Cássio Jandir Pagnoncelli

Somas hipergeométricas definidas e indefinidas

Curitiba

2013

Cássio Jandir Pagnoncelli

Somas hipergeométricas definidas e indefinidas

Monografia apresentada junto ao curso de Bacharelado em Ciência da Computação do Departamento de Informática, no setor de Ciências Exatas, como requisito parcial para a obtenção do título de Bacharel.

Universidade Federal do Paraná – UFPR

Departamento de Informática

Bacharelado em Ciência da Computação

Orientador: Dr. Renato José da Silva Carmo

Curitiba

2013

Cássio Jandir Pagnoncelli

Somas hipergeométricas definidas e indefinidas

Monografia apresentada junto ao curso de Bacharelado em Ciência da Computação do Departamento de Informática, no setor de Ciências Exatas, como requisito parcial para a obtenção do título de Bacharel.

Trabalho aprovado. Curitiba, 15 de julho de 2013:

Dr. Renato José da Silva Carmo
Orientador

Dr. André Luiz Pires Guedes
Convidado 1

Dr. Jaime Cohen
Convidado 2

Curitiba
2013

Resumo

Apresentamos os problemas das somas hipergeométricas definida e indefinida e provas de algumas identidades hipergeométricas e suas respectivas soluções algorítmicas bem como alguns exemplos e implementações.

Palavras-chaves: celine. gosper. zeilberger. wilf. wz. soma hipergeométrica definida. soma hipergeométrica indefinida. identidades combinatoriais. identidades hipergeométricas.

Abstract

Here we present the indefinide hypergeometric sum and definite hypergeometric sum problems along with some hypergeometric identities proofs and its algorithmic solutions as well as examples and implementations.

Key-words: celine. gosper. zeilberger. wilf. wz. definite hypergeometric sum. indefinite hypergeometric sum. combinatorial identities. hypergeometric identities.

Sumário

1 Prefácio

Suponha que nossa carreira dependesse de "resolver" somas como

$$\sum_{k=-\infty}^{\infty} \binom{n}{k}, \qquad \sum_{k=-\infty}^{\infty} \binom{n}{k}^2, \qquad \sum_{k=0}^{n} x^k, \qquad \sum_{k=a}^{b-1} k \cdot k!$$

e, com uma certa frequência, deparásse-mo-nos com algumas razoavelmente mais intimidadoras, como

$$\sum_{k=-\infty}^{\infty} \binom{x+1}{2k+1} \binom{x-2k}{n-k} 2^{2k+1} \quad \text{e} \quad \sum_{k=-\infty}^{\infty} (-1)^k \binom{2n}{k} \binom{2k}{k} \binom{4n-2k}{2n-k}.$$

Primeira soma. Exercitando um pouco a memória, o resultado é 2^n e uma prova por indução faz o resto do trabalho.

Segunda soma. Existem $\binom{n}{k}$ maneiras de escolher k elementos em $\{1, 2, \cdots, n\}$ e $\binom{n}{n-k}$ maneiras de escolher $n-k$ elementos de $\{n+1, n+2, \cdots, 2n\}$. Então, existem $\binom{n}{k}\binom{n}{n-k} = \binom{n}{k}^2$ maneiras de formar tal par de escolhas. Isto é equivalente a dizer que existem $\binom{2n}{n}$ maneiras de escolher n elementos de $\{1, 2, \cdots, 2n\}$. Portanto, a referida soma também pode ser escrita como $\binom{2n}{n}$.

Terceira soma. Como veremos com mais detalhes,

$$\sum_{k=0}^{n} x^k - \sum_{k=0}^{n} x^{k+1} = 1 - x^{n+1} \quad \text{se e somente se} \quad \sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}.$$

Quarta soma. Após peregrinar em busca de uma solução por inspeção, o resultado revela-se como sendo $b! - a!$.

Por mais engenhosas que sejam estas soluções, não podemos depender de truques e sutilezas assim para resolver somas desta natureza. Na mesma linha de raciocínio, imagine-nos dependendo da criatividade para resolver problemas do cálculo infinitesimal.

Para $\int x \,\mathrm{d}x$ é fácil: tome a metade de um quadrado de lado x e acrescente, é claro, uma constante. Para $\int_a^b e^x \,\mathrm{d}x$, também não é difícil provar que a função que perfaz a mesma linha de sua integral é e^x e, logo, o resultado é $e^b - e^a$. Que tal, porém, resolver $\displaystyle\int \sqrt{1 - \operatorname{sen}^2(x)}\,\mathrm{d}x$? A solução envolve perceber que $\int \cos(x) = \operatorname{sen}(x) + O(1)$ e, ainda, usar o teorema de Pitágoras; é mais tênue do que os exemplos anteriores.

Para as duas somas que deixamos sem solução, talvez nos falte criatividade para as resolver. Entretanto, devemos salientar que a falta de criatividade definitivamente não impede-nos de resolver estas somas. Inclusive, não apresentamos somas mais abundantes de coeficientes binomiais e exponenciais por questões meramente estéticas.

As somas que apresentamos são exemplos de somas de termos hipergeométricos em uma e duas varíaveis, dos quais discutiremos no capítulo seguinte. Estas somas, basicamente, podem ser divididas em duas categorias de acordo com seus limites: (i) aquelas cujos somandos são sobre todos os inteiros; e (ii) aquelas cujos somandos são sobre intervalos finitos. Chamaremos de soma *definida* uma soma do tipo (i) e de soma *indefinida* uma soma do tipo (ii). Não trataremos somas onde um e apenas um dos limites é finito.

Felizmente, existem algoritmos para resolver somas, tanto definidas quanto indefinidas, para certos tipos de somandos. Encontrar "formas fechadas" para estas somas são de particular interesse de combinatória, análise de algoritmos, estatística discreta, processamento de sinais e outras áreas. Nosso objetivo é apresentar estes algoritmos e suas implementações.

1.1 Uma soma definida

Voltando na segunda soma que apresentamos, no início do capítulo, observe que seu somando $F(n,k) = \binom{n}{k}^2$ satisfaz a recorrência

$$n\binom{n}{k}^2-(2n-1)\left(\binom{n-1}{k}^2+\binom{n-1}{k-1}^2\right)+(n-1)\left(\binom{n-2}{k}^2-2\binom{n-2}{k-1}^2+\binom{n-2}{k-2}^2\right) \quad = \quad 0.$$

Esta recorrência é obtida pelo algoritmo $\mathrm{celineIJ}$, que apresentaremos no capítulo 4. Somando esta recorrência sobre todos os k inteiros, temos que a soma, $f(n)$, resulta

$$f(n) \;=\; \frac{2(2n-1)}{n}\, f(n-1) \;=\; \cdots \;=\; \frac{(2n)!}{n!^2} \;=\; \binom{2n}{n},$$

que é uma forma alternativa de escrever a soma em questão.

1.2 Uma soma indefinida

No início do capítulo, apresentamos o exemplo $\sum_{k=a}^{b-1} F(k)$, onde $F(k) = k \cdot k!$, e afirmamos que esta soma resulta $b! - a!$.

Neste exemplo, buscamos uma "anti-derivada discreta". Para uma primeira aproximação do assunto, seria o análogo discreto da anti-derivada.

No cálculo infinitesimal, $x^3/3$ é a anti-derivada de x^2 em relação a x,

$$\int_a^b x^2\,\mathrm{d}x \;=\; \frac{b^3}{3} - \frac{a^3}{3}.$$

Aqui, temos $z(k) = k!$ fazendo o mesmo papel da anti-derivada,

$$\sum_{k=a}^{b-1} F(k) = z(b) - z(a) = b! - a!.$$

Este $z(k)$ é obtido via algoritmo de Gosper, que será discutido no capítulo 5.

1.3 Prova de identidades com WZ

Suponha que a penúltima soma não resolvida admite uma "forma fechada",

$$\sum_{k=-\infty}^{\infty} \binom{x+1}{2k+1}\binom{x-2k}{n-k} 2^{2k+1} = \binom{2x+2}{2n+1}.$$

Podemos decidir esta proposição com bastante destreza.

1. Forme
$$F(n,k) \stackrel{\text{def.}}{=} \frac{\binom{x+1}{2k+1}\binom{x-2k}{n-k}2^{2k+1}}{\binom{2x+2}{2n+1}}$$
para provar $\sum_{k=-\infty}^{\infty} F(n,k) = 1$.

2. Observe que
$$F(n+1,k) - F(n,k) = R(n,k+1)F(n,k+1) - R(n,k)F(n,k), \tag{1.3.1}$$
onde
$$R(n,k) = \frac{k(2k+1)(x-2n-1)}{(k-n-1)(2n-2x-1)(n-x)},$$

3. Some (1.3.1) sobre todos os k inteiros verificando que o termo da direita é telescópico resultando 0. Então,
$$\sum_{k=-\infty}^{\infty} F(n+1,k) = \sum_{k=-\infty}^{\infty} F(n,k)$$
e, logo, $\sum_{k=-\infty}^{\infty} F(n,k)$ é independente de n. Noutras palavras, $\sum_{k=-\infty}^{\infty} F(n,k)$ é uma constante.

4. A identidade vale se e somente se esta constante é 1. Ou seja, decida por "sim" se e somente se
$$\sum_{k=-\infty}^{\infty} F(0,k) = 1.$$

1.4 Interação com MAXIMA

Ao longo do texto, vamos recorrer ao MAXIMA[1], que, especialmente, porta a biblioteca zeilberger, escrita por Fabrizio Caruso. No nosso texto, será corriqueira a interação com o MAXIMA.

Cada interação é delimitada por um colchete externo contendo dois colchetes internos: o primeiro deles contém o comando e o segundo, a resposta. Este é o ambiente de interação:

```
(%i1) := /* wxmaxima e xmaxima são interfaces para Maxima. */
(%i2) := /* expansão de Taylor de 1/(1-x) + e^x em torno de x = 0 */
(%i3) := taylor(1/(1-x) + exp(x), x, 0, 8);
```

$$/T/\quad 2+2x+\frac{3x^2}{2}+\frac{7x^3}{6}+\frac{25x^4}{24}+\frac{121x^5}{120}+\frac{721x^6}{720}+\frac{5041x^7}{5040}+\frac{40321x^8}{40320}+\cdots$$

Nosso pacote, que, de certo modo, complementa aquele escrito por Caruso, está disponível em

http://www.inf.ufpr.br/cjp07/permalink/hypergeo.tar.bz2

1.5 Roteiro

O capítulo 2 trata de funções geradoras. Funções geradoras prestam-se, dentre outras utilidades, a resolver recorrências. Relações de recorrências estão presentes ao longo de todo o texto, em todos os capítulos.

No capítulo 3, apresentamos o escopo de funções onde trabalharemos. Este escopo é a classe hipergeométrica. Na classe hipergeométrica estão presentes, por exemplo, os termos hipergeométricos, que são amplamente mencionados nos capítulos seguintes. Definiremos a noção de “forma fechada”, intencionalmente ainda nebulosa.

1. Capítulo 4. Dadas as definições e ferramentas nos capítulos anteriores, este é o nosso ponto de partida.

   Apresentaremos o problema da soma hipergeométrica definida. O algoritmo desenvolvido pela irmã Mary Celine Fasenmyer, na sua tese de PhD (FASENMYER, 1947), resolve este problema ao encontrar recorrências satisfeitas por certos tipos de funções hipergeométricas.

2. Capítulo 5. Gosper, um dos pesquisadores do projeto Macsyma, escreveu um algoritmo (GOSPER, 1978) para resolver somas hipergeométricas indefinidas.

[1] Disponível em http://maxima.sourceforge.net/

Essencialmente, resolver uma soma hipergeométrica indefinida envolve obter uma "anti-derivada discreta", a qual formalizaremos posteriormente.

Este algoritmo tem como dependência a obtenção da $\mathrm{resultante}$ de dois polinômios. Prestaremos uma descrição da $\mathrm{resultante}$ no apêndice B.

3. Capítulo 6. Herbert Wilf e Doron Zeilberger criaram o algoritmo WZ, que, junto com os trabalhos de Celine, Gosper e Petkovšek, unificaram a teoria das funções hipergeométricas no caso univariado; por este trabalho, Wilf e Zeilberger ganharam o prêmio Steele de 1998.

   Aqui, ainda apresentaremos a noção de pares Wilf-Zeilberger e certificados para funções racionais.

4. Capítulo 7. Similar ao algoritmo da irmã Celine, o algoritmo de Zeilberger também resolve o problema da soma hipergeométrica definida, mas é mais eficiente em relação ao tempo de computação.

5. Capítulo 8. Por fim, ministramos algumas considerações finais sobre estes algoritmos e uma bibliografia, talvez viesada, para alguns trabalhos baseados nesta teoria.

1.6 Considerações iniciais

Apesar destes algoritmos resolverem problemas complexos de combinatória, eles não descartam os belos argumentos vistos em problemas de contagem; argumentos estes que vêm, ainda que esporadicamente, acompanhados de uma profunda compreensão do problema em questão.

Originalmente, os algoritmos que apresentaremos foram implementados para MAPLE e MATHEMATICA por Zeilberger (University of New Jersey), Wilf (University of Pennsylvania), Peter Paule e Markus Schörn (Johannes Kepler Universität), que são soluções fechadas. Alguns anos mais tarde, Fabrizio Caruso (Johannes Kepler Universität) escreveu um pacote para MAXIMA que implementa os dois principais algoritmos, gosper e $\mathrm{zeilberger}$.

Notação adotada

$a \equiv b \pmod{d}$	$\mathrm{mdc}(a, b \bmod d) = 1$
$a^{\overline{k}}$	$a\,(a+1)\,(a+2)\cdots(a+k-1)$
$a^{\underline{k}}$	$a\,(a-1)\,(a-2)\cdots(a-k+1)$
$\Gamma(x)$	função gama, $\int_0^\infty t^{x-1}e^{-t}\mathrm{d}t$. Em especial, $\Gamma(n+1) = n!$ se $n \in \mathbb{N}$.
$[t(x, n)]\,G(x)$	coeficiente de $t(x,n)$ na expansão de $G(x)$ como uma série sobre n
$\mathrm{grau}\,(P)$	é o maior expoente dentre os monômios de P
$\mathrm{numerador}\,(E)$	numerador da expressão E
$\mathrm{denominador}\,(E)$	denominador da expressão E
$\mathrm{cmm}\,(P)$	coeficiente do maior monômio de P
$\sum_k t(k)$	soma de $t(k)$ sobre todos os k inteiros

2 Funções Geradoras

Funções geradoras (FLAJOLET; SEDGEWICK, 2009) prestam-se para uma variedade de aplicações ao longo da matemática pura e alguns ramos da matemática aplicada, como teoria da probabilidade, combinatória, processamento de sinais e análise de Fourier (WILF, 2005). Dentre elas, temos a resolução de recorrências. Relações de recorrência estão presentes em muitos trechos desta obra e, em capítulos seguintes, suporemos uma rasa familiaridade do leitor no seu manuseio.

