Gosper's Algorithm Part 1
Seminar on Automatic proofs of binomial sum identities by Valentin Féray

Vanessa Hauri

March 2020

1 Introduction

This algorithm allows us to do indefinite hypergeometric sums in simple closed form, or if it cannot be done in a given case, it proves the impossibility of it. Furthermore, it is vital in the execution of the WZ algorithm (Chapter 7) and in the operation of the creative telescoping algorithm.
More precisely, we are looking at a sum

$$s_n = \sum_{k=0}^{n-1} t_k$$

where t_k is a hypergeometric term not depending on n. We then have that

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

is a rational function of k. Our goal is to express s_n in a closed form, i.e. without the summation sign.
Note that $s_{n+1} - s_n = t_n$. Thus, we want to know if given t_n there exists a hypergeometric term z_n s.t.

$$z_{n+1} - z_n = t_n \tag{1}$$

If we can find such a z_n then we will have expressed the sum in the simple form of a single hypergeometric term plus a constant.
Note that any such z_n will have the form

$$z_n = z_{n+1} + t_{n+1} = z_{n-2} + t_{n-2} + t_{n-1} = ... = z_0 + \sum_{k=0}^{n-1} t_k = s_n + c$$

where $c = z_0$ is a constant.

Remark: Gosper's algorithm boils down to the following question: Given a hypergeometric term t_n, is there a hypergeometric term z_n satisfying $z_{n+1} - z_n = t_n$? If the answer is yes, then s_n can be expressed as a hypergeometric term plus a constant and the algorithm outputs such a term. In that case t_n is called Gosper-summable. If Gosper's algorithm returns a negative answer, that proves that it has no hypergeometric solution.

Remark: In the following, all our arithmetic operations will take place in some field K of characteristic 0.

2 Hypergeometrics to rationals to polynomials

Let z_n be a hypergeometric term satisfying (1). Then we have

$$z_n = t_n \cdot \frac{1}{\frac{z_{n+1}}{z_n} - 1}$$

which is a rational function of n. Let now

$$z_n = y(n) t_n$$

where $y(n)$ is a rational function of n. Substituting this for z_n in (1) shows us that $y(n)$ satisfies

$$r(n)y(n+1) - y(n) = 1 \tag{2}$$

where $r(n) = \frac{t_{n+1}}{t_n}$. We have thus reduced the problem of finding hypergeometric solutions of (1) to finding rational solutions of (2).
To reduce the problem further to that of finding polynomial solutions, assume that we can rewrite

$$r(n) = \frac{a(n)c(n+1)}{b(n)c(n)} \tag{3}$$

where $a(n), b(n), c(n)$ are polynomials in n and it holds that

$$gcd(a(n), b(n+h)) = 1 \tag{4}$$

for all nonnegative integers h.

Remark: We will see later that a factorization of this type exists for every rational function, for now let's just assume it is true.

We are looking for a nonzero rational solution of (2) in the form

$$y(n) = \frac{b(n-1)x(n)}{c(n)} \tag{5}$$

where $x(n)$ is an unknown rational function of n. As we substitute (3) and (5) into (2) we see that $x(n)$ satisfies

$$a(n)x(n+1) - b(n-1)x(n) = c(n) \tag{6}$$

Theorem: Let $a(n), b(n), c(n)$ be polynomials satisfying $gcd(a(n), b(n+h)) = 1$ for all nonnegative integers h. If $x(n)$ is a rational function of n satisfying (6), then $x(n)$ is a polynomial in n.

Proof. Outline of the proof: We will prove this by contradiction. Let $x(n) = f(n)/g(n)$ where $f(n)$ and $g(n)$ are relatively prime polynomials in n. Then we can rewrite (6) as

$$a(n)f(n+1)g(n) - b(n-1)f(n)g(n+1) = c(n)g(n)g(n+1)$$

Let us now suppose that the conclusion of the theroem is false. If $x(n)$ is not a polynomial, this means that $g(n)$ is a non-constant polynomial. Let now N be the largest nonnegative integer st. $gcd(g(n), g(n+N))$ is a non-constant polynomial and let $u(n)$ be a non-constant irreducible common divisor of $g(n)$ and $g(n+N)$. We can then show that $u(n+1)|b(n+N)$ and that $u(n+1)|a(n)$. However, this would mean that $u(n+1)$ is a non-constant factor of both $a(n)$ and $b(n+N)$ which contradicts (4). Hence, $g(n)$ has to be constant, and so $x(n)$ is a polynomial in n. □

