

Seminarvortrag
Computational Mathematics

Der Fasenmyer-Algorithmus

von
Torsten Sprenger
Universität Kassel

sprenger@mathematik.uni-kassel.de
www.mathematik.uni-kassel.de/~sprenger

November 2005

Generalvoraussetzungen:

1. $F(n, k)$ sei ein hypergeometrischer Term, d.h.
 $\frac{F(n, k+1)}{F(n, k)} \in \mathbb{Q}(n, k)$, $\frac{F(n+1, k)}{F(n, k)} \in \mathbb{Q}(n, k)$.
2. $F(n, k)$ besitze einen endlichen Träger, d.h.
die Menge $T_F(n) = \{k \mid F(n, k) \neq 0\}$ sei endlich für alle $n \in \mathbb{N}$.
3. Sei $S(n) = \sum_{k=-\infty}^{\infty} F(n, k)$.

Ziele:

1. Bestimmung einer geschlossenen Form für $S(n)$
2. Bestimmung einer Rekursionsgleichung für $S(n)$

Beispiele:

1. $\sum_{k=-\infty}^{\infty} \binom{n}{k} = 2^n$
2. $\sum_{k=-\infty}^{\infty} \binom{n}{k}^2 = \binom{2n}{n}$
3. $\sum_{k=-\infty}^{\infty} \binom{n-k}{k}$
4. $\sum_{k=-\infty}^{\infty} (-1)^k \binom{2n}{k}^3 = (-1)^n \frac{(3n)!}{n!^3}$

Definition 1 Sei F ein hypergeometrischer Term in n und k und S eine Strukturmenge ($S \subseteq \mathbb{Z}^2$, nichtleer und endlich). Ferner sei $a_{ij} \in \mathbb{Q}[n]$ für alle $(i, j) \in S$ und $a_{ij} \neq 0$ für mindestens ein $(i, j) \in S$. Es gelte

$$\sum_{(i,j) \in S} a_{ij}(n) F(n+i, k+j) = 0. \quad (1)$$

Dann ist (1) eine **k-freie Rekursionsgleichung** (oder kurz **k-freie Rekursion**) für F .

Sei $S(n) = \sum_{k=-\infty}^{\infty} F(n, k)$. Gilt $a_i \in \mathbb{Q}[n]$ für alle $i \in \{0, \dots, m\}$ und $a_i \neq 0$ für mindestens ein i aus $\{0, \dots, m\}$, so ist

$$\sum_{i=0}^m a_i(n) S(n+i) = 0. \quad (2)$$

eine **homogene, lineare Rekursionsgleichung für $S(n)$ mit polynomialen Koeffizienten** (kurz **holonome Rekursion**).

Mit Hilfe des Fasenmyer-Algorithmus kann man k -freie Rekursionsgleichungen für F bestimmen.

Algorithmus 2 (Fasenmyer-Algorithmus)

1. **Eingabe:** ein hypergeometrischer Term $F(n, k)$
eine Strukturmengende S .
2. Mache den Ansatz

$$\sum_{(i,j) \in S} a_{ij}(n)F(n+i, k+j) = 0, \quad (3)$$

wobei a_{ij} rationale Funktionen in n sind.

3. Teile (3) durch $F(n, k)$ und vereinfache die auftretenden Quotienten $F(n+i, k+j)/F(n, k)$. Man erhält

$$\sum_{(i,j) \in S} a_{ij}(n)R_{ij}(n, k) = 0, \quad (4)$$

wobei R_{ij} rationale Funktionen in n und k sind.

4. Multipliziere mit einem gemeinsamen Nenner, um die Gleichung

$$\sum_{(i,j) \in S} a_{ij}(n)p_{ij}(n, k) = 0 \quad (5)$$

zu bekommen, bei der p_{ij} Polynome in n und k sind.

5. Betrachte die linke Seite der Polynomgleichung als Polynom in k ($P_F^S(k) = \sum_{(i,j) \in S} a_{ij}(n)p_{ij}(n, k)$ nennt man das assoziierte Polynom) und vergleiche die Koeffizienten mit 0, um ein homogenes lineares Gleichungssystem für die a_{ij} 's über dem Körper der rationalen Funktionen in n zu erhalten.
6. Wenn nur die triviale Lösung $a_{ij} \equiv 0$ für alle $(i, j) \in S$ existiert, dann gibt es keine k -freie Rekursionsgleichung für F bzgl. S .

Ausgabe: „Es existiert keine k -freie Rekursion für F bzgl. S .“

Andernfalls setze die erhaltene Lösung in (3) ein und multipliziere mit einem gemeinsamen Nenner.