Partiremos de uma recorrência para derivar a função geradora de série de potências e, com ela, encontraremos uma forma fechada para a sequência de Fibonacci. Além da série de potências, também apresentaremos a série exponencial via expansão de Taylor e, por fim, uma ferramenta computacional para resolução de recorrências.

Devido à natureza dos problemas e soluções e à ampla presença de recorrências que apresentaremos nos capítulos a seguir, não podemos negligenciar a necessidade de uma compreensão, ainda que pouco profunda, de resolução de algumas recorrências, o que justifica a existência deste capítulo ao invés de um curto manual para a supracitada ferramenta computacional.

2.1 A série geométrica

A recorrência

$$g_n = x \cdot g_{n-1} + 1 \qquad (n \geqslant 1;\, g_0 = 1)$$

é a origem de uma importante série no escopo de funções geradoras.

Notação (Sequência)**.** Denotaremos a sequência $(g_0, g_1, g_2, \cdots)$ por $\{g_n\}_{n=0}^{\infty}$.

Se resolvermos esta recorrência, obteremos o n-ésimo termo da sequência $\{g_n\}_{n=0}^{\infty}$. Pelo método indutivo, temos

$$g_n = \sum_{k=0}^{n} x^k, \quad \text{para todo } n \text{ natural}.$$

Se fizermos a operação $g_n - x g_n$, obteremos

$$\begin{aligned} \underbrace{g_n - xg_n}_{(1-x)g_n} &= (1+x+x^2+\cdots+x^n) - (x+x^2+x^3+\cdots+x^{n+1}) \\ &= 1 - x^{n+1}, \end{aligned}$$

que nos conta que

$$g_n = \frac{1-x^{n+1}}{1-x}.$$

(Na literatura, g_n é chamada de soma geométrica porque a razão dos termos consecutivos da sequência é sempre uma constante.) Isso fica muito mais interessante quando olhamos para esta soma no infinito. Em especial, $\lim_{n\to\infty} x^{n+1} = 0$ quando $|x| < 1$ e, daí,

$$\begin{aligned}\lim_{n\to\infty} \frac{1-x^{n+1}}{1-x} &= \lim_{n\to\infty} \frac{1}{1-x} - \underbrace{\lim_{n\to\infty} \frac{x^{n+1}}{1-x}}_{=0} \\ &= \frac{1}{1-x};\end{aligned}$$

isto é,

$$\lim_{n\to\infty} \sum_{j=0}^{n} x^j = \sum_{n\geqslant 0} x^n = \frac{1}{1-x}, \tag{2.1.1}$$

que chamamos de série geométrica.

Para localizar-nos no campo das funções geradoras, componha rx com $1/(1-x)$ e o resultado com αx para obtermos a série onde os coeficientes formam a bem conhecida progressão geométrica de razão r e primeiro termo α.

Na mesma linha de raciocínio, dizemos que $\{a_n\}_{n=0}^{\infty}$ é a sequência gerada pela série de potências $\sum_{n\in\mathbb{N}} a_n x^n$. Usando a equação (2.1.1), se conseguirmos escrever a referida soma em termos de $1/(1-x)$, teremos uma descrição finita de uma sequência infinita.

Definição. À expressão equivalente a $\sum_{n\in\mathbb{N}} a_n x^n$ em termos de $1/(1-x)$ damos o nome de *função geradora ordinária de série de potências (“FGOSP”)* da sequência $\{a_n\}_{n=0}^{\infty}$.

Notação. $[t(x,n)]\, s(x)$ denota o coeficiente de $t(x,n)$ na expansão em série de $s_n(x)$ sobre a variável n.

Exemplo.

$$\begin{aligned}[x^n]\,\frac{1}{1-x} &= 1 \\ [x^n]\left(\frac{2}{1-2x} - \frac{1}{1-x}\right) &= 2^{n+1} - 1.\end{aligned}$$

A expansão de uma *FGOSP* em somas de potências gera, intuitivamente, uma sequência: o coeficiente de x^n é o n-ésimo termo da sequência. Mais precisamente, o n-ésimo termo da sequência $\{a_n\}_{n=0}^{\infty}$ cuja *FGOSP* é $G(x)$ é o coeficiente de x^n na expansão de $G(x)$ como uma série de potências; i.e.,

$$a_n = [x^n]\, G(x).$$

2.2 Expansão de Taylor

A série de Taylor é a expansão em somas de potências de uma função $f(x)$ de imagem real em torno do ponto $x = a$, dada por

$$\begin{aligned} f(x) &= \lim_{n\to\infty} \sum_{k=0}^{n} \frac{f^{(k)}(a)}{k!} (x-a)^k \\ &= \sum_{n\geqslant 0} \frac{f^{(n)}(a)}{n!} (x-a)^n. \end{aligned}$$

É da série de Taylor que ganhamos ou verificamos várias identidades, como

$$\begin{aligned} \cos x &= \sum_{\substack{n \text{ natural} \\ n \text{ par}}} (-1)^{n/2} \frac{x^n}{n!} \\ \sin x &= \sum_{\substack{n \text{ natural} \\ n \text{ ímpar}}} (-1)^{\frac{n-1}{2}} \frac{x^n}{n!} \\ e^x &= \sum_{n\geqslant 0} \frac{x^n}{n!} \\ \ln(1+x) &= \sum_{n\geqslant 1} \frac{(-1)^{n+1}}{n} x^n \\ \ln\left(\frac{1+x}{1-x}\right) &= \sum_{\substack{n \text{ natural} \\ n \text{ ímpar}}} 2\frac{x^n}{n} \end{aligned}$$

No caso particular de avaliarmos uma função em torno do ponto $a = 0$, dizemos que a expansão é uma série de Maclaurin; i.e.,

$$\begin{aligned} f(x) &= \sum_{n\geqslant 0} \frac{f^{(n)}(0)}{n!} x^n \\ &= f(0) + f'(0)x + \frac{f''(0)}{2} x^2 + \frac{f^{(3)}(0)}{3!} x^3 + \cdots. \end{aligned}$$

2.3 A série exponencial

Expandindo e^x pela série de Mclaurin (i.e., pela série de Taylor em torno do ponto $a=0$), temos

$$\begin{aligned}
e^x &= \sum_{n\geqslant 0} \frac{f^{(n)}(0)}{n!}x^n \\
&= \sum_{n\geqslant 0} \frac{x^n}{n!}\left(\left.\underbrace{\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\mathrm{d}}{\mathrm{d}x}\left(\cdots\left(\frac{\mathrm{d}}{\mathrm{d}x}(e^x)\right)\cdots\right)\right)}_{n \text{ vezes}}\right|_{x=0}\right) \\
&= \sum_{n\geqslant 0} \frac{x^n}{n!}\left(e^x|_{x=0}\right) \\
&= \sum_{n\geqslant 0} \frac{x^n}{n!} \\
&= 1+x+\frac{x^2}{2}+\frac{x^3}{6}+\cdots,
\end{aligned}$$

chamada de série exponencial. As derivações pontuais, na igualdades acima, correspondem à notação de Leibniz.

Definição. Nos mesmos moldes da série geométrica, dizemos que a série exponencial $\sum_{n\geqslant 0} a_n\,(x^n/n!)$ gera a sequência $\{a_n\}_{n=0}^{\infty}$. Esta série, escrita em termos de e^x, é dita a *função geradora exponencial (“FGE”)* desta sequência.

Para conter-nos dentro dos limites planejados, omitiremos um exemplo. Exemplos bem elaborados sobre FGEs podem ser encontrados em (WILF, 2005) e (FLAJOLET; SEDGEWICK, 2009).

2.4 Resolução de recorrências

Uma das aplicações de funções geradoras é na resolução de recorrências, isto é, na obtenção de uma função equivalente e não-recorrente à recorrência. Para ampliar o nosso arsenal de funções geradoras conhecidas para além da *FGOSP* e *FGE*, temos

$$\begin{aligned}
\arctan x &= \sum_{k\geqslant 0}(-1)^k \frac{x^{2k+1}}{2k+1}\\
\log\frac{1}{1-x} &= \sum_{k\geqslant 1}\frac{x^k}{k}\\
(1+x)^n &= \sum_{k\geqslant 0}\binom{n}{k}x^k\\
\frac{1}{(1-x)^n} &= \sum_{k\geqslant 0}\binom{n+k-1}{k}x^k.
\end{aligned}$$

Aqui, vamos codificar a sequência de Fibonacci em uma *FGOSP* a fim de encontrar uma forma fechada (i.e., não-recorrente) para esta sequência.

Exemplo (Fibonacci via funções geradoras). Considere que desejamos encontrar uma forma fechada para a sequência de Fibonacci, $f\colon \mathbb{N}\to\mathbb{N}$, dada por

$$f(n) = \begin{cases} n, & \text{se } n \leqslant 1,\\ f(n-1)+f(n-2), & \text{se } n\geqslant 2,\end{cases}$$

onde o $(n+1)$-ésimo termo desta sequência é dado por $f(n+1) = f(n) + f(n-1)$, sempre que $n\geqslant 1$.

Antes de tudo, vamos definir a *FGOSP* $G_f(x)$ sobre o domínio de f, $G_f(x) \overset{\text{def.}}{=} \sum_{n\geqslant 0} f(n)x^n$. Assim, vamos buscar uma forma não recorrente de $f(n) = [x^n]\,G_f(x)$.

Olhando para o caso geral da recorrência,

$$f(n+1) = f(n)+f(n-1), \qquad (n\geqslant 1)$$

se multiplicarmos ambos os termos desta equação por x^n e somarmos sobre os casos em que a recorrência é satisfeita ($n\geqslant 1$), obteremos

$$\sum_{n\geqslant 1} f(n+1)x^n = \sum_{n\geqslant 1}(f(n)+f(n-1))x^n. \tag{2.4.1}$$

Para deduzirmos $[x^n]\,G_f(x)$, precisamos determinar $G_f(x)$ em forma não recorrente. É intuitivo, então, que devemos reescrever a equação acima em termos de $G(x)$.

Desenvolvendo o primeiro termo da equação supracitada temos

$$\begin{aligned}
\sum_{n\geqslant 1} f(n+1)x^n &= f(2)x+f(3)x^2+f(4)x^3+\cdots\\
&= \frac{G_f(x)-x}{x},
\end{aligned}$$

enquanto do segundo termo de (2.4.1) temos

$$\begin{aligned}\sum_{n\geqslant 1}(f(n)+f(n-1))\,x^n &= \sum_{n\geqslant 1} f(n)\,x^n + \sum_{n\geqslant 1} f(n-1)\,x^n \\ &= G_f(x) + x\,G_f(x) \\ &= (1+x)\,G_f(x),\end{aligned}$$

o que nos fornece

$$(1+x)\,G_f(x) \;=\; \frac{G_f(x)-x}{x}.$$

Isolando $G_f(x)$, vamos obter

$$\begin{aligned}G_f(x) &= \frac{-x}{x^2+x-1} \\ &= \frac{x}{1-x-x^2} \\ &= \frac{x}{(1-\frac{1+\sqrt{5}}{2}x)(1-\frac{1-\sqrt{5}}{2}x)} \\ &= \frac{1}{\frac{1+\sqrt{5}}{2}-\frac{1-\sqrt{5}}{2}}\left(\frac{1}{1-\frac{1+\sqrt{5}}{2}x}-\frac{1}{1-\frac{1-\sqrt{5}}{2}x}\right) \\ &= \frac{\sqrt{5}}{5}\left(\sum_{n\geqslant 0}\left(\frac{1+\sqrt{5}}{2}\right)^n x^n - \sum_{n\geqslant 0}\left(\frac{1-\sqrt{5}}{2}\right)^n x^n\right) \\ &= \sum_{n\geqslant 0}\underbrace{\frac{\sqrt{5}}{5}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n-\left(\frac{1-\sqrt{5}}{2}\right)^n\right)}_{f(n)} x^n\end{aligned}$$

e, para todo n natural,

$$f(n) \;=\; [x^n]\,G_f(x) \;=\; \frac{\sqrt{5}}{5}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n-\left(\frac{1-\sqrt{5}}{2}\right)^n\right).$$

Resumindo, codificamos $f(n)$ na *FGOSP* $G(x) = \sum_{n\geqslant 0} f(n)\,x^n$ para obter uma identidade via definição de $f(n)$,

$$(1+x)\,G(x) \;=\; (G(x)-x)/x,$$

e a resolvemos para $G(x)$. A forma fechada de $f(n)$ está contida em $G(x)$ e pode ser recuperada por

$$f(n) \;=\; [x^n]\;G(x).$$

Ainda neste exemplo, se trocássemos a *FGOSP* $G_f(x)$ pela *FGE*, teríamos que isolar $G_f(x)$ em

$$\frac{G_f(x)-1-x}{x^2} \;=\; \frac{G_f(x)-1}{x}+G_f(x)+e^x$$

para obter o coeficiente de $x^n/n!$ na expansão de $G_f(x)$; i.e., a função geradora escolhida determina em qual equação $G_f(x)$ deve ser isolada.

A tarefa de escolher qual função geradora usar é uma sutileza. Dependendo desta escolha, podemos deparar-nos a uma equação cuja solução para $G_f(x)$ seja de dificuldade computacional intrinsicamente mais difícil do que usando alguma outra função geradora. No exemplo da recorrência de Fibonacci, usando a FGE, precisamos resolver uma equação diferencial, enquanto usando a FGOSP, o trabalho resume-se a alguns poucos passos simples de manipulação simbólica.

2.5 Recorrências em MAXIMA

Ao longo do texto, recorreremos à alguns comandos do MAXIMA. A documentação para estes comandos pode ser consultada em (VILLATE, 2013).

MAXIMA disponibiliza a função solve_rec no pacote homônimo para resolver recorrências lineares com coeficientes polinomiais. Resolução de recorrências revelar-se-á crucial em quase todos os algoritmos que apresentaremos.

Listaremos, a seguir, alguns exemplos de resolução de recorrências com esta ferramenta.

Exemplo (Soma da progressão aritmética).

```
(%i1) := load(solve_rec);
(%i2) := solve_rec(f(n) - f(n-1) = n, f(n), f(0)=0);
```

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

Exemplo (Fibonacci).

```
(%i1) := solve_rec(f(n) = f(n-1) + f(n-2), f(n), f(0)=0, f(1)=1);
```

$$f(n) = \frac{(\sqrt{5}+1)^n}{\sqrt{5}\,2^n} - \frac{(\sqrt{5}-1)^n\,(-1)^n}{\sqrt{5}\,2^n}$$

Exemplo. Resolvendo

$$n^2 f(n) + (1-n^2) f(n-1) = 1 \qquad (n > 1; f(1) = 1)$$

para $f(n)$. Aqui, ratsimp simplifica uma fração.

```
(%i1) := e:solve_rec((n^2)*f(n) + (1-n^2)*f(n-1) = 1, f(n), f(1)=1);
(%i2) := ratsimp(e);
```

$$f(n) = 1$$

3 A classe hipergeométrica

Neste capítulo traremos uma visão global das funções hipergeométricas. Os objetos aqui apresentados são necessários para prosseguir na leitura dos capítulos seguintes.

