorithm

- Hypergeometric sums don't always range over the full set of indices where the summand doesn't vanish.
- Gosper's algorithm is more powerful than Sister Celine's in that sense, because it can decide whether a hypergeometric sum has an indefinite sum of a certain type.
- Indefinite sums are like indefinite integrals. $F(x)$ is the indefinite integral of $f(x)$ if

$$\int_a^b f(x)\,dx = F(b) - F(a) \qquad \forall\, a, b.$$

Similarly, $H(k)$ is the indefinite sum of $h(k)$ if

$$\sum_{k=a}^{b-1} h(k) = H(b) - H(a)\,.$$

- Running Gosper's algorithm on a hypergeometric term is like trying to find the antiderivative of a function in calculus.

Definition A function t_k of k is a *hypergeometric term* if t_{k+1}/t_k is a rational function of k.

What Gosper's algorithm does

Gosper's algorithm can decide whether a hypergeometric term has an indefinite sum which is also a hypergeometric term. If it exists, the algorithm will find it.

What we want to do

Given a sum $\sum_{k=a}^{b} t_k$ where t_k is a hypergeometric term, suppose that we can find another function z_k of k such that

$$t_k = z_{k+1} - z_k \, .$$

Then

$$\begin{aligned}\sum_{k=a}^{b-1} t_k &= (z_{a+1} - z_a) + (z_{a+2} - z_{a+1}) + \ldots + (z_b - z_{b-1}) \\ &= z_b - z_a \, .\end{aligned}$$

So for such a z_k, the sum telescopes and we have the "antisum", or indefinite sum, that we we looking for.

Gosper's algorithm decides whether a hypergeometric term t_k has an indefinite sum z_k which is also a hypergeometric term.

How it's done

- We start by assuming we have a solution z_k (a hypergeometric term) to the equation $t_k = z_{k+1} - z_k$, and see what form it would have to take.

- If z_k is a solution, then its ratio with t_k is

$$\frac{z_k}{t_k} = \frac{z_k}{z_{k+1} - z_k} = \frac{1}{\frac{z_{k+1}}{z_k} - 1},$$

which is a rational function of k since z_k is a hypergeometric term.

- So $z_k = y(k)t_k$ for some rational function $y(k)$.

- The equation $t_k = z_{k+1} - z_k$ becomes
  $t_k = y(k+1)t_{k+1} - y(k)t_k$ or

$$1 = y(k+1)r(k) - y(k)$$

  where $r(k)$ is the rational function t_{k+1}/t_k.

- Now comes the tricky part. Suppose we can write $r(k)$ in the form

$$r(k) = \frac{a(k)}{b(k)} \frac{c(k+1)}{c(k)}$$

  where $a(k)$, $b(k)$, $c(k)$ are polynomials, and where $a(k)$ and $b(k+h)$ have no common factors for any nonnegative integer h.

- Then suppose we write

$$y(k) = \frac{b(k-1)x(k)}{c(k)}$$

where $x(k)$ is some rational function.

- The equation $1 = y(k+1)r(k) - y(k)$ becomes

$$\begin{aligned} 1 &= \frac{b(k)x(k+1)}{c(k+1)} \frac{a(k)}{b(k)} \frac{c(k+1)}{c(k)} - \frac{b(k-1)x(k)}{c(k)} \\ &= \frac{a(k)x(k+1) - b(k-1)x(k)}{c(k)}, \end{aligned}$$

or

$$a(k)x(k+1) - b(k-1)x(k) = c(k)\,.$$

- Then something great happens:

THEOREM ***(Gosper, 1978)***
Suppose $a(k)$, $b(k)$, $c(k)$ are polynomials such that $\gcd(a(k), b(k+h)) = 1$ for all nonnegative integer h, and suppose that $x(k)$ is a rational function satisfying $a(k)x(k+1) - b(k-1)x(k) = c(k)$. Then $x(k)$ is a polynomial with determinable degree.

- The first important observation is that the polynomials $a(k)$, $b(k)$ and $c(k)$ are known and fixed.

- This means that the degree of $x(k)$ can take only certain values. In fact we can find the degree of $x(k)$ explicitly in terms of the degrees of $a(k)$, $b(k)$ and $c(k)$.