Ausgabe: die durch den letzten Schritt entstandene, i. allg. nicht eindeutige k -freie Rekursion für F bzgl. der Strukturmengende S .

Beweis Da F ein hypergeometrischer Term ist, erhält man durch Induktion, dass alle $F(n+i, k+j)/F(n, k)$ rationale Funktionen in n und k sind. Also folgt die Gleichung (4), die zu (3) äquivalent ist. Multiplizieren wir (4) mit einem gemeinsamen Nenner, so erhält man die zu (4) äquivalente Polynomgleichung (5). Die linke Seite der Gleichung (5) ist genau dann 0, wenn sie das Nullpolynom in k ist, was den Koeffizientenvergleich in Schritt 5 rechtfertigt. Demzufolge besitzt das daraus resultierende homogene und lineare Gleichungssystem für die a_{ij} 's über dem Körper der rationalen Funktionen in n genau dann eine nicht-triviale Lösung, wenn es eine k -freie Rekursionsgleichung für F bzgl. S gibt. \square

Sitzung 1 (Der Fasenmyer-Algorithmus)

Satz 3 Sei

$$\sum_{(i,j) \in S} a_{ij}(n)F(n+i, k+j) = 0$$

eine k -freie Rekursionsgleichung für F . Dann ist (unter den Generalvoraussetzungen)

$$\sum_i \left(\sum_{j \in S(i)} a_{ij}(n) \right) S(n+i) = 0$$

eine holonome Rekursionsgleichung für $S(n)$ mit $S(i) = \{j \mid (i, j) \in S\}$.

Beweis

$$\begin{aligned} \sum_{(i,j) \in S} a_{ij}(n)F(n+i, k+j) &= 0 \quad \Bigg| \quad \sum_{k=-\infty}^{\infty} \\ \sum_{k=-\infty}^{\infty} \left(\sum_{(i,j) \in S} a_{ij}(n)F(n+i, k+j) \right) &= \sum_{k=-\infty}^{\infty} 0 \\ \sum_{(i,j) \in S} a_{ij}(n) \left(\sum_{k=-\infty}^{\infty} F(n+i, k+j) \right) &= 0 \\ \sum_{(i,j) \in S} a_{ij}(n)S(n+i) &= 0 \\ \sum_i \left(\sum_{j \in S(i)} a_{ij}(n) \right) S(n+i) &= 0 \end{aligned}$$

□

Sitzung 2 (Summation über k -freie Rekursionsgleichungen)

Frage: Existiert zu jedem hypergeometrischen Term eine k -freie Rekursionsgleichung?

Man kann zeigen, dass zu dem hypergeometrischen Term $F(n, k) = \frac{1}{n^2+k^2}$ keine k -freie Rekursion existieren kann. Aus diesem Grund kommen wir nun zu dem Begriff des zulässigen Terms.

Definition 4 Sei $pp, qq \in \mathbb{N}_0$. Es gelte

- $P \in \mathbb{Q}[n, k]$
- $a_p \in \mathbb{Z}, b_p \in \mathbb{Z}$ und $c_p \in \mathbb{Q}$ für alle $p \in [1..pp]$
- $u_q \in \mathbb{Z}, v_q \in \mathbb{Z}$ und $w_q \in \mathbb{Q}$ für alle $q \in [1..qq]$
- $x, y \in \mathbb{Q}$.

Dann ist

$$F(n, k) := P(n, k) \frac{\prod_{p=1}^{pp} \Gamma(a_p n + b_p k + c_p)}{\prod_{q=1}^{qq} \Gamma(u_q n + v_q k + w_q)} x^n y^k$$

ein zulässiger hypergeometrischer Term.

Sitzung 3 (Zulässige hypergeometrische Terme)

Satz 5 Sei F ein zulässiger hypergeometrischer Term in n und k . Dann existiert eine k -freie Rekursionsgleichung für F und somit auch eine holonome Rekursion für $S(n)$.

Beweis Wir skizzieren die Beweisidee und betrachten dazu rechteckige Strukturmengen $S_{IJ} = \{(i, j) \mid 0 \leq i \leq I, 0 \leq j \leq J\}$. Untersuchen wir das beim Fasenmyer-Algorithmus auftretende lineare Gleichungssystem, so stellt man folgendes fest:

Anzahl der Gleichungen: $\deg_k(P_F^{S_{IJ}}) + 1$

Anzahl der Variablen: $|S_{IJ}|$

Bei gewisser Wahl von I kann man nun J hinreichend gross wählen, so dass

$$\deg_k(P_F^{S_{IJ}}) + 1 < |S_{IJ}|$$

gilt und damit mehr Variablen als Gleichungen auftreten. Folglich besitzt das lineare Gleichungssystem eine nicht-triviale Lösung. Aus dieser Tatsache folgt (mit Satz 3) schliesslich die Behauptung. \square

Korollar 6 Wendet man den Fasenmyer-Algorithmus auf zulässige Terme an, so terminiert dieser Algorithmus.