Apresentaremos o termo hipergeométrico e sua generalização natural para duas variáveis, que permite-nos definir a classe de funções hipergeométrica. Por fim, explicitaremos a noção de forma fechada.

3.1 Termo hipergeométrico

Definição (Termo hipergeométrico). Dizemos que $t(k)$ é um termo hipergeométrico em k se e somente se sua razão discreta, $t(k+1)\,/\,t(k)$, é uma função racional em k,

$$\frac{t(k+1)}{t(k)} = \frac{(k+a_1)(k+a_2)\cdots(k+a_p)}{(k+b_1)(k+b_2)\cdots(k+b_q)}\,\alpha, \tag{3.1.1}$$

com coeficientes num corpo $\mathbb{K}$ de característica 0.

Ao longo do nosso texto, resolveremos problemas envolvendo $\mathbb{K}=\mathbb{Q}$, possivelmente com alguma extensão, como $\mathbb{Q}(n)$ ou $\mathbb{Q}(n,k)$.

Obter uma fatoração de $t(k+1)\,/\,t(k)$ como em (3.1.1) pode envolver fatorar as expressões do numerador e do denominador desta razão. Fatorar um polinômio super-quártico, segundo o teorema de Matiyasevich (MATIYASEVICH, 1970), excede os limites da computação para $\mathbb{K}=\mathbb{R}$; esta é a resposta para o décimo problema da famosa lista de Hilbert. Contudo, existem soluções para algumas restrições deste problema, como (FRÖHLICH; SHEPHERDSON, 1955) e (LENSTRA; JR; LOVASZ, 1982).

Exemplo (Termo hipergeométrico). Considere $t(k) = (2k+1)!\,/\,(4k+1)!$, cuja razão discreta,

$$\begin{aligned}
\frac{t(k+1)}{t(k)} &= \frac{(2(k+1)+1)!}{(4(k+1)+1)!}\frac{(4k+1)!}{(2k+1)!}\\
&= \frac{(2k+3)!}{(2k+1)!}\frac{(4k+1)!}{(4k+5)!}\\
&= \frac{(2k+3)(2k+2)}{(4k+5)(4k+4)(4k+3)(4k+2)}\\
&= \frac{\left(k+\frac{3}{2}\right)}{\left(k+\frac{5}{4}\right)\left(k+\frac{3}{4}\right)\left(k+\frac{1}{2}\right)}\,\frac{2^2}{4^4}\\
&= \frac{\left(k+\frac{3}{2}\right)}{\left(k+\frac{5}{4}\right)\left(k+\frac{3}{4}\right)\left(k+\frac{1}{2}\right)}\,2^{-6},
\end{aligned}$$

é da forma (3.1.1) e prontamente o reconhecemos como um termo hipergeométrico.

Exemplo. Agora, considere $t(k) = e^{-k^2}$. A razão discreta, $e^{-(k+1)^2}/e^{-k^2} = e^{-2k-1}$, não é uma razão de dois polinômios em k e nem mesmo uma constante independente de k pois ao tomarmos conhecimento de que a soma das potências é infinita,

$$\frac{e^{-(k+1)^2}}{e^{-k^2}} = e^{-2k-1} = \sum_{n\geqslant 0} (-1)^n (2k+1)^n,$$

resta-nos apenas rejeitar e^{-k^2} como sendo um termo hipergeométrico em k.

Exemplos (Termos hipergeométricos).

$$1, \qquad \frac{1}{k!}, \qquad \frac{(7k+1)!^2}{(2k)!^3}, \quad (-1)^k \binom{k+2}{2}$$

são termos hipergeométricos na variável k.

Devemos, ainda, apresentar algumas propriedades de termos hipergeométricos. O conjunto de todos os termos hipergeométricos numa variável munido da operação de multiplicação forma um grupo abeliano mas sequer suporta a propriedade de fechamento para a adição.

3.2 Termo duplamente hipergeométrico

Definição. Um termo $F(n,k)$ é duplamente hipergeométrico se as razões

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

são funções racionais em n e k.

Exemplos (Termo duplamente hipergeométrico).

$$\binom{n}{k}, \qquad -1, \qquad (-1)^k \binom{n+k}{k}\binom{2n}{n}, \qquad n^2\,3^k, \qquad \frac{(n+k)^2}{(n-k)^2}(-2)^{n+k}.$$

3.3 Classe hipergeométrica

Eventualmente, referiremo-nos às funções da classe hipergeométrica e, então, precisamos explicitar qual é a esta classe. Entretanto, trabalharemos num subconjunto restrito destas funções.

Definição (Classe hipergeométrica). A *classe hipergeométrica* com v variáveis é o conjunto de todas as combinações lineares de termos hipergeométricos em v variáveis, onde cada um destes termos é definido sobre um corpo de característica zero.

Neste texto, trabalhamos apenas com $v \leqslant 2$.

3.4 Forma fechada

A essência dos nossos problemas consiste em "resolver" uma soma cujo somando é um termo hipergeométrico. Aqui, "resolver" uma soma significa obter uma expressão equivalente e em forma fechada para ela. Antes de tudo, vamos formalizar a noção de forma fechada.

Definição (Forma fechada). Uma forma fechada é uma combinação linear de termos hipergeométricos com coeficientes num corpo de característica zero.

Exemplos. Se n, r e s são naturais, então

$$\begin{aligned}
2^n &\quad \text{é uma forma fechada para} \quad \sum_k \binom{n}{k} \\
\tbinom{n+1}{2} &\quad \text{é uma forma fechada para} \quad \sum_{k=0}^{n} i \\
(-1)^k \tbinom{s}{n-r} &\quad \text{é uma forma fechada para} \quad \sum_k (-1)^k \binom{r}{k} \binom{s+k}{n}.
\end{aligned}$$

O último exemplo demanda uma explicação adequada. $\binom{s}{n-r}$ é um termo hipergeométrico no corpo $\mathbb{Q}(s,n,r)$ e a forma fechada supracitada tem um coeficiente, $(-1)^k$, em $\mathbb{Q}(k)$. Note, ainda, que todas as variáveis envolvidas são independentes entre si.

4 O problema da soma hipergeométrica definida

O algoritmo da irmã Mary Celine Fasenmyer, descrito em sua tese de PhD (FASENMYER, 1947), é uma solução para o problema de determinar uma recorrência satisfeita por uma dada soma de termos duplamente hipergeométricos.

É garantida, (FASENMYER, 1947) e (PETKOVsEK; WILF; ZEILBERGER, 1996), a existência de tal recorrência para uma soma cujo somando é um termo hipergeométrico em duas variáveis: uma recorrência de ordem suficientemente grande pode ser ajustada à soma em questão.

Este algoritmo conta-nos que, por exemplo, os termos $R(n,k)$ e $S(n,k)$ duplamente hipergeométricos em

$$r(n) \stackrel{\text{def.}}{=} \sum_k \underbrace{(-1)^k \binom{n+k-1}{k}}_{R(n,k)} \qquad \text{e} \qquad s(n) \stackrel{\text{def.}}{=} \sum_k \underbrace{\binom{2n}{n}\binom{n}{k}}_{S(n,k)}$$

satisfazem, respectivamente, as recorrências

$$\begin{aligned} R(n,k) + R(n,k-1) - R(n-1,k) &= 0 \\ (2-4n)(S(n-1,k-1) + S(n-1,k)) + nS(n,k) &= 0 \end{aligned}$$

e, posteriormente, ainda veremos que, a partir destas recorrências, podemos obter uma recorrência para cada um destes somatórios; nos exemplos, são

$$\begin{aligned} r(n) &= 0 \\ s(n) &= \frac{4}{n}(2n-1)s(n-1). \end{aligned}$$

4.1 Descrição do problema

A primeira peça do nosso arcabouço é resolver o problema

instância: $f(n) \stackrel{\text{def.}}{=} \sum_k F(n,k)$, onde $F(n,k)$ é duplamente hipergeométrico.

resposta: Uma recorrência satisfeita por $F(n,k)$.

Usando a recorrência encontrada pelo algoritmo, podemos reescrevê-la em termos de $f(n)$, como veremos. Resolvendo esta última recorrência, temos uma forma fechada para $f(n)$.

4.2 O algoritmo da irmã Celine

Suponha que nos seja dada a tarefa de resolver uma soma $f(n) = \sum_k F(n,k)$, onde F é duplamente hipergeométrico em $(n,k) \in \mathbb{N} \times \mathbb{Z}$. Seria suficiente encontrarmos uma recorrência para $f(n)$, resolvê-la e argumentar que o problema está resolvido.

A grande ideia da irmã Celine consiste em encontrar uma recorrência para $F(n,k)$ e somá-la sobre todos os k inteiros, construindo, assim, uma recorrência em termos de $f(n)$. Resolver esta recorrência para $f(n)$ implica resolver $\sum_k F(n,k)$, como desejado.

A recorrência que procuramos é da forma

$$\sum_{i=0}^{I} \sum_{j=0}^{J} a_{i,j}(n)\, F(n-j, k-i) \quad = \quad 0, \tag{4.2.1}$$

onde o somatório duplo é explicado por percorrer as duas variáveis, n e k, colocando um coeficiente $a_{i,j}(n) \in \mathbb{Q}(n)$ para cada $F(n-j, k-i)$. Adiante, veremos que existem coeficientes não todos nulos que satisfazem (4.2.1) para I e J limitados.

Para apresentar o algoritmo, devemos apresentar três funções importantes: $\text{polinomio}\,(expr, var)$, $\text{coeficiente}\,(expr, var, grau)$ e $\text{resolva}\,(eq, vars)$.

- ⋆ $\text{polinomio}\,(expr, var)$ distribui o produto sobre a soma na expressão $expr$ e coleta neste resultado as potências de var individualmente. O resultado é que $expr$ é reescrito como uma soma das potências de var. No MAXIMA, esta funcionalidade corresponde à chamada $collectterms(expand(expr), var)$;
- ⋆ $\text{coeficiente}\,(expr, var, grau)$ obtém o coeficiente de var^{grau} na expressão $expr$. No MAXIMA, esta funcionalidade corresponde à chamada de $coeff(expr, var, grau)$;
- ⋆ $\text{resolva}\,(eq, vars)$ resolve a equação eq para as variáveis $vars$. No MAXIMA, esta funcionalidade corresponde à chamada de $solve(eq, vars)$;

Uma alternativa para esta documentação é via shell do MAXIMA com o comando $describe$. Por exemplo, para consultar $collectterms$, digite $describe(collectterms)$; no terminal.

Algoritmo 1: celineIJ $(F(n,k),\ I,\ J)$

Entrada: $F(n,k)$ duplamente hipergeométrico e $(I,J) \in \mathbb{N}^2$.
Saída: Uma matriz cuja célula (i,j) corresponde ao coeficiente $a_{i,j}(n)$.

1 $N(n,k) \leftarrow \text{numerador}\left(\sum_{i=0}^{I}\sum_{j=0}^{J} a_{i,j}(n)\dfrac{F(n-j,k-i)}{F(n,k)}\right)$
2 $N(k) \leftarrow \text{polinomio}(N(n,k),\ k)$
3 **para cada** $i \in [0..\,\text{grau}\,(N(k))]$ **faça**
4 $\quad C_i \leftarrow \text{coeficiente}(N(k),\ k,\ i)$
5 **retorna** $\text{resolva}\left(C=0,\ \left\{\{a_{i,j}\}_{i=0}^{I}\right\}_{j=0}^{J}\right)$

Procedemos com uma descrição deste algoritmo.

Linhas 1-2: Primeiramente, (4.2.1) é equivalente a

$$\sum_{i=0}^{I}\sum_{j=0}^{J} a_{i,j}(n)\,\frac{F(n-j,k-i)}{F(n,k)} \quad = \quad 0 \tag{4.2.2}$$

e, logo, apenas o numerador da expressão resultante de (4.2.2), que chamamos de $N(n,k)$, nos interessa.

Atente-se que $F(n+1,k)\,/\,F(n,k)$ e $F(n,k+1)\,/\,F(n,k)$ são funções racionais em n e k, por isso a escolha de $F(n,k)$ como denominador em (4.2.2).

Linhas 3-4: Armazenamos em C_i o coeficiente de k^i da expressão $N(k)$. A expressão $N(k)$ é equivalente à $N(n,k)$ mas reescrita como um polinômio em k, na segunda linha do algoritmo.

Uma condição necessária e suficiente para (4.2.2) é que $N(k)=0$. Entretanto, isto só acontece quando $C_i = 0$, para todo $0 \leqslant i \leqslant \text{grau}\,(N(k)) = g$, porque $k^0, k^1, \cdots, k^g$ são linearmente independentes.

Linhas 5-6: Resolvemos o sistema de equações $C = 0$ para os $a_{i,j}$'s.

Por razões que logo se revelarão, todos os $a_{i,j}(n)$'s devem depender apenas de n e nunca de k.

Exemplo (Binomial). Vamos encontrar uma recorrência de ordem $I = J = 1$ para $F(n,k) \stackrel{\text{def.}}{=} \binom{n}{k}$ usando o algoritmo celineIJ. Conforme (4.2.1), procuraremos uma recorrência, com $a_{i,j} \in \mathbb{Q}(n)$, da forma

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

O primeiro passo é obter o numerador de (4.2.2). Temos que

$$\sum_{i=0}^{I}\sum_{j=0}^{J} a_{i,j}(n)\frac{F(n-j,k-i)}{F(n,k)} \quad = \quad a_{0,0} + a_{0,1}\frac{k}{n-k+1} + a_{1,0}\frac{n-k}{n} + a_{1,1}\frac{k}{n}.$$

Colocando o segundo termo da equação sobre um mesmo denominador, $\mathrm{mmc}(1, n-k+1, n) = (n-k+1)\,n$, temos

$$\frac{a_{0,0}\,(n-k+1)\,n + a_{0,1}\,k\,n + a_{1,0}\,(n-k)(n-k+1) + a_{1,1}\,k\,(n-k+1)}{(n-k+1)\,n}$$

cujo numerador chamaremos de $N(n,k)$.

No segundo passo, expandimos todas as potências de $N(n,k)$ e agrupamos os coeficientes cujos fatores de k^i são idênticos, isto é, reescrevemos $N(n,k)$ como um polinômio em k,

$$\Big(a_{1,0}-a_{1,1}\Big)\,k^2 + \Big(-a_{0,0}\cdot n + a_{0,1}\cdot n - a_{1,0}\cdot n - a_{1,0}\cdot(n+1) + a_{1,1}\cdot(n+1)\Big)\,k^1 + \Big(a_{0,0}\cdot(n+1)\,n + a_{1,0}\cdot(n+1)\,n\Big)\,k^0,$$

e o chamamos de $N(k)$.