- For example, suppose $a(k)$ and $b(k)$ are of different degrees or have different leading coefficient. Then since $x(k+1)$ and $x(k)$ have the same leading term, the leading terms of $a(k)x(k+1)$ and $b(k-1)x(k)$ cannot cancel each other out, which means that the leading term of $c(k)$ must be of the same power. This means that

$$\begin{aligned}\deg c(k) &= \max\{\deg(a(k)x(k+1)), \deg(b(k-1)x(k))\} \\ &= \max\{\deg a(k) + \deg x(k), \deg b(k) + \deg x(k)\} \\ &= \deg x(k) + \max\{\deg a(k), \deg b(k)\}\end{aligned}$$

so that the degree of $x(k)$ is

$$\deg c - \max\{\deg a(k), \deg b(k)\}.$$

- Once we have the maximal degree d for $x(k)$, we set $x(k) = \sum_{i=0}^{d} a_i k^i$. Substituting this expression in the equation $a(k)x(k+1) - b(k-1)x(k) = c(k)$ gives a system of linear equations (the coefficients of the powers of n must be equal).

- Depending on whether this system has a solution or not, the indefinite sum
$$z_k = \frac{b(k-1)x(k)}{c(k)} t_k$$
exists or doesn't.

EXAMPLE If we run Gosper's algorithm on $t_k = k!$ (i.e. we want the indefinite sum $\sum_k k!$), we get no solutions:

$$r(k) = \frac{t_{k+1}}{t_k} = \frac{(k+1)!}{k!} = (k+1)\,,$$

so we can take $a(k) = k+1$ and $b(k) = c(k) = 1$. Then the equation $a(k)x(k+1) - b(k-1)x(k) = c(k)$ is

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

and since $a(k)$ and $b(k)$ have different degrees, the maximal degree of $x(k)$ is

$$\deg c(k) - \max\{\deg a(k), \deg b(k)\} = 0 - \max\{1,0\} = -1$$

This means that no indefinite sum that is a hypergeometric term exists in this case.

EXAMPLE However if we take $t_k = kk!$, then

$$r(k) = \frac{(k+1)(k+1)!}{kk!} = \frac{(k+1)^2}{k} = \frac{k+1}{1}\frac{k+1}{k}\,.$$

So we take $a(n) = k+1$, $b(k) = 1$ and $c(k) = k$.

In this case, the equation $a(k)x(k+1) - b(k-1)x(k) = c(k)$ is

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

which has the solution $x(k) = 1$.

This gives the indefinite sum

$$z_k = \frac{b(k-1)x(k)}{c(k)}t_k = \frac{1}{k}kk! = k!\,.$$

The tricky part

We have yet to explain why it is always possible to write the ratio $r(k) = t_{k+1}/t_k$ in the form

$$r(k) = \frac{a(k)}{b(k)} \frac{c(k+1)}{c(k)}$$

with polynomial $a(k)$, $b(k)$, $c(k)$ and $\gcd(a(k), b(k+h)) = 1$ for all nonnegative integers h.

One way that works is as follows.

- Write $r(k) = f(k)/g(k)$ for polynomials $f(k)$ and $g(k)$.
- Suppose that for some nonnegative integer h, $f(k)$ and $g(k+h)$ have a common factor $u(k) \neq 1$. This means

$$\begin{aligned} f(k) &= \bar{f}(k)u(k) \\ g(k) &= \bar{g}(k)u(k-h)\,, \end{aligned}$$

so that

$$r(k) = \frac{f(k)}{g(k)} = \frac{\bar{f}(k)}{\bar{g}(k)} \frac{u(k)}{u(k-h)}\,.$$

- But then

$$\frac{u(k)}{u(k-h)} = \frac{u(k)u(k-1)\dots u(k-h+1)}{u(k-1)\dots u(k-h+1)u(k-h)}$$

can be written in the form $\bar{c}(k+1)/\bar{c}(k)$.

- We then repeat this process with $\bar{f}$ and $\bar{g}$ and include the part coming from the common factor into the $\bar{c}$ term.
- After a finite number of steps, $\bar{f}(k)$ and $\bar{g}(k+h)$ no longer have common factors, for any h, and we call them $a(k)$ and $b(k)$ and combine the $\bar{c}(k)$ terms together to get $c(k)$.