In der Praxis ist der Fasenmyer-Algorithmus allerdings ineffizient, da die auftretenden linearen Gleichungssysteme in der Regel sehr gross sind und CAS dann schnell an ihre Grenzen stossen. Aus diesem Grund ist der Zeilberger-Algorithmus [Koe98] dem Fasenmyer-Algorithmus vorzuziehen, da dieser sehr effizient ist. Möchte man aber Rekursionsgleichungen für mehrfache Summen bestimmen, so kann man nur sehr wenige Summen iterativ mit dem Zeilberger-Algorithmus ermitteln. Mit dem Fasenmyer-Algorithmus hingegen kommt man (theoretisch) immer zum Erfolg. An dieser Stellen möchten wir nun einige Identitäten, bei denen mehrfache Summen auftreten, mit dem Fasenmyer-Algorithmus beweisen.

Beispiele:

$$1. \sum_i \sum_j \binom{n}{j} \binom{j}{i} x^i y^{j-i} z^{n-j} = (x + y + z)^n$$

$$2. \sum_i \sum_j \binom{n}{i} \binom{n+i}{i} \binom{i}{j}^3 = \sum_k \binom{n}{k}^2 \binom{n+k}{k}^2$$

$$3. \sum_i \sum_j \binom{r}{i} \binom{s}{j} \binom{t}{n-i-j} = \binom{r+s+t}{n}$$

Sitzung 4 (Mehrfache Summation)

Zwei Strategien, die den Fasenmyer-Algorithmus effizienter und somit auch im multivariaten Fall durchführbar machen, werden im Folgenden kurz beschrieben. Für eine exakte Einführung siehe [Weg97] oder [Spr04].

Definition 7 Sei F ein zulässiger Term in n und k und S eine Strukturmenge für F . Dann ist S eine **P -maximale Strukturmenge**, wenn es keine Strukturmenge S' mit $S \subsetneq S'$ gibt, für die $\deg_k P_S^F = \deg_k P_{S'}^F$ gilt.

Geht man nun von (einer möglichst kleinen) rechteckigen Strukturmenge aus, so bestimmt man die kleinste P -maximale Strukturmenge, die die rechteckige Strukturmenge umfasst. Die Anzahl der Gleichungen des linearen Gleichungssystems verändert sich dadurch nicht, aber man erhöht die Anzahl der Variablen. Somit steigt auch die Wahrscheinlichkeit eine k -freie Rekursionsgleichung zu finden.

Sitzung 5 (P -maximale Strukturmenge)

Eine weitere Effizienzsteigerung bringt eine Verallgemeinerung des Fasenmyer-Algorithmus. Dabei müssen die Rekursionsgleichungen nicht mehr notwendig k -frei sein, sondern die Koeffizienten können noch von k abhängen. Wenn man diese Rekursion in eine sogenannte (nicht-triviale) Zertifikatsrekursion umwandeln kann, so erhält man wiederum eine holonome Rekursion für die Summe $S(n)$.

Definition 8 Sei F ein hypergeometrischer Term in n und k und Δ_k bezeichne den Vorwärtsoperator ($\Delta_k(f(k)) = f(k+1) - f(k)$). Dann nennen wir

$$\sum_{(i,j) \in S_0} a_{ij}(n)F(n+i, k+j) + \Delta_k \left(\sum_{(i,j) \in S_1} b_{ij}(n, k)F(n+i, k+j) \right) = 0$$

eine **Zertifikats-Rekursion** oder **Zertifikat** von F . Der **Hauptteil** der Zertifikats-Rekursion ist gegeben durch $\sum_{(i,j) \in S_0} a_{ij}(n)F(n+i, k+j)$. Außerdem bezeichnen wir das Zertifikat als **trivial**, wenn der Hauptteil des Zertifikats 0 ist, andernfalls ist die Rekursion **nicht-trivial**.

Hat man eine nicht-triviale Zertifikatsrekursion für $F(n, k)$ gefunden, so erhält man durch Summation eine holonome Rekursion. Das liegt an zwei Dingen. Zum einen ist der Hauptteil des Zertifikats k -frei und zum andern verschwindet der Δ_k -Term durch Teleskopieren.