Os passos 3-4 envolvem armazenar os coeficientes de $N(k)$ no respectivo C_i,

$$\begin{aligned} C_0 &\leftarrow a_{0,0}\cdot(n+1)\,n + a_{1,0}\cdot(n+1)n \\ C_1 &\leftarrow -a_{0,0}\,n + a_{0,1}\,n - a_{1,0}\,n - a_{1,0}\cdot(n+1) + a_{1,1}\cdot(n+1) \\ C_2 &\leftarrow a_{1,0} - a_{1,1}. \end{aligned}$$

O quinto passo trata de resolvermos $C_i = 0$, $0 \leqslant i \leqslant \mathrm{grau}\,(N(k))$, para $\left\{ \{a_{i,j}\}_{i=0}^{I} \right\}_{j=0}^{J}$. Uma das formas de resolver este sistema é usar o algoritmo de eliminação gaussiana. A solução procurada é

$$A = \begin{pmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{pmatrix} = \begin{pmatrix} -a_{1,1} & 0 \\ a_{1,1} & a_{1,1} \end{pmatrix},$$

isto é, a solução apresenta um grau de liberdade. Enfim, terminamos a execução de $\mathrm{celineIJ}\left(\binom{n}{k},\ 1,\ 1\right)$.

A recorrência buscada é, então,

$$-a_{1,1}\binom{n}{k} + a_{1,1}\binom{n-1}{k} + a_{1,1}\binom{n-1}{k-1} = 0, \tag{4.2.3}$$

também conhecida como identidade de Pascal,

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}. \tag{4.2.4}$$

Para complementar nosso exemplo, suponha ainda que queiramos resolver $f(n) \stackrel{\text{def.}}{=} \sum_k \binom{n}{k}$. Como os $a_{i,j}$'s são livres de k, somamos (4.2.4) sobre todos os k inteiros,

$$\sum_k \binom{n}{k} = \sum_k \left(\binom{n-1}{k} + \binom{n-1}{k-1} \right) = \underbrace{\sum_k \binom{n-1}{k}}_{f(n-1)} + \underbrace{\sum_k \binom{n-1}{k-1}}_{f(n-1)},$$

para eliminar uma das variáveis, observando que

$$f(n) = 2\,f(n-1) = \cdots = 2^n.$$

4.3 Uma recorrência para a soma definida

Uma vez que $\mathrm{celineIJ}$ encontra uma solução, ainda estamos interessados em encontrar uma recorrência para a soma definida.

Do exemplo anterior,

$$\begin{pmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{pmatrix} = \mathrm{celineIJ}\left(\binom{n}{k}, 1, 1\right) = \begin{pmatrix} -a_{1,1} & 0 \\ a_{1,1} & a_{1,1} \end{pmatrix}$$

e, então, a recorrência (4.2.3) implícita é

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

Somando esta recorrência sobre todos os k inteiros, temos

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

se e somente se

$$-a_{1,1}f(n) + a_{1,1}f(n-1) + a_{1,1}f(n-1) = 0$$

ou, equivalentemente, $f(n) = 2f(n-1)$. A condição inicial, $f(0) = 1$, leva-nos a concluir que $f(n) = 2^n$.

Esta explicação corresponde à execução do algoritmo a seguir para a supracitada matriz.

Algoritmo 2: $\mathrm{celinerec}\,(M)$

Entrada: Uma matriz M de l linhas e c colunas.
Saída: Uma recorrência para a soma definida.

para cada $i \in [0..l-1]$ **faça**
  $a_i \leftarrow 0$
  para cada $j \in [0..c-1]$ **faça**
    $a_i \leftarrow a_i + M[i,j]$

retorna $\sum_{i=0}^{c-1} a_i\, f(n-i) = 0$

Em suma, basta obter a soma marginal da linha i, a_i, e a recorrência procurada para a soma definida será $\sum_{i=0}^{c-1} a_i\, f(n-i) = 0$.

A justificativa para a inibição dos coeficientes da matriz dada independerem de k é que a recorrência para a soma definida, que deve ser resolvida para $f(n)$, independa de k. Caso contrário, a solução sujeitar-se-ia a apenas alguns valores de k, indo contra o objetivo deste capítulo.

Acompanhado ao nosso pacote, vêm implementações homônimas de $\mathrm{celineIJ}$ e $\mathrm{celinerec}$.

Exemplo. Considere a soma $f(n) \stackrel{\text{def.}}{=} \sum_k F(n,k)$, onde $F(n,k) \;=\; (-1)^k \binom{n+k-1}{k}$.

```
(%i1) := F(n,k) := (-1)^k * binomial(n+k-1, k);
(%i2) := M : celineIJ(F,n,k,2,2);
(%i3) := M;
```

$$[[K_3, K_2, -K_3 - K_2],\ [K_1, -K_4 + 2K_3 + K_2 - K_1, K_4],\ [K_1 - K_3, -K_4, 0]]$$

Depois de encontrada uma recorrência para $F(n,k)$ via celineIJ, obtemos, também, uma recorrência para $f(n)$ via celinerec,

```
(%i1) := celinerec(M);
```

$$(2K_3 + K_2)f_{n-1} + (-K_4 - K_3 + K_1)f_{n-2} \;=\; 0$$

onde $K_1, \cdots, K_4$ são constantes a serem determinadas.

4.4 Limitantes para I e J

Definição (Termo hipergeométrico próprio). $F(n,k)$ é um termo hipergeométrico próprio (*thp*) se ele pode ser escrito na forma

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

onde

- ⋆ $P(n,k)$ é um polinômio em n e k;
- ⋆ os a's, b's, u's, v's são inteiros pré-determinados (i.e., não dependem de nenhum outro parâmetro);
- ⋆ uu e vv são números naturais;
- ⋆ x, c's e w's são números complexos;

Estas restrições são tais que garantem que F seja *bem definido* no ponto (n,k) quando o argumento de todos os fatorais no numerador são números naturais, i.e., c_i e w_i também devem ser números naturais. Além disso, só permitiremos que $F(n,k) = 0$ se

- ⋆ $P(n,k) = 0$; ou
- ⋆ F é bem definido em (n,k) e pelo menos um dos fatoriais no denominador tiver o argumento negativo –neste caso, use $(-z)!\,\Gamma(z) \;=\; \frac{\pi}{\sin(\pi z)}$.

Exemplos de thp's são

- $\star$ $\binom{n}{k}^2 2^k$
- $\star$ $(n-k)\,\frac{(n+k-1)!}{(k-1)!(n-1)!}\,(2+i)^k$
- $\star$ termos duplamente hipergeométricos.

Teorema 4.4.1 (Limitantes para I e J)**.** *Existem inteiros positivos I e J e polinômios $a_{i,j} \in \mathbb{Q}(n)$ para $0 \leqslant i \leqslant I$ e $0 \leqslant j \leqslant J$, não todos nulos, tais que*

$$\sum_{i=0}^{I} \sum_{j=0}^{J} a_{i,j}\, F(n-j, k-i) \quad = \quad 0, \tag{4.4.1}$$

para todo n natural e k inteiro quando

- $\star$ *F é bem definido em $(n-j, k-i)$ e*
- $\star$ *A recorrência (4.4.1) vale para todo (n,k) quando $F(n,k) \neq 0$ é satisfeito.*

Além disso, (I^*, J^*),

$$\begin{aligned} J^* &= \sum_s |b_s| + \sum_s |v_s| \\ I^* &= 1 + \mathrm{grau}\,(P) + J^* \left(-1 + \sum_s |a_s| + \sum_s |u_s| \right), \end{aligned}$$

é um limitante para (I,J).

Prova. Veja a tese de PhD da irmã Celine (FASENMYER, 1947) ou a prova em (PETKOVsEK; WILF; ZEILBERGER, 1996, p.64-68). □

4.5 Implementação e considerações finais

O algoritmo deste capítulo é fundamental para toda a teoria das funções hipergeométricas mas, na prática, ele não é implementado. O algoritmo de Zeilberger (capítulo 7) resolve o mesmo problema de forma mais eficiente em relação ao tempo.

Existem generalizações (PETKOVsEK; WILF; ZEILBERGER, 1996) do teorema fundamental expandindo a solução –o algoritmo da irmã Celine– em que $\boldsymbol{k}$ é um vetor de inteiros em $F(n, \boldsymbol{k})$. Também, há outras generalizações (e.g., (ZEILBERGER, 1982)) envolvendo a teoria q-Analog: q-sums e q-multisums. Neste texto, tratamos apenas do caso univariado.

5 O problema da soma hipergeométrica indefinida

O algoritmo de Gosper, (PETKOVsEK; WILF; ZEILBERGER, 1996) e (GOSPER, 1978), resolve o problema da soma hipergeométrica indefinida. O objetivo é encontrar uma forma fechada para $s_n \stackrel{\text{def.}}{=} \sum_{k=0}^{n-1} t_k$, onde t_k é um termo hipergeométrico em k, independente de n. Poderemos, ao final deste capítulo, resolver somas como

$$\sum_{k=0}^{n-1} k \cdot k!, \qquad \sum_{k=0}^{n-1} x^k, \qquad \sum_{k=0}^{n-1} \frac{1}{(1-x)^k}, \qquad \sum_{k=0}^{n-1} \binom{2k}{k} \Big/ 4^k \quad \text{e} \quad \sum_{k=0}^{n^2} \binom{k}{2}$$

(x constante independente de n)

e, por outro lado, afirmar que somas como

$$\sum_{k=1}^{n} \frac{1}{k}, \qquad \sum_{k=1}^{n} \frac{\binom{2k}{k}}{k}, \qquad \sum_{k=1}^{n} \frac{k^2}{k!} \quad \text{e} \quad \sum_{k=0}^{n-1} \frac{1}{k!}$$

não possuem solução na classe hipergeométrica.

Apresentaremos o problema da soma hipergeométrica indefinida e o algoritmo de Gosper acompanhado de alguns exemplos.

5.1 Descrição do problema

O problema da soma hipergeométrica indefinida e sua solução, pelo algoritmo de Gosper, ata o conceito de derivada discreta de uma soma [hipergeométrica indefinida] com um termo hipergeométrico. O conceito é análogo ao teorema fundamental do cálculo, que estabelece que

$$\int_a^b f(x)\, \mathrm{d}x \quad = \quad F(b) - F(a),$$

onde F é a anti-derivada da função f de domínio real; tendo resolvido $F(x) = \int f(x)\,\mathrm{d}x$, basta quantificar os limites, a e b, e computar $F(b) - F(a)$.

A ideia de Gosper é encontrar um $z(k)$ tal que $z(k+1) - z(k) = t(k)$, para todo k inteiro. Dispondo de tal $z(k)$, teríamos que

$$\sum_{k=a}^{b} t(k) \quad = \quad \sum_{k=a}^{b} (z(k+1) - z(k))$$

e, como esta última soma é telescópica, o resultado é $z(b+1) - z(a)$.

É claro que, sem restrições, sempre existe um $z(k)$ para a soma envolvida. Por exemplo, um $z(k)$ para $\sum_{k=0}^{n-1} k \cdot k!$ seria esta própria soma. Entretanto, este $z(k)$ não é frutífero e, portanto, precisamos restringir o espaço de valores que $z(k)$ pode assumir, levando-nos a definir o problema deste capítulo.

instância: um termo hipergeométrico $t(k)$, independente dos limites a e b da soma.

decidir: existe um termo hipergeométrico $z(k)$ satisfazendo

$$z(k+1) - z(k) = t(k) \tag{5.1.1}$$

?

Para uma instância positiva (i.e., cuja resposta seja "sim"), queremos $z(k)$; caso contrário, queremos uma garantia de que não exista tal $z(k)$. O algoritmo de Gosper resolve este problema devolvendo um $z(k)$ para o termo hipergeométrico $t(k)$ fornecido ou, na ausência definitiva de uma solução hipergeométrica, devolve "não".

Deduzimos, a partir de (5.1.1), que

$$z(n) = z(n-1) + t(n-1) = \cdots = z(0) + \sum_{k=0}^{n-1} t(k)$$

se e somente se

$$\sum_{k=0}^{n-1} t(k) = z(n) - z(0) \qquad (z(0) \text{ constante})$$

e, sendo $z(n)$ um termo hipergeométrico, $z(n) - z(0)$ é uma forma fechada para esta soma. Este $z(n)$ é dito *anti-derivada discreta* de $t(n)$ ou, também, *anti-diferença* de $t(n)$ porque sua derivada discreta (veja apêndice (A)) resulta em $t(n)$. O mecanismo é análogo à primitivação e integração do cálculo infinitesimal.

Quando um $t(k)$ não admite um $z(k)$ satisfazendo (5.1.1), estamos diante de um $t(k)$ que não tem solução hipergeométrica no corpo $\mathbb{Q}(k)$. Isso significa que $t(k)$ não pode ser indefinidamente somado porque não existe uma *anti-diferença* $z(k)$ na classe hipergeométrica (vide capítulo 3) que satisfaça (5.1.1). Neste caso, dizemos que a soma indefinida do termo $t(k)$ não é *Gosper-somável*.

5.2 O algoritmo de Gosper

Com a restrição de que $z(n)$ deve ser um termo hipergeométrico satisfazendo (5.1.1), sabemos que $y(n)$,

$$y(n) \stackrel{\text{def.}}{=} \frac{z(n)}{t(n)} = \frac{z(n)}{z(n+1)-z(n)} = \frac{1}{\frac{z(n+1)}{z(n)}-1},$$

é uma função racional em n (vide capítulo (3.1)). Trocando $z(n)$ por $y(n)t(n)$ em (5.1.1), temos

$$\begin{aligned} z(n+1)-z(n) = t(n) &\iff y(n+1)t(n+1) - y(n)t(n) = t(n) \\ &\iff y(n+1)\frac{t(n+1)}{t(n)} - y(n) = 1, \end{aligned}$$

uma recorrência linear de primeira ordem com coeficientes racionais. Noutras palavras, dado $t(n)$, basta-nos resolver

$$y(n+1)\frac{t(n+1)}{t(n)} - y(n) = 1 \tag{5.2.1}$$

e devolver $y(n)t(n)$, nosso termo $z(n)$ procurado. Portanto, encontrar soluções hipergeométricas para (5.1.1) é equivalente a encontrar soluções racionais para (5.2.1).