Therefore we have reduced the problem of finding hypergeometric solutions of (1) to finding polynomial solutions of (6). If $x(n)$ is a nonzero polynomial solution of (6), then

$$z_n = \frac{b(n-1)x(n)}{c(n)} t_n$$

is a hypergeometric solution of (1) and vice versa.

Remark: Let us quickly take a look at the general outline of the Gosper's algorithm that we have learned so far.

Gosper's Algorithm Outline

INPUT: A hypergeometric term t_n
OUTPUT: A hypergeometric term z_n satisfying (1) if one exists; $\sum_{k=0}^{n-1} t_k$ otherwise.
1. Form the ration $r(n) = t_{n+1}/t_n$ which is a rational function of n.
2. Write $r(n) = \frac{a(n)c(n+1)}{b(n)c(n)}$ where $a(n), b(n), c(n)$ are polynomials satisfying (4).
3. Find a nonzero polynomial solution $x(n)$ of (6) if one exists; otherwise return $\sum_{k=0}^{n-1} t_k$ and stop.
4. Return $\frac{b(n-1)x(n)}{c(n)} t_n$ and stop.

Remark: The sum we are looking for is $s_n = z_n - z_{k_0}$ where k_0 is the lower summation bound.

Example: Let

$$S_m = \sum_{k=0}^{m} k^2 2^k$$

Can this sum be expressed in closed form? Let's work through the algorithm together. We see immediately that $t_n = n^2 2^n$. Thus we have that

$$r(n) = \frac{(n+1)^2 2^{n+1}}{n^2 2^n} = \frac{2(n+1)^2}{n^2}$$

Here, the choices for $a(n), b(n), c(n)$ are obvious, namely $a(n) = 2, b(n) = 1, c(n) = n^2$. It is easy to see that this choice satisfies (3) and (4). Equation (6) thus becomes

$$2x(n+1) - x(n) = n^2$$

The polynomial x(n) satisfying this equation is not easy to be found. We will cover how to do this next week in the chapter on Step 3. For now, let's just assume we have already found the correct $x(n)$: $x(n) = n(n-4) + 6$. One can easily check that this $x(n)$ satisfies our equation above. Hence,

$$z_n = \frac{1 \cdot x(n) \cdot n^2 2^n}{n^2} = 2^n(n(n-4) + 6)$$

which satisfies $z_{n+1} - z_n = t_n$. Finally, $s_m = z_m - z_0 = 2^m(m^2 - 4m + 6) - 6$, so the closed form we are looking for is

$$S_m = s_{m+1} = 2^{m+1}(m^2 - 2m + 3) - 6$$

3 The full algorithm: Step 2

We will now take a closer look at how to obtain the factorization (3) of a given rational function $r(n)$. Let $r(n) = f(n)/g(n)$ where $f(n)$ and $g(n)$ are relatively prime polynomials. If $gcd(f(n), g(n+h)) = 1$ then we can take $a(n) = f(n), b(n) = g(n), c(n) = 1$

and we have the desired factorization. Otherwise let $u(n)$ be a non-constant common factor of $f(n)$ and $g(n+h)$ for some nonnegative integer h. Let $f(n) = \bar{f}(n)u(n)$ and $g(n) = \bar{g}(n)u(n-h)$. Then

$$r(n) = \frac{f(n)}{g(n)} = \frac{\bar{f}(n)u(n)}{\bar{g}(n)u(n-h)}$$

Defintion: Given a polynomial $p(x) = a_n x^n + ... + a_0$ of degree n with roots a_i for $i \in [1,n]$ and a polynomial $q(x) = b_m x^m + ... + b_0$ of degree m with roots b_j for $j \in [1,m]$ the resultant is defined by

$$R(p,q) = a_n^m b_m^n \prod_{i=1}^{n} \prod_{j=1}^{m} (\alpha_i - \beta_j)$$

Alternative Definition: Note that a more useful method when working with computers is defining the resultant as the determinant of the Sylvester Matrix of two polynomials. Let p(x) and q(x) be as above. The Sylvester Matrix is a square matrix of dimension $n+m$, formed by filling the matrix (beginning with the upper left corner) with the coefficients of p(x), then shifting down one row and one column to the right and filling in the coefficients starting there until they hit the right side. The process is then repeated for the coefficients of q(x).