Summary (Gosper's algorithm)

- Gosper's algorithm, when it gives a positive answer, provides a closed form formula (a hypergeometric term) for a hypergeometric sum by doing the more powerful job of computing the indefinite sum.

- Also, if Gosper's algorithm finds no answer, then no answer that is a hypergeometric term exists.

Zeilberger's algorithm

- Some functions, like e^{-x^2} have no indefinite integral expressible in terms of elementary functions, but have definite integrals over certain definite ranges:

$$\int_{-\infty}^{\infty} e^{-x^2}\, dx = \sqrt{\pi}\,.$$

- The same thing often occurs with hypergeometric sums: a hypergeometric term t_k can have no indefinite sum, but $\sum_a^b t_k$ can have a nice formula for certain values of a and b, especially if we sum over the whole range $\sum_k t_k$.

Example The partial sum of binomial coefficients

$$\sum_{k=0}^{m} \binom{n}{k}$$

does not have a nice formula for general m, while $\sum_{k=0}^{n} \binom{n}{k}$ does.

- This is why Sister Celine's algorithm is important despite appearing less powerful than Gosper's, and being much slower.

- Zeilberger's algorithm does essentially the same thing as Sister Celine's algorithm, but does it much faster.

Zeilberger's algorithm relies on the following theorem:

THEOREM ***(Wilf, Zeilberger, 1992)*** *If $F(n,k)$ is a proper hypergeometric term, then it satisfies a recurrence of the form*

$$\sum_{j=0}^{J} a_j(n)F(n+j,k) = G(n,k+1) - G(n,k)$$

in which

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

is a rational function of n and k (the certificate).

Why it's useful

- If $F(n,k)$ vanishes outside of a finite k-interval for every n, then so does $G(n,k)$ since it is a rational multiple of $F(n,k)$.
- So if we sum the recurrence over all k, the $G(n,k)$ terms on the right hand side telescope to 0.
- If we let $f(n) = \sum_k F(n,k)$, we get a recurrence on the sum:

$$\sum_{j=0}^{J} a_j(n) f(n+j) = 0\,.$$

- So we get something very similar to what Sister Celine's algorithm outputs.
- But we get it much faster.

How it works

- Zeilberger's algorithm takes advantage of Gosper's algorithm, which is very fast.
- Instead of running Gosper on $F(n,k)$, we run it on a general linear combination of the $F(n+j,k)$:

$$a_0 F(n,k) + a_1 F(n+1,k) + \ldots + a_J F(n+J,k)\,,$$

  for some J.
- On top of solving for the polynomial $x(k)$ in Gosper, we now have to solve for the $a_i(n)$ as well.
- Both systems are linear. If there is a solution, we have found the recurrence. Otherwise we increase J by 1 and start again.
- The process eventually stops because the theorem of Wilf and Zeilberger ensures that there is a such a recurrence.

Petkovšek's algorithm

- So far we saw that although Gosper's algorithm can sometimes provide a nice formula for a hypergeometric sum, as a single hypergeometric term, such a nice formula doesn't always exist.
- But it is possible to do more, by relaxing what we mean by nice formula:

Definition We say that a function $f(n)$ is of *closed form* if it is a sum of a fixed number (not depending on n) of hypergeometric terms.

- So now the question is: "When does a hypergeometric sum $f(n) = \sum_k F(n,k)$ have a closed form?"

- If $F(n, k)$ is doubly hypergeometric, we know that the algorithms of Sister Celine and Zeilberger provide a linear recurrence with polynomial coefficients satisfied by the $f(n)$.

- So the question can be reformulated as "When does a linear recurrence with polynomial coefficients have a closed form solution?"

- Petkovšek's algorithm `Hyper` can decide that question, even in the case where the equation is inhomogeneous (the inhomogeneous term must itself be of closed form, i.e. the sum of a fixed number of hypergeometric terms).

EXAMPLE Consider the recurrence

$$(n-1)y(n+2) - (n^2+3n-2)y(n+1) + 2n(n+1)y(n) = 0\,.$$

In a way similar to what happens for linear differential equations, we expect two linearly independent solutions here.