Ainda, fatoramos $t(n+1)/t(n)$ como

$$\frac{a(n)}{b(n)}\,\frac{c(n+1)}{c(n)}, \tag{5.2.2}$$

onde $a(n)$, $b(n)$ e $c(n)$ são polinômios em n satisfazendo

$$\operatorname{mdc}(a(n),\, b(n+h)) = 1, \qquad h \in \mathbb{N} \tag{5.2.3}$$

e, segundo a escolha de Gosper (GOSPER, 1978), procuramos uma solução da forma

$$y(n) = \frac{b(n-1)\,x(n)}{c(n)}. \tag{5.2.4}$$

O resultado é que temos, de (5.2.1),

$$\underbrace{\underbrace{\underbrace{\frac{b(n)\,x(n+1)}{c(n+1)}}_{=\,y(n+1)} \cdot \underbrace{\frac{a(n)\,c(n+1)}{b(n)\,c(n)}}_{=\,t(n+1)/t(n)}}_{=\,\frac{a(n)}{c(n)}x(n+1)} - \underbrace{\frac{b(n-1)x(n)}{c(n)}}_{=\,y(n)}}_{=\,\frac{a(n)x(n+1)\,-\,b(n-1)x(n)}{c(n)}} = 1 \tag{5.2.5}$$

se e somente se

$$a(n)x(n+1) - b(n-1)x(n) = c(n). \tag{5.2.6}$$

A escolha de Gosper para que $y(n)$ seja fatorado como (5.2.4) explica-se por levar $x(n)$ a ser um polinômio.

Teorema 5.2.1. *Sejam $a(n)$, $b(n)$ e $c(n)$ polinômios em n sujeitos a (5.2.3). Se $x(n)$ é uma função racional de n satisfazendo (5.2.6), então $x(n)$ é um polinômio em n.*

Prova. Veja (PETKOVsEK; WILF; ZEILBERGER, 1996, p.76) ou (GOSPER, 1978). □

Portanto, encontrar soluções hipergeométricas para (5.1.1) é equivalente a encontrar soluções polinomiais para (5.2.6).

Algoritmo 3: $\text{gosper}(t,n)$

Entrada: Um termo hipergeométrico $t(n)$ na variável n.
Saída: $z(n)$ satisfazendo (5.1.1).
1 $(a(n), b(n), c(n)) \leftarrow \text{gosperabc}(t, n)$
2 resolva $a(n)x(n+1) - b(n-1)x(n) = c(n)$ para $x(n)$
3 **se** $x(n)$ *não for um polinômio não nulo* **então retorna** *"não"*
4 **retorna** $\dfrac{b(n-1)x(n)}{c(n)}t(n)$

Em seguida, esclarecemos os passos deste algoritmo.

Linha 1: gosperabc obtém uma fatoração de $t(n+1)\ /\ t(n)$ conforme (5.2.2). Dedicamos a seção (5.3), a seguir, para apresentar e esclarecer este algoritmo.

Linha 2: Sabemos que a resposta deste algoritmo, $y(n)t(n)$, depende de resolvermos a recorrência (5.2.1). Ao invés disto, resolvemos $x(n)$ em (5.2.6) para substitui-lo, posteriormente, em (5.2.4).

Uma maneira simples de resolver esta recorrência é tentar ajustar um polinômio em n de grau $d = 0, 1, 2, \cdots$. Limitantes para d podem ser encontrados em (PETKOVsEK; WILF; ZEILBERGER, 1996).

Linha 3: Como a fatoração de (5.2.2) é única, como veremos, se tal polinômio não-nulo inexistir, definitivamente não há uma anti-diferença hipergeométrica para $t(n)$ e o algoritmo devolve "não".

Linha 4: Devolve $y(n)t(n)$, a anti-diferença hipergeométrica procurada.

Ao final deste capítulo, apresentamos alguns exemplos de execução, implementações disponíveis e aplicações.

5.3 Fatoração de Gosper

Para conveniência dos cálculos, gosperabc exige que a razão discreta $t(n+1)/t(n)$ esteja numa forma especial. Trata-se de uma razão de polinômios mônicos e primos entre si de modo que o coeficiente do maior monômio, tanto do numerador quanto do denominador, seja compensado numa constante exógena à esta razão. Isto é, nosso objetivo é escrever a razão discreta $t(n+1)/t(n)$ como

$$\frac{t(n+1)}{t(n)} = Z \cdot \frac{f(n)}{g(n)}, \tag{5.3.1}$$

onde Z é uma constante e $f(n)$ e $g(n)$ são polinômios satisfazendo

$$\mathrm{cmm}(f(n)) = \mathrm{cmm}(g(n)) = 1 \quad \text{e} \quad \mathrm{mdc}(f(n), g(n)) = 1.$$

Esta forma especial pode ser obtida via gosperzfg, a seguir.

Algoritmo 4: $\mathrm{gosperzfg}\,(t, n)$

Entrada: Um termo hipergeométrico $t(n)$ na variável n.
Saída: $(Z,\, f(n),\, g(n))$ na forma (5.3.1).

1 $r(n) \leftarrow t(n+1) \;/\; t(n)$
2 $F(n) \leftarrow \mathrm{numerador}\,(r(n))$
3 $G(n) \leftarrow \mathrm{denominador}\,(r(n))$
4 $f(n) \leftarrow F(n) \;/\; \mathrm{cmm}(F(n))$
5 $g(n) \leftarrow G(n) \;/\; \mathrm{cmm}(G(n))$
6 $Z \leftarrow \mathrm{cmm}(F(n)) \;/\; \mathrm{cmm}(G(n))$
7 **retorna** $(Z,\, f(n),\, g(n))$

Procedemos com uma explicação sucinta dos passos de gosperzfg.

Linhas 1-3: Tomamos $F(n)$ e $G(n)$ como sendo o numerador e o denominador, respectivamente, da expressão resultante de $t(n+1) \;/\; t(n)$.

Linhas 4-5: Forçamos $F(n)$ e $G(n)$ a serem polinômios mônicos,

$$f(n) \leftarrow F(n)/\mathrm{cmm}(F(n)) \quad \text{e} \quad g(n) \leftarrow G(n)/\mathrm{cmm}(G(n)).$$

Note, ainda, que $f(n)$ e $g(n)$ têm as mesmas raízes de $F(n)$ e $G(n)$, respectivamente.

Linha 6: Os fatores $\mathrm{cmm}(F(n))$ e $\mathrm{cmm}(G(n))$ são compensados em Z.

Linha 7: Obtemos Z, $f(n)$ e $g(n)$ conforme (5.3.1).

Nosso pacote porta uma implementação homônima deste algoritmo.

Exemplo (gosperzfg). Aqui, reescrevemos $t(n+1)/t(n)$, onde $t(n) = \binom{2n}{n} 2^n$, conforme (5.3.1).

```
(%i1) := t(n) := binomial(2*n,n) * 2^n;
(%i2) := gosperzfg(t, n);
```

$$[8,\; n+\frac{1}{2},\; n+1]$$

Daqui, continuamos com $\mathrm{gosperabc}$, o algoritmo que obtém a fatoração (5.2.2).

Algoritmo 5: $\mathrm{gosperabc}\,(t,n)$

Entrada: Um termo hipergeométrico $t(n)$.
Saída: $(a(n), b(n), c(n))$ satisfazendo (5.2.2).

1 $(Z, f(n), g(n)) \leftarrow \mathrm{gosperzfg}\,(t,n)$; /* $t(n+1)/t(n) = Z f(n)/g(n)$ */
2 $R(h) \leftarrow \mathrm{resultante}\,(f(n), g(n+h))$; /* $R(h) = \prod_{i=1}^{D}(h-r_i),\ \ r_i < r_{i+1}$ */
3 $N \leftarrow 0$
4 **para cada** $i \in [1..D]$ **faça**
5   **se** $r_i \geqslant 0$ **então**
6     $N \leftarrow N+1$
7     $h_N \leftarrow r_i$
8 $p_0(n) \leftarrow f(n)$
9 $q_0(n) \leftarrow g(n)$
10 **para cada** $i \in [1..N]$ **faça**
11   $s_j(n) \leftarrow \mathrm{mdc}(p_{j-1}(n), q_{j-1}(n+h_j))$
12   $p_j(n) \leftarrow p_{j-1}(n)\ /\ s_j(n)$
13   $q_j(n) \leftarrow q_{j-1}(n)\ /\ s_j(n-h_j)$
14 $a(n) \leftarrow Z \cdot p_N(n)$
15 $b(n) \leftarrow q_N(n)$
16 $c(n) \leftarrow \prod_{i=1}^{N}\prod_{j=1}^{h_i} s_i(n-j)$
17 **retorna** $(a(n), b(n), c(n))$; /* $\frac{t(n+1)}{t(n)} = \frac{a(n)}{b(n)}\frac{c(n+1)}{c(n)}$ */

A execução de $\mathrm{gosperabc}\,(t,n)$ garante que $t(n+1)/t(n)$ possa ser fatorado como (5.2.2) sujeito a $\mathrm{mdc}(a(n), b(n+h)) = 1$, para todo h natural, como veremos. Prosseguimos com uma explicação deste algoritmo.

Linha 1: $t(n+1)/t(n)$ é fatorado como (5.3.1).

Linhas 2-7 Se, de antemão, tivéssemos garantido que (5.2.3), então bastaria-nos tomar $a(n) = f(n)$, $b(n) = g(n)$ e $c(n) = 1$. Porém, como este não é o caso geral, precisamos forçar esta condição.

Uma maneira de garantir tal condição, diferentemente da abordagem de Gosper, é usar $\mathrm{resultante}\,(f(n), g(n+h))$. No apêndice (B), apresentamos uma revisão lúcida da $\mathrm{resultante}$.

Neste trecho, obtemos as N raízes $h_1, \cdots, h_N \in \mathbb{N}$ de $R(h)$ em ordem crescente, $i < j \implies h_i < h_j$.

Linhas 8-13: Removemos de $f(n)$ e $g(n)$ os fatores $\mathrm{mdc}(f(n), g(n+h))$ e $\mathrm{mdc}(f(n-h), g(n))$, respectivamente, onde $h \in \{h_1, \cdots, h_N\}$. Estes fatores que foram removidos são igualmente compensados na razão $c(n+1)/c(n)$, na linha 16.

O objetivo, atingido ao final da linha 13, é que $\mathrm{mdc}(p_N(n), q_N(n+h)) = 1$.

Linhas 14-17: A fatoração buscada é

$$\underbrace{\frac{Z\cdot p_N(n)}{q_N(n)}}_{=a(n)/b(n)}\cdot\underbrace{s_N(n-1)s_N(n-2)s_N(n-3)\cdots s_N(n-h_N)}_{=c(n+1)/c(n)},$$

onde $s_N(n-1),\cdots,s_N(n-h_N)$ são os fatores comuns de $f(n)$ e $g(n)$ que outrora proibiam (5.2.3).

Note que, por definição de p_j, q_j e s_j,

$$\mathrm{mdc}(p_k(n),q_k(n+h_k)) \quad = \quad \mathrm{mdc}\left(\frac{p_{k-1}(n)}{s_k(n)},\frac{q_{k-1}(n+h_k)}{s_k(n)}\right) \quad = \quad 1, \tag{5.3.2}$$

para todo $1\leqslant k\leqslant N$.

Por fim, devolvemos a tripla $(a(n),b(n),c(n))$.

A corretude de $\mathrm{gosperabc}$ permanece intacta quando as raízes $h_1,\cdots,h_N$ não são ordenadas. Entretanto, quando $h_1<h_2<\cdots<h_N$, o termo $c(n)$ gerado tem o menor grau possível dentre todas as fatorações (5.2.2) que satisfazem (5.2.3). Assim, o tamanho do sistema linear de (5.2.6) torna-se menor e é resolvido mais rapidamente.

Teorema 5.3.1 ((PETKOVsEK; WILF; ZEILBERGER, 1996), proposição 5.3.1)**.** *Sejam* $0\leqslant k\leqslant i,j\leqslant N$*,* $h\in\mathbb{N}$ *e* $h<h_{k+1}$*. Então,*

$$\mathrm{mdc}(p_i(n),q_j(n+h))=1. \tag{5.3.3}$$

Prova. Como $p_i(n)\,|\,f(n)$ e $q_j(n)\,|\,g(n)$, segue que

$$\mathrm{mdc}(p_i(n),q_j(n+h))\mid\mathrm{mdc}(f(n),g(n+h)). \qquad (h\geqslant 0)$$

A prova é dividida em dois casos: (i) quando $h\in\mathbb{N}$ não é uma raíz de $R(h)$; (ii) quando $h\in\mathbb{N}$ é uma raíz de $R(h)$.

i Se $h\in\mathbb{N}$ não é uma raíz de $R(h)$, então $R(h)\neq 0$ e, logo, $\mathrm{mdc}(f(n),g(n+h))=1$.

ii Este trecho é propriamente uma prova por indução em k. Inicialmente, temos que $h_1<h_2<\cdots<h_N$.

Passo. Suponha que (5.3.3) valha para $h<h_k$. Resta mostrar que $h=h_k$. Como $p_i(n)\,|\,p_k(n)$ e $q_j(n)\,|\,q_k(n)$, segue que

$$\mathrm{mdc}(p_i(n),q_j(n+h_k))\mid\mathrm{mdc}(p_k(n),q_k(n+h_k)).$$

De (5.3.2), temos que

$$\mathrm{mdc}(p_k(n),q_k(n+h_k)) \quad = \quad 1,$$

o que completa a prova.

Base. Para $k = 0$, não há uma raíz de $R(h)$ tal que $h < h_1$.

$\square$

Teorema 5.3.2 ((PETKOVsEK; WILF; ZEILBERGER, 1996), teorema 5.3.1)**.** *Seja $\mathbb{K}$ um corpo de característica 0 e $r \in \mathbb{K}(n)$ uma função racional não-nula. Então, existem polinômios $a, b, c \in \mathbb{K}(n)$ tais que b e c são mônicos e*

$$r(n) = \frac{a(n)}{b(n)} \frac{c(n+1)}{c(n)}, \tag{5.3.4}$$

onde

(i) $\mathrm{mdc}(a(n), b(n+h)) = 1$*, para todo h natural*

(ii) $\mathrm{mdc}(a(n), c(n)) = 1$

(iii) $\mathrm{mdc}(b(n), c(n+1)) = 1.$

Estes polinômios são gerados por gosperabc*.*

Prova. Sejam $a(n)$, $b(n)$ e $c(n)$ os polinômios gerados por gosperabc.

(i) Tome $i = j = k = N$ em (5.3.3).

(ii) Se $a(n)$ e $c(n)$ compartilham um fator não-constante, então $p_N(n)$ também compartilha um fator não-constante com $s_i(n-j)$, para algum $1 \leqslant i \leqslant N$ e $1 \leqslant j \leqslant h_i$.

Por definição, $q_{i-1}(n + h_i - j) = q_i(n + h_i - j)s_i(n-j)$, então $p_N(n)$ e $q_{i-1}(n + h_i - j)$ compartilham de um fator comum. Como $h_i - j <_i$, isto contradiz o teorema 5.3.1.

Logo, $a(n)$ e $c(n)$ são primos entre si.