$$\begin{bmatrix} \mathrm{a}_0 & \mathrm{a}_1 & \mathrm{a}_2 & ... & \mathrm{a}_n & 0 & ... & 0 \\ 0 & \mathrm{a}_0 & \mathrm{a}_1 & ... & \mathrm{a}_{n-1} & \mathrm{a}_n & ... & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & \mathrm{a}_0 & ... & ... & \mathrm{a}_{n-1} & \mathrm{a}_n \\ \mathrm{b}_0 & \mathrm{b}_1 & \mathrm{b}_2 & ... & \mathrm{b}_m & 0 & ... & 0 \\ 0 & \mathrm{b}_0 & \mathrm{b}_1 & ... & \mathrm{b}_{m-1} & \mathrm{b}_m & ... & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & \mathrm{b}_0 & ... & ... & \mathrm{b}_{m-1} & \mathrm{b}_m \end{bmatrix}$$

How do we know when (4) is satisfied or how do we find the values of h that violate it otherwise? Let $R(h)$ denote the resultant of $f(n)$ and $g(n+h)$. Then $R(h)$ is a polynomial in h with the property that $R(\alpha) = 0$ if and only if $gcd(f(n), g(n+\alpha))$ is not a constant polynomial. Therefore the values of h that violate (4) are precisely the nonnegative integer zeros of $R(h)$.

Remark: Let us now take a closer look at step 2 of Gosper's Algorithm.

Gosper's Algorithm Step 2

2.1. Let $r(n) = Z\frac{f(n)}{g(n)}$ where f, g are monic relatively prime polynomials and Z is a constant.

$R(h) := Resultant_n(f(n), g(n+h))$;

Let $S = \{h_1, h_2, ..., h_N\}$ be the set of nonnegative integer zeros of $R(h)$. ($N \geq 0, 0 \leq h_1 < h_2 < ... < h_N$).

2.2. $p_0(n) := f(n); q_0(n) := g(n)$; for $j = 1, 2, ..., N$ do

$s_j(n) := gcd(p_{j-1}(n), q_{j-1}(n+h_j))$;

$p_j(n) := p_{j-1}(n)/s_j(n)$;

$q_j(n) := q_{j-1}(n)/s_j(n-h_j)$.

$a(n) := Zp_N(n)$;

$b(n) := q_N(n)$;

$c(n) := \prod_{i=1}^{N}\prod_{j=1}^{h_i} s_i(n-j)$.

Example: Let us take the same example as before, where $r(n) = \frac{2(n+1)^2}{n^2}$. We then take $Z = 2, f(n) = (n+1)^2, g(n) = n^2$ and we see that $f(n)$ and $g(n)$ are relatively prime polynomials. Then

$$R(h) = Resultant_n(f(n), g(n+h)) = (h-1)^4$$

Clearly, the only nonnegative integer zero is $h = 1$. Hence, following the algorithm step by step we get that $s_1(n) = (n+1)^2, p_1(n) = 1, q_i(n) = 1$ which gives us $a(n) = 2, b(n) = 1, c(n) = n^2$, and this is, as we have seen before, exactly what we wanted.

Remark: We can compute directly that the three polynomials produced by this algorithm satisfy condition (3). To show that they also satisfy condition (4), we can note that by definition of p_j, q_j, and s_j,

$$gcd(p_k(n), q_k(n+h_k)) = gcd(\frac{p_{k-1}(n)}{s_k(n)}, \frac{q_{k-1}(n+h_k)}{s_k(n)}) = 1$$

for all k s.t. $1 \leq k \leq N$.