The first step of Petkovšek's algorithm finds that the two solutions can be obtained by solving two polynomial recurrences.

The algorithm gives the solutions $y(n) = 2^n$ and $y(n) = n!$, which yields the general solution

$$y(n) = A2^n + Bn!$$

for arbitrary constants A and B.

Recent developments

Recent developments in WZ theory all use the fact that the theory can be written down and explained in terms of operators.

Definition Given a function $F(n,k)$ of the (discrete) variables n and k, we define the operators Δ_n, Δ_k, N and K by

$$\begin{aligned} \Delta_n F(n,k) &= F(n+1,k) - F(n,k) \\ N\,F(n,k) &= F(n+1,k) \\ \Delta_k F(n,k) &= F(n,k+1) - F(n,k) \\ K\,F(n,k) &= F(n,k+1)\,. \end{aligned}$$

With this notation, the equation
$F(n+1,k) - F(n,k) = G(n,k+1) - G(n,k)$ becomes

$$\Delta_n F(n,k) = \Delta_k G(n,k)$$

while the Wilf-Zeilberger theorem states the existence of a polynomial $P(n,N)$ such that

$$P(n,N)F(n,k) = \Delta_k G(n,k)\,.$$

We can now describe briefly two recent trends in WZ theory research.

Parallel with differential topology (Zeilberger)

- It is possible to mimic parts of differential topology in the discrete case.
- Zeilberger defines difference forms in a way similar to differential forms, by using the difference operators in each of the variables instead of the partial derivatives.
- It is also possible to define a notion of discrete manifold, and to sum (integrate) difference forms over them.
- Some of the important theorems on the integration of differential forms can be translated into the discrete setup, like Stokes theorem and the Poincaré lemma.

- Certain difference forms, called *WZ forms*, are closely related to hypergeometric identities.
- In this language, proving a hypergeometric identity from its associated WZ form just consists of checking that the form is *closed*.
- Hypergeometric identities coming from *exact* WZ forms are trivial in a special sense.
- So the interesting hypergeometric identities are those coming from the closed but not exact WZ forms.
- This defines a notion of WZ cohomology.

Elimination theory (Chyzak)

- Recall that with the operator notation, the Wilf-Zeilberger theorem stated the existence of a nonzero polynomial $P(n, N)$ such that

$$P(n, N)F(n, k) = \Delta_k G(n, k) = (K - 1)G(n, k)\,.$$

- Recall also that $G(n, k)$ is a rational multiple of $F(n, k)$:

$$G(n, k) = R(n, k)F(n, k)\,.$$

- This means that

$$\begin{aligned} P(n,N)F(n,k) &= (K-1)\big(R(n,k)F(n,k)\big) \\ &= R(n,k+1)F(n,k+1) - R(n,k)F(n,k) \\ &= (K\cdot R(n,k))(K\cdot F(n,k)) - R(n,k)F(n,k) \\ &= ((K\cdot R)K - 1\big)\, F(n,k) \end{aligned}$$

so that

$$P(n,N) + 1 - (K\cdot R)K$$

annihilates $F(n,k)$.

- What makes this equation useful is the absence of the variable k, the summation index.

- Normally, given $F(n, k)$ which is doubly hypergeometric, we will be able to write down a system of operators in N, K, n, k that annihilate $F(n, k)$.

- It is actually possible to make this system a system of noncommutative polynomials in N, K, n and k.

- Then with a noncommutative analogue of the Buchberger algorithm for Gröbner bases, we can try to eliminate the variable k from the system.

- More is possible in the more general context or *Ore algebras*.

References

For the main points of WZ theory:

- M. Petkovšek, H. Wilf, D. Zeilberger, $A = B$. A. K. Peters, Ltd., 1996.

available in PDF form for free from

`http://www.cis.upenn.edu/~wilf/AeqB.html`

For recent developments:

- D. Zeilberger, *Closed form (pun intended!)*. Contemporary Mathematics 143 (1993), 579-608.
- F. Chyzak, B. Salvy, *Non-commutative elimination in Ore algebras proves multivariate holonomic identities*. Journal of Symbolic Computation 26 #2 (1998), 187-227.