(iii) Se $b(n)$ e $c(n+1)$ compartilham um fator comum não-constante, então o mesmo acontece com $q_N(n)$ e $s_i(n-j)$, para algum $1 \leqslant i \leqslant N$ e $0 \leqslant j < h_i$.

Por definição, $p_i(n-j) = p_i(n-j)s_i(n-j)$, então $p_{i-1}(n)$ e $q_N(n+j)$ compartilham de um fator comum. Como $j < h_i$, isto contradiz o teorema 5.3.1.

Logo, $b(n)$ e $c(n+1)$ são primos entre si.

$\square$

Teorema 5.3.3 ((PETKOVsEK; WILF; ZEILBERGER, 1996), lema 5.3.1)**.** *Seja $\mathbb{K}$ um corpo de característica 0 e sejam $a, b, c, A, B, C \in \mathbb{K}(n)$ polinômios tais que*

$$\mathrm{mdc}(a(n), c(n)) = \mathrm{mdc}(b(n), c(n+1)) = \mathrm{mdc}(A(n), B(n+h)) = 1. \qquad (h \geqslant 0)$$

Se

$$\frac{a(n)}{b(n)}\frac{c(n+1)}{c(n)} \quad = \quad \frac{A(n)}{B(n)}\frac{C(n+1)}{C(n)}, \tag{5.3.5}$$

então $c(n)$ *divide* $C(n)$.

Prova. Sejam

$$\begin{aligned} g(n) &= \mathrm{mdc}(c(n), C(n)) \\ d(n) &= c(n) \;/\; g(n) \\ D(n) &= C(n) \;/\; g(n). \end{aligned}$$

Então,

$$\mathrm{mdc}(d(n), D(n)) \;=\; \mathrm{mdc}(a(n), d(n)) \;=\; \mathrm{mdc}(b(n), d(n+1)) \;=\; 1.$$

Reescrevendo (5.3.5) como

$$A(n)\,b(n)\,c(n)\,C(n+1) \quad = \quad a(n)\,B(n)\,C(n)\,c(n+1)$$

e cancelando $g(n)\,g(n+1)$ em ambos os termos, o resultado é

$$A(n)\,b(n)\,d(n)\,D(n+1) \quad = \quad a(n)\,B(n)\,D(n)\,d(n+1),$$

que mostra-nos que

$$\begin{aligned} d(n) &\text{ divide } B(n)\,d(n+1) \\ d(n+1) &\text{ divide } A(n)\,d(n). \end{aligned}$$

Usando estas duas últimas relações, cálculos simples mostram que

$$\begin{aligned} d(n) &\text{ divide } B(n)B(n+1)\cdots B(n+k-1)d(n+k) \\ d(n) &\text{ divide } A(n-1)A(n-2)\cdots A(n-k)d(n-k). \end{aligned} \qquad (k \in \mathbb{N})$$

Como $\mathbb{K}$ tem característica 0, então

$$\mathrm{mdc}(d(n), d(n+k)) \;=\; \mathrm{mdc}(d(n), d(n-k)) \;=\; 1,$$

para todo inteiro k suficientemente grande.

Segue, então, que $d(n)$ divide

$$B(n)\,B(n+1)\,\cdots B(n+k-1) \quad \text{e} \quad A(n-1)\,A(n-1)\,\cdots A(n-k),$$

para todo inteiro k suficientemente grande. Entretanto, estes polinômios são primos entre si e, logo, $d(n)$ é uma constante.

Analogamente, $c(n)$ divide $C(n)$. □

Teorema 5.3.4 ((PETKOVsEK; WILF; ZEILBERGER, 1996), corolário 5.3.1)**.** *A fatoração de (5.3.4) descrita no teorema (5.3.2) é única.*

Prova. Suponha que

$$r(n) = \frac{a(n)}{b(n)}\frac{c(n+1)}{c(n)} = \frac{A(n)}{B(n)}\frac{C(n+1)}{C(n)},$$

onde os polinômios a,b,c,A,B,C satisfazem as propriedades (i), (ii) e (iii) e b,c,B,C são mônicos. Pelo teorema 5.3.3, $c(n)$ divide $C(n)$ e vice-versa. Como eles são ambos mônicos, então resta que $c(n) = C(n)$.

Portanto, $A(n)\,b(n) = a(n)\,B(n)$. Pelo item (i), $b(n)$ divide $B(n)$ e vice-versa. Como $b(n)$ e $B(n)$ são mônicos, então $b(n) = B(n)$.

Na mesma linha de raciocínio, $a(n) = A(n)$. □

5.4 Exemplos

Exemplo ((PETKOVsEK; WILF; ZEILBERGER, 1996), p.78). Considere a soma $S_N = \sum_{0 \leqslant n \leqslant N} (4n+1)\frac{n!}{(2n+1)!}$ cujo somando,

$$t_n \stackrel{\text{def.}}{=} (4n+1)\frac{n!}{(2n+1)!},$$

é um termo hipergeométrico. A razão discreta, t_{n+1}/t_n, pode ser obtida com

```
(%i1):= t(n):=(4*n+1)*n!/((2*n+1)!);
(%i2):= factor(ratsimp(minfactorial(t(n+1)/t(n))));
```

$$\frac{4n+5}{2(2n+3)(4n+1)}$$

Posto na forma hipergeométrica, é o termo

$$\frac{t_{n+1}}{t_n} = \frac{n+5/4}{4(n+3/2)(n+1/4)}$$

em que os parâmetros $a_1 = 5/4$, $b_1 = 3/2$, $b_2 = 1/4$ e $x = 1/4$ servem para determinar $a(n)$, $b(n)$ e $c(n)$ na fatoração única (5.2.2), que são

$$a(n) = 1 \qquad b(n) = 2(2n+3) \qquad c(n) = 4n+1.$$

Resolvendo a recorrência[1]

$$a(n)x(n+1) - b(n-1)x(n) - c(n) = 0,$$

[1] De (5.2.5).

encontramos que $x(n) = -1$, para todo n natural,

```
(%i1) := a(n):=1;
(%i2) := b(n):=2*(2*n+3);
(%i3) := c(n):=4*n+1;
(%i4) := x(n):=-1;
(%i5) := ratsimp(a(n)*x(n+1)-b(n-1)*x(n)-c(n));
```

$$0$$

Então, o $z(n)$ que satisfaz (5.1.1) é

$$\frac{b(n-1)}{c(n)}x(n)t_n = -2\frac{n!}{(2n)!}$$

e resta que

$$S_N = z(n+1)-z(0) = 2\left(1-\frac{(n+1)!}{(2(n+1))!}\right),$$

que é a resposta,

```
(%i1) := load(zeilberger);
(%i2) := z(n):=-2*n!/((2*n)!);
(%i3) := S_N : minfactorial(z(n+1)-z(0));
(%i4) := ratsimp(minfactorial(ratsimp(
            GosperSum(t(k),k,0,n) - S_N)));
```

$$0$$

Exemplo (Usando MAXIMA). Considere o termo hipergeométrico $t(n) = (-1)^k\binom{n}{2}$.

A anti-diferença $z(n)$ que satisfaz $z(n+1) - z(n) = t(n)$ é

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

verificado no MAXIMA:

```
(%i1) := load(zeilberger);
(%i2) := t(n):=(-1)^n * binomial(n, 2);
(%i3) := z(n):=AntiDifference(t(n),n);
(%i4) := ratsimp(z(n));
```

$$-\frac{(2n^2-4n+1)(-1)^n}{8}$$

5.5 Implementações e considerações finais

Alguns sistemas de computação algébrica implementam nas suas rotinas de soma simbólica o algoritmo de Gosper extendido, uma generalização simples para múltiplos termos hipergeométricos de entrada, como é o caso do MAXIMA, MATHEMATICA e MAPLE.

Em especial, MAXIMA conta com duas implementações:

nusum no pacote nusum: Resolve uma soma definida calculando a anti-diferença e a aplicando nos limites dados. Este pacote foi originalmente implementado em Macsyma pelo próprio Gosper Jr.

GosperSum e **AntiDifference** no pacote zeilberger: **GosperSum** é similar ao ***nusum*** mas implementa uma generalização do algoritmo de Gosper, enquanto **AntiDifference** corresponde à implementação de gosper. Este pacote foi escrito por Fabrizio Caruso.

Exemplos (***nusum*** *no pacote nusum*). Eis um exemplo positivo

```
(%i1) := load(nusum);
(%i2) := nusum((-1)^k * k/(4*k^2 - 1), k, 1, n);
```

$$-\frac{\left(n+\frac{3}{2}\right)(-1)^{n+1}}{2\,(4(n+1)-1)}-\frac{1}{4}$$

e um exemplo negativo

```
(%i1) := nusum((-1)^k * k/(2*k^2 - 1), k, 1, n);
```

$$\sum_{k=1}^{n}\frac{k\,(-1)^k}{2\,k^2\;-\;1}$$

Exemplos (***GosperSum*** *no pacote zeilberger*). Eis um exemplo positivo

```
(%i1) := load(zeilberger);
(%i2) := GosperSum((-1)^k * k/(4*k^2 - 1), k, 1, n);
```

$$-\frac{\left(n+\frac{3}{2}\right)(-1)^{n+1}}{2\,(4(n+1)-1)}-\frac{1}{4}$$

e um exemplo negativo

```
(%i1) := GosperSum((-1)^k * k/(2*k^2 - 1), k, 1, n);
```

$$NON\_GOSPER\_SUMMABLE$$

Exemplos (**AntiDifference** no pacote zeilberger). Os termos $z(n)$ nos nossos últimos exemplos:

```
(%i1) := t(n):=(-1)^n * n/(4*n^2 - 1);
(%i2) := z(n):=AntiDifference(t(n),n);
(%i3) := z(n);
```

$$-\frac{(n+\frac{1}{2})(-1)^n}{2(4n^2-1)}.$$

```
(%i1) := t(n):=(-1)^n * n/(2*n^2 - 1);
(%i2) := AntiDifference(t(n),n);
```

$$NO\_HYP\_ANTIDIFFERENCE$$

A última resposta do MAXIMA conta-nos que não há solução hipergeométrica para o referido exemplo.

6 Decisão de algumas identidades com WZ

O algoritmo de WZ (PETKOVsEK; WILF; ZEILBERGER, 1996) presta-se a resolver algumas identidades da forma $\sum_k t(n,k) = s(n)$, onde $t(n,k)$ é duplamente hipergeométrico e $s(n)$ é um termo hipergeométrico, fornecendo provas muito sucintas.

Apresentamos a descrição do problema com detalhes, definimos os famosos pares WZ e seus certificados e, ainda, apresentamos o algoritmo WZ acomapanhado de um exemplo com uma aplicação em MAXIMA.

6.1 Descrição do problema e do algoritmo

Considere o problema de decidir se uma equação da forma

$$\sum_k t(n,k) \quad = \quad s(n), \qquad (n \geqslant 0)$$

onde $s(n)$ é um termo hipergeométrico em n e $t(n,k)$ é duplamente hipergeométrico em (n,k). O algoritmo WZ resolve este problema de decisão em grande parte dos casos, não em todos; esta garantia será tratada posteriormente.

A forma de WZ provar esta identidade é, segundo Wilf e Zeilberger, normalizar a soma, $S(n) \stackrel{\text{def.}}{=} \sum_k F(n,k)$, onde

$$F(n,k) \quad = \quad \begin{cases} t(n,k)\,/\,s(n), & \text{se } s(n) \neq 0 \\ t(n,k), & \text{se } s(n) = 0, \end{cases} \tag{6.1.1}$$

e provar que $S(n)$ é uma constante. É imediato que se $s(n) \neq 0$, então esta constante deve ser 1; caso contrário, deve ser 0.

Uma prova suficiente é mostrar que $S(n)$ é uma constante e que esta constante é 0 ou 1, de acordo com $s(n)$ ser ou não ser nulo.

Aqui, recorremos à notação de derivação discreta, dada no apêndice (A). Para mostrar que $S(n)$ é uma constante, basta mostrar que sua derivada discreta, $\Delta S(n) \equiv S(n+1) - S(n)$, é nula. Uma forma de provar que $\Delta S(n) = 0$ é encontrar um termo hipergeométrico $G(n,k)$, com $\lim_{k\to\pm\infty} G(n,k) = 0$, satisfazendo

$$F(n+1,k) - F(n,k) \quad = \quad G(n,k+1) - G(n,k) \tag{6.1.2}$$

e somar (6.1.2) sobre todos os k inteiros, restando que $s(n+1) - s(n) = 0$. Este par de funções é dito *par WZ*.

Definição (Par WZ). Um par de funções (F,G) é um *par WZ* se e somente se

$$F(n+1,k)-F(n,k) \;=\; G(n,k+1)-G(n,k) \quad \text{e} \quad \lim_{k\to\pm\infty} G(n,k) \;=\; 0.$$

Sob a hipótese de que $\lim_{k\to\pm\infty} G(n,k)=0$, ocorre que

$$\begin{aligned}\sum_k (F(n+1,k) - F(n,k)) &= \sum_k (G(n,k+1) - G(n,k)) \\ &= 0\end{aligned}$$

e, consequentemente,

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

i.e., $\sum_k F(n,k)$ independe de n e, portanto, trata-se de uma constante. Isto é suficiente para decidir o problema.

Como dito, nem sempre WZ prova todas as identidades da forma supracitada; quando ele prova, é o caso que existe um $G(n,k)$ que satisfaz (6.1.2). Para encontrar este $G(n,k)$ usamos o algoritmo de Gosper, como veremos.

6.2 O algoritmo WZ

Antes de apresentarmos o algoritmo WZ, precisamos da noção de certificado de um par WZ e um algoritmo para o obter. O *certificado* $R(n,k)$ de um par (F,G) advém de

$$G(n,k) \;=\; R(n,k)\,F(n,k) \tag{6.2.1}$$

e, então, defini-mo-lo como $R(n,k) \stackrel{\text{def.}}{=} G(n,k)\,/\,F(n,k)$. Também é cabível dizer que vamos procurar um certificado para F. Ora pois, dado F, a obtenção de G é imediata a partir de R, se tal certificado existir.

Algoritmo 6: certificado$(F(n,k))$

Entrada: $F(n,k)$ como em (6.1.1)
Saída: $R(n,k)$, se existir; "não", caso contrário.

1 $f(k) \leftarrow F(n+1,k)-F(n,k)$; /* $\Delta_n F(n,k)$ */
2 $G(n,k) \leftarrow gosper(f,k)$; /* anti-diferença de $\Delta_n F(n,k)$ */
3 **se** *o algoritmo de Gosper falhar* **então** **retorna** *"não"*
4 **retorna** $G(n,k)\,/\,F(n,k)$; /* $R(n,k)$ */

Procedemos com uma breve descrição deste algoritmo.

Linha 1: $f(k) = \underset{n}{\Delta} F(n,k)$ faz o papel do termo da esquerda em (6.1.2).

Linha 2: Procuramos uma anti-diferença $g(k) \equiv G(n,k)$ que satisfaça

$$f(k) = g(k+1) - g(k).$$

Recordando-nos do algoritmo de Gosper, $f(k)$ e $g(k)$ são mapeados como $t(k)$ e $z(k)$ em (5.1.1).

Linhas 3-4: Se não houver uma anti-diferença $g(k)$, a obtenção do certificado não é possível e resta-nos apenas devolver “não”. Caso contrário, o que temos é uma anti-diferença hipergeométrica $g(k) = G(n,k)$ e procedemos devolvendo o $R(n,k)$ correspondente, como em (6.2.1).

Exemplo (Obtenção do certificado). Vamos encontrar um certificado para $F(n,k) = \binom{n}{k}/2^n$. Tomando $f(k) = \Delta_n F(n,k)$ e fornecendo-o como entrada para o algoritmo de Gosper, $\mathrm{gosper}(f(k), k)$, temos que

$$G(n,k) = -\frac{k}{2^k} \quad \text{se e somente se} \quad R(n,k) = -\frac{k\,2^{n-k}}{\binom{n}{k}}.$$

```
(%i1):= F(n,k) := binomial(n,k) / 2^k;
(%i2):= f(k) := ratsimp(makefact(F(n+1,k) - F(n,k)));
(%i3):= G(n,k) := gosper(f, k);
(%i4):= R(n,k) := G(n,k) / F(n,k);
```

$$-\frac{k\,2^{n-k}}{\binom{n}{k}}$$

Nosso pacote também traz uma implementação homônima de certificado.

```
(%i1):= F(n,k) := binomial(n,k) / 2^n;
(%i2):= certificado(F,n,k);
```

Exemplo.

$$-\frac{k\,2^{n-k}}{\binom{n}{k}}$$

Eis, finalmente, o algoritmo WZ.

Algoritmo 7: $\mathrm{wz}\,(t(n,k),\ s(n))$

Entrada: $t(n,k)$ duplamente hipergeométrico e $s(n)$ hipergeométrico.
Saída:

$$\begin{cases} \text{“não consegui provar”}, & \text{se a prova } \textit{não} \text{ foi bem sucedida;} \\ \text{“não”}, & \text{se } \sum_k t(n,k) \neq s(n); \\ \text{“sim”}, & \text{se } \sum_k t(n,k) = s(n). \end{cases}$$

1 $F(n,k) \leftarrow \begin{cases} t(n,k)\ /\ s(n), & \text{se } s(n) \neq 0 \\ t(n,k), & \text{se } s(n) = 0 \end{cases}$
2 $R(n,k) \leftarrow \mathrm{certificado}\,(F(n+1,k) - F(n,k),\ k)$
3 **se** *o certificado falhou* **então** **retorna** *“não consegui provar”*
4 $G(n,k) \leftarrow R(n,k)F(n,k)$
5 **se** $F(n+1,k) - F(n,k) \neq G(n,k+1) - G(n,k)$ **então** **retorna** *“não”*
6 **se** $(F(0,k) = 0 \wedge s(n) = 0) \vee (F(0,k) = 1 \wedge s(n) \neq 0)$ **então** **retorna** *“sim”*
7 **retorna** *“não”*

A seguir, uma descrição de wz.

Linha 1: Formamos um termo duplamente hipergeométrico $F(n,k)$ de acordo com $s(n)$ ser ou não ser nulo.

Linhas 2-3: Buscamos um certificado na variável k para $\Delta_n F(n,k)$; se a obtenção do certificado falhar, então o WZ também falha em provar a identidade e devolve “não consegui provar”, já na terceira linha.

Linhas 4-5: A partir deste ponto, WZ já pode discriminar a identidade como sendo ou não uma tautologia. Aqui, o par (F,G) é formado e, na linha 5, checamos se a identidade vale.

Para decidir esta igualdade, basta colocar ambos os termos sobre um mesmo denominador, simplificar os fatoriais e considerar apenas os numeradores, que são polinômios; assumimos que a tarefa de decidir se dois polinômios são idênticos é uma trivialidade.

Linhas 6-7: Decidimos por “sim” se e somente se

$$\begin{cases} F(0,\,k) = 0, & \text{se } s(n) = 0 \\ F(0,\,k) = 1, & \text{se } s(n) \neq 0. \end{cases} \tag{6.2.2}$$

Na verdade, poderíamos testar $F(n,k) = const$ em qualquer ponto n. A escolha por $n = 0$ é arbitrária e não há uma justificativa definitiva de que este seja melhor do que outro n qualquer. Atente que $const$ deve ser conforme (6.2.2).

6.3 Exemplos

Exemplo (Binomial). Vamos provar a famosa $\sum_k \binom{n}{k} = 2^n$.

O primeiro passo, segundo Wilf e Zeilberger, é normalizar o termo hipergeométrico para provar $\sum_k F(n,k) = 1$, onde $F(n,k) = \binom{n}{k} 2^{-n}$.

Um certificado para F, obtido via $\mathrm{gosper}(F(n+1) - F(n,k), k)$, é $R(n,k) = \frac{k}{2(k-n-1)}$. Assim, formamos a contra-parte de F no par WZ,

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

Agora, com a ajuda do MAXIMA,

```
(%i1):= F(n,k) := n!/(k!*(n-k)!)*2^(-n);
(%i2):= R(n,k) := k/(2*(k-n-1));
(%i3):= G(n,k) := F(n,k)*R(n,k);
(%i4):= ratsimp(minfactorial(F(n+1,k)-F(n,k)-G(n,k+1)+G(n,k)));
```

$$0$$

verificamos que

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

Finalmente, como $\sum_k F(0,k) = 1$, então resta-nos que $\sum_k \binom{n}{k} = 2^n$.

Nosso pacote ainda acompanha alguns exemplos de provas de identidades com o algoritmo WZ.

7 De Celine para Zeilberger

O algoritmo de Zeilberger resolve o problema da soma hipergeométrica definida encontrando uma recorrência para um termo duplamente hipergeométrico. Este algoritmo substitui o algoritmo da irmã Celine porque é mais eficiente em relação ao tempo.

Apresentamos, neste capítulo, o problema, o algoritmo e alguns exemplos.

7.1 Descrição do problema

Nosso problema é resolver uma soma da forma $f(n) \stackrel{\text{def.}}{=} \sum_k F(n,k)$, onde $F(n,k)$ é duplamente hipergeométrico em (n,k).

Buscamos uma recorrência da forma

$$\sum_{j=0}^{J} a_j(n) F(n+j,k) \quad = \quad G(n,k+1) - G(n,k), \tag{7.1.1}$$

onde $G(n,k)/F(n,k)$ é uma função racional em (n,k).

Como vimos no algoritmo WZ, $G(n,k)$ forma, junto com $F(n,k)$, um par WZ, (F,G). Relembrando, a função $R(n,k) = G(n,k)/F(n,k)$ é um certificado para $F(n,k)$.

Antes de tudo, reescrevemos o problema de outro modo. Precisamos apresentar a notação para operador de deslocamento de variável, um para cada variável do termo duplamente hipergeométrico.

Definição (operador N).

$$\begin{aligned} N^i\, g(n,\, k) \quad &\stackrel{\text{def.}}{=} \quad \begin{cases} g(n,\, k), & \text{se } i = 0 \\ N^{i-1} g(n+1,\, k), & \text{se } i > 0. \end{cases} \\ &= \quad g(n+i,\, k), \text{ se } i \text{ é natural} \end{aligned}$$

Definição (operador K).

$$\begin{aligned} K^i\, g(n,\, k) \quad &\stackrel{\text{def.}}{=} \quad \begin{cases} g(n,\, k), & \text{se } i = 0 \\ K^{i-1} g(n,\, k+1), & \text{se } i > 0 \end{cases} \\ &= \quad g(n,\, k+i), \text{ se } i \text{ é natural.} \end{aligned}$$

Por exemplo, a recorrência de WZ, em termos dos operadores N e K, é

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

se e somente se

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

Com estes operadores, o problema é encontrar, então, uma combinação linear de $\{1, N, N^2, \cdots, N^J\}$ cujos coeficientes sejam polinômios em n,

$$p(n,N) = a_0(n) + a_1(n)N + a_2(n)N^2 + \cdots + a_J(n)N^J,$$

que satisfaça

$$p(n,N)\,F(n,k) = (K-1)\,G(n,k); \tag{7.1.2}$$

esta é uma forma de reescrever (7.1.1) de modo que se possa buscar uma combinação linear, que é mais simples de se trabalhar.

Eis, então, nosso problema.

instância: Um termo duplamente hipergeométrico $F(n,k)$.

resposta: A combinação linear $p(n,N)$ de (7.1.2).

A tarefa do algoritmo de Zeilberger resume-se a encontrar esta combinação linear. Esta combinação linear permite montar a recorrência (7.1.1), que satisfaz $f(n)$. Ela ajuda a resolver $f(n)$ do mesmo modo que a resposta de celineIJ ajuda; soma-se (7.1.1) sobre todos os k inteiros,

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

se e somente se

$$\sum_k a_j(n)\,f(n+j) = 0.$$

Teorema 7.1.1. *[Existência de uma recorrência livre de k, (PETKOVsEK; WILF; ZEILBERGER, 1996, p.105)] Seja $F(n,k)$ um termo duplamente hipergeométrico em (n,k). Então, F satisfaz uma recorrência não trivial da forma*

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

onde $G(n,k)/F(n,k)$ é uma função racional em n e k.

Demonstração. Considere a recorrência

$$\sum_{i=0}^{I}\sum_{j=0}^{J} a_{i,j}(n)\,F(n+j,\,k+i) \quad = \quad 0, \tag{7.1.3}$$

que, segundo o teorema fundamental, existe. Usando nossos operadores N e K, podemos reescrever (7.1.3) da forma $P(N,\,n,\,K)\,F(n,\,k) \;=\; 0$, que é trivialmente demonstrável.

Considere, agora, o polinômio $P(u,\,v,\,w)$ e faça a expansão em série de potências na variável w em torno do ponto $w=1$,

$$P(u,\,v,\,w) \quad = \quad P(u,\,v,\,1)\;+\;(1-w)\,Q(u,\,v,\,w), \tag{7.1.4}$$

onde Q é um polinômio. Então, (7.1.3) implica

$$0 \;=\; P(N,n,K)F(n,k) \;=\; (P(N,n,1)+(1-K)Q(N,n,K))F(n,k)$$

e, aplicando (7.1.4), temos

$$P(N,\,n,\,1)\;F(n,\,k) \quad = \quad (K-1)\;Q(N,\,n,\,K)\;F(n,k). \tag{7.1.5}$$

No termo da esquerda de (7.1.5), apenas o operador N está presente, então não teremos anti-diferença em k, apenas em n. Se tomarmos

$$G(n,\,k) \quad := \quad Q(N,\,n,\,K)\;F(n,k),$$

então podemos reescrever o termo da direita de (7.1.5) como $G(n,\,k+1) \;-\; G(n,\,k)$; como $Q(N,\,n,\,K)$ contém os dois operadores em graus arbitrários, resta que $G(n,\,k)$ é uma função racional em $(n,\,k)$ multiplicada por $F(n,k)$.

Agora, assumimos que (7.1.5) não é uma recorrência trivial. (A prova completa desta proposição é dada em (GRAHAM; KNUTH; PATASHNIK, 1994).) O argumento é que existe um polinômio em N e K de grau suficientemente alto que pode ser ajustado à recorrência e nos fornece pelo menos um $P(N,\,n,\,K)$ não trivial que elimina $F(n,k)$.

Seja $P=P(N,\,n,\,K)$ um destes $P(N,\,n,\,K)$'s com grau pelo menos K. Como $K-1$ é um fator de P, temos que

$$P(N,\,n,\,K) \quad = \quad P(N,\,n,\,1)\;-\;(K-1)\;Q(N,\,n,\,K),$$

que podemos tomar como definição do operador Q.

Suponha que $P(N,\,n,\,1) \;=\; 0$. Então, $(K-1)\,G(n,k) \;=\; 0$; i.e., G é independente de k. Portanto, $G(n,k)$ é um termo hipergeométrico na variável n; $G(n,k)$ satisfaz uma recorrência de primeira ordem com coeficientes polinomiais em n, como esperado. Logo, existe um operador $H(N,\,n)$ satisfazendo

$$H(N,\,n)\;G(n,\,k) \quad = \quad 0.$$

Agora temos dois casos. Em (7.1.5),

- ⋆ Se $Q = 0$, então $P(N, n, K) \;=\; P(N, n, 1)$ é um operador não-nulo e livre de k que é independente de (k, K) e, também, ele elimina $F(n,k)$.

- ⋆ Se $Q \neq 0$, então $H(N, n)\ Q(N, n, K)$ é um operador não-nulo livre de k que, trivialmente, elimina $F(n,k)$.

Em ambos os casos, temos um operador que elimina $F(n, k)$ cujo grau de K é menor que $P(N, n, K)$. Isso deriva uma contradição porque P tem grau pelo menos K. □

7.2 O algoritmo de Zeilberger

Visto o teorema (7.1.1), o algoritmo de Zeilberger é bastante curto: primeiro, obtemos um certificado $R(n,k)$ para $F(n,k)$ formando $(F(n,k),\ F(n,k)R(n,k))$, um par WZ. Depois, tentamos, para $J = 0, 1, \cdots$, ajustar um operador $p(n, N)$ à recorrência de (7.1.2). Por fim, devolvemos o ajuste $p(n, N)$ com o menor grau de N.

Algoritmo 8: $\mathrm{zeilberger}\,(F(n,k))$

Entrada: $F(n, k)$.
Saída: $p(n, N)$.

$R(n, k) \leftarrow \mathrm{certificado}\ (F(n, k))$
$G(n, k) \leftarrow R(n,k)\ F(n,k)$
para cada $j \in \mathbb{N}$ **faça**
  Tente ajustar $p(n, N) \;=\; \sum_{j=0}^{J} a_j(n) N^j$ em

$$p(n, N)\ F(n, k) \quad = \quad (K - 1)\ G(n, k)$$

  se *o ajuste for possível* **então** **retorna** *p(n, N)*

Assim,

$$F(n,k) \cdot \mathrm{zeilberger}\,(F(n,k)) \quad = \quad 0$$

é a recorrência que $F(n,k)$ satisfaz.

7.3 Exemplos

Exemplo (Quadrado da binomial). Para

$$f(n) \quad \stackrel{\text{def.}}{=} \quad \sum_k \underbrace{\binom{n}{k}^2}_{=F(n,k)}$$

encontramos um operador de ordem $J=1$,

$$p(n,N) \quad = \quad -4n-2 \;+\; (n+1)N,$$

com o certificado

$$R(n,k) \;=\; \frac{(-3n-3+2k)\,k^2}{(k-n-1)^2}.$$

Noutras palavras, $F(n,k)$ satisfaz

$$-2(2n+1)F(n,k) \;+\; (n+1)F(n+1,k) \quad = \quad G(n,k+1) \;-\; G(n,k), \tag{7.3.1}$$

onde

$$G(n,k) \quad = \quad F(n,k)R(n,k) \quad = \quad \frac{(-3n-3+2k)\,k^2}{(k-n-1)^2}\binom{n}{k}^2.$$

Para resolver $f(n)$, é necessário somar (7.3.1) sobre todos os k inteiros,

$$(-4n-2)\,f(n) \;+\; (n+1)\,f(n+1) \quad = \quad 0,$$

que resulta numa forma fechada para $f(n)$,

$$f(n) \quad = \quad \frac{\left(\frac{3}{2}\right)^{\overline{n-1}}}{2^{\overline{n-1}}}\,2^{2n-1} \quad = \quad \frac{4^n}{\sqrt{\pi}}\frac{\Gamma(n+\frac{1}{2})}{\Gamma(n+1)}.$$

Exemplo (Média da binomial). Para $\sum_k k\binom{n}{k}$, temos

$$\begin{aligned} p(n,N) &= -2n-2+nN \\ R(n,k) &= \frac{(n+1)(k-1)}{k-n-1}. \end{aligned}$$

Exemplo. Para

$$\sum_k (-1)^k k\binom{n}{k}\binom{n+k-1}{k},$$

temos

$$\begin{aligned} p(n,N) &= n \;+\; 1 \;+\; nN \\ R(n,k) &= \frac{(2n+1)\,(k-1)\,k}{(k-n-1)\,n}. \end{aligned}$$

7.4 Implementação em MAXIMA

No MAXIMA, a função **Zeilberger** do pacote zeilberger implementa uma variação do algoritmo de Zeilberger. Ela devolve um conjunto de soluções (e não somente uma solução). Destas soluções, cada uma delas é um par contendo o certificado $R(n,k)$ e a respectiva lista de coeficientes $a_j(n)$, $1 \leqslant j \leqslant J$.

```
(%i1) := load(zeilberger);
(%i2) := Zeilberger(binomial(n,k)^2,k,n);
```

$$[[-\frac{k^2(3n-2k+3)}{(n-k+1)^2},[-2(2n+1),n+1]]]$$

```
(%i1) := Zeilberger(k*binomial(n,k),k,n);
```

$$[[\frac{(k-1)(n+1)}{n-k+1},[2(n+1),-n]]]$$

```
(%i1) := Zeilberger((-1)^k * k * binomial(n,k) * binomial(n+k-1,k),
            k, n);
```

$$[[\frac{(k-1)k(2n+1)}{n(n-k+1)},[-(n+1),-n]]]$$

Nosso pacote contém uma função zeilbergerrec, que, dado um termo duplamente hipergeométrico, devolve uma recorrência na forma (7.1.1).

```
(%i1) := F(n,k) := binomial(n,k)^2;
(%i2) := zeilbergerrec(F,n,k);
```

$$(n+1)F_{n+2,k}-2(2n+1)F_{n+1,k} = \frac{k^2(3n-2k+3)\binom{n}{k}^2}{(n-k+1)^2}-\frac{k^2(3n-2k+3)\binom{n}{k+1}^2}{(n-k+1)^2}$$

8 Considerações finais

Neste ponto, ligamos os problemas das somas hipergeométricas definida e indefinida com os problemas da integral definida e indefinida. O algoritmo da irmã Celine resolve o problema da soma hipergeométrica definida e estabelece uma ponte entre o problema da integral definida e da soma [hipergeométrica] definida. O algoritmo de Gosper resolve o problema da soma hipergeométrica indefinida e estabelece uma ponte entre o problema da integral indefinida e da soma [hipergeométrica] indefinida.

Devemos notar que, apesar de haver muitas semelhanças entre identidades contínuas e discretas,

$$\int_a^b x\,\mathrm{d}x = \underbrace{\frac{x^2}{2}}_{F(x)}\Big|_{x=a}^{b} = F(b)-F(a), \qquad \sum_{k=a}^{b-1}\underbrace{k}_{t_k} = \underbrace{\frac{k^2}{2}}_{z_k}\Big|_{k=a}^{b} = z_b - z_a$$

$$\int_0^b x^n\,\mathrm{d}x = \underbrace{\frac{x^{n+1}}{n+1}}_{F(x)}\Big|_{x=0}^{b} = F(b)-F(0), \qquad \sum_{k=0}^{b-1}\underbrace{k^{\underline{n}}}_{t_k} = \underbrace{\frac{k^{\underline{n+1}}}{n+1}}_{z_k}\Big|_{k=0}^{b} = z_b - z_0$$

$$\int x e^x\,\mathrm{d}x = \underbrace{xe^x - e^x}_{F(x)} + O(1), \qquad \sum \underbrace{k2^k}_{t_k} = \underbrace{k2^k - 2^{k+1}}_{z_k} + O(1)$$

$$\int x\log x\,\mathrm{d}x = \underbrace{\frac{x^2}{2}\left(\log x - \frac{1}{2}\right)}_{F(x)} + O(1), \qquad \sum \underbrace{kH_k}_{t_k} = \underbrace{\frac{k^{\underline{2}}}{2}\left(H_k - \frac{1}{2}\right)}_{z_k} + O(1),$$

nem todas as soluções na versão contínua podem ser mapeadas para a classe hipergeométrica, como é o caso de $\int \frac{1}{x}\,\mathrm{d}x = \log x + O(1)$; a sequência dos números harmônicos, $\{H_n\}_{n=0}^{\infty}$, não admite uma forma fechada:

```
(%i1) := load(zeilberger);
(%i2) := AntiDifference(1/k, k);
```

$$NO\_HYP\_ANTIDIFFERENCE$$

Entenda-se que as duas últimas somas são indefinidas sobre a variável k, apenas omitimos a variável para evitar conflito com a notação que viemos usando.

Algumas aplicações interessantes destes algoritmos envolvem funções holonômicas, wavelets e equações bi-basic (PETKOVsEK; WILF; ZEILBERGER, 1996). Ademais, existem algumas generalizações dos algoritmos que apresentamos que são apresentadas em (KAUERS; PAULE, 2012) além daquelas que já comentadas no capítulo 4.5.

Referências

FASENMYER, S. M. C. Some generalized hypergeometric polynomials. *Bull. Amer. Math. Soc*, v. 53, p. 806–812, 1947. Disponível em: <http://www.ams.org/journals/bull/1947-53-08/S0002-9904-1947-08893-5/S0002-9904-1947-08893-5.pdf>. 14, 29, 35 Citado 3 vezes nas páginas 14, 29, and 35.

FLAJOLET, P.; SEDGEWICK, R. *Analytic Combinatorics*. Cambridge University Press, 2009. ISBN 9780521898065. Disponível em: <http://algo.inria.fr/flajolet/Publications/books.html>. 17, 20 Citado 2 vezes nas páginas 17 and 20.

FRÖHLICH, A.; SHEPHERDSON, J. On the factorisation of polynomials in a finite number of steps. *Mathematische Zeitschrift*, Springer-Verlag, v. 62, n. 1, p. 331–334, 1955. ISSN 0025-5874. Disponível em: <http://dx.doi.org/10.1007/BF01180640>. 25 Citado na página 25.

GOSPER, R. W. Decision procedure for indefinite hypergeometric summation. *Proceedings on National Academic Science USA*, v. 75, n. 1, p. 40–42, 1978. 14, 37, 39 Citado 3 vezes nas páginas 14, 37, and 39.

GRAHAM, R. L.; KNUTH, D. E.; PATASHNIK, O. *Concrete Mathematics: A Foundation for Computer Science*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1994. ISBN 0201558025. 59 Citado na página 59.

KAUERS, M.; PAULE, P. *The Concrete Tetrahedron: Symbolic Sums, Recurrence Relations, Generating Functions, Asymptotic Estimates*. [S.l.]: Springer Wien - New York, 2012. 63 Citado na página 63.

LENSTRA, A.; JR, H. L.; LOVASZ, L. Factoring polynomials with rational coefficients. *Mathematische Annalen*, Springer-Verlag, v. 261, n. 4, p. 515–534, 1982. ISSN 0025-5831. Disponível em: <http://dx.doi.org/10.1007/BF01457454>. 25 Citado na página 25.

MATIYASEVICH, Y. Enumerable sets are diophantine. *Doklady Akademii Nauk SSSR*, v. 191, p. 279–282, 1970. 25 Citado na página 25.

PETKOVsEK, M.; WILF, H. S.; ZEILBERGER, D. *A = B*. Wellesley, MA 02482-3717: A K Peters. Ltd., 1996. 29, 35, 37, 39, 40, 43, 44, 45, 46, 51, 58, 63 Citado 12 vezes nas páginas 29, 35, 37, 39, 40, 43, 44, 45, 46, 51, 58, and 63.

VILLATE, J. *Maxima Manual*. [S.l.], 2013. Disponível em: <http://maxima.sourceforge.net/docs-/manual% -/en/maxima.html>. 23 Citado na página 23.

WILF, H. S. *generatingfunctionology*. Wellesley, MA 02482: A K Peters. Ltd., 2005. 17, 20 Citado 2 vezes nas páginas 17 and 20.

ZEILBERGER, D. Sister celine's technique ans its generalizations. *Journal of Mathematical Analysis and Applications*, v. 85, p. 114–145, 1982. 35 Citado na página 35.

Apêndices

APÊNDICE A – Derivada discreta

A derivada discreta de uma função f de domínio inteiro no seu argumento n, $\Delta_n f(n)$, é definida por

$$\Delta_n f(n) \quad \stackrel{\text{def.}}{=} \quad f(n+1) - f(n)$$

e, se esta função tem somente um argumento, chamamos o operador Δ_n simplesmente de Δ.

A recorrência implícita nesta definição,

$$\Delta^k f(n) \quad = \quad \begin{cases} f(n), & \text{se } k = 0 \\ \Delta^{k-1} f(n+1) - \Delta^{k-1} f(n), & \text{se } k \geqslant 1, \end{cases}$$

nos leva, com algumas páginas de demonstração omitidas, à formula de Newton,

$$f(n+p) \quad = \quad \sum_{0 \leqslant k \leqslant n} \binom{n}{k} \Delta^k f(p),$$

que é o análogo discreto da expansão de Taylor em torno do ponto p.

Muito similar ao cálculo infinitesimal, se a e b são constantes e f e g são funções,

$$\Delta a \;=\; 0, \qquad \Delta(af \,+\, bg) \;=\; a\Delta f \,+\, b\Delta g$$

$$\Delta(fg) \;=\; f\Delta g \,+\, g\Delta f \,+\, \Delta f \Delta g, \qquad \Delta\left(\frac{f}{g}\right) \;=\; \frac{g\Delta f \,-\, f\Delta g}{(g+\Delta g)\; g}$$

$$\sum_{n=a}^{b-1} \Delta f(n) \;=\; f(b) - f(a).$$

APÊNDICE B – A resultante de dois polinômios

Suponha que queiramos determinar se dados dois polinômios p e q com coeficientes racionais sobre uma mesma variável têm uma raíz comum.

No caso de $\text{grau}\,(p) \;=\; \text{grau}\,(q) \;=\; 1$,

$$p(x) \;=\; p_0 + p_1 x \qquad\qquad q(x) \;=\; q_0 + q_1 x,$$

basta isolarmos as raízes e as comparar: $p_0 q_1 \overset{?}{=} q_0 p_1$; ou, em forma matricial,

$$\begin{pmatrix} p_0 & p_1 \\ q_0 & q_1 \end{pmatrix} \cdot \begin{pmatrix} x^0 \\ x^1 \end{pmatrix} \;=\; \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Agora, vamos para o caso geral. Sejam p e q dois polinômios univariados de graus m e n não-nulos dados por

$$\begin{aligned} p(x) &= p_0 + p_1 x + \cdots + p_m x^m \\ q(x) &= q_0 + q_1 x + \cdots + q_n x^n. \end{aligned}$$

Definição (Matriz de Sylvester). A matriz de Sylvester associada a p e q, de ordem $(m+n)\times(m+n)$, é obtida da seguinte forma:

1. A primeira linha da matriz é
$$\begin{pmatrix} p_m & p_{m-1} & \cdots & p_1 & p_0 & 0 & \cdots & 0 \end{pmatrix}.$$
2. Nas próximas $n-1$ linhas, desloque em uma célula para a direita os coeficientes dos monômios, deixando as demais células como 0.
3. A partir da $(n+1)$-ésima linha, repita o processo para o polinômio q ao longo das próximas m linhas.

Exemplo ($S_{2,3}$). Quando $m = 2$ e $n = 3$, a matriz de Sylvester é

$$S_{p,q} \;=\; \begin{pmatrix} p_2 & p_1 & p_0 & 0 & 0 \\ 0 & p_2 & p_1 & p_0 & 0 \\ 0 & 0 & p_2 & p_1 & p_0 \\ q_3 & q_2 & q_1 & q_0 & 0 \\ 0 & q_3 & q_2 & q_1 & q_0 \end{pmatrix}.$$

O determinante desta matriz é chamado de *resultante* dos polinômios p e q,

$$\text{resultante}\,(p,\,q) \quad \stackrel{\text{def.}}{=} \quad \det\, S_{p,q}.$$

Para continuar, precisamos de dois conceitos de álgebra linear. Relembrando, o *núcleo* de uma matriz M é o conjunto de vetores x tais que $M \cdot x \;=\; 0$. Também, o *posto* de uma matriz M é o número máximo de colunas (ou, equivalentemente, de linhas) linearmente independentes de M.

Agora, buscamos soluções de

$$S_{p,q}^T \cdot \begin{pmatrix} x \\ y \end{pmatrix} \quad = \quad 0, \tag{B.0.1}$$

onde $|x| = n$ e $|y| = m$, para que os pares (x,y), formando, respectivamente, polinômios de graus $n-1$ e $m-1$, satisfaçam

$$x \cdot p \;+\; y \cdot q \quad = \quad 0,$$

que é um caso particular da identidade de Bézout.

Pela definição de núcleo, o núcleo de $S_{p,q}^T$ nos dá todas as soluções $(x,y)^T$ de (B.0.1). Olhando para os graus dos polinômios e para as dimensões das matrizes envolvidas, estas soluções satisfazem $\text{grau}\,(x) < \text{grau}\,(q)$ e $\text{grau}\,(y) < \text{grau}\,(p)$.

Das propriedades de núcleo em espaços vetoriais de polinômios, temos que

$$\text{grau}\,(\text{mdc}(p,q)) \quad = \quad m+n-\text{posto}\left(S_{p,q}\right);$$

i.e., $\text{posto}\left(S_{p,q}\right)$ determina o grau do máximo divisor comum de p e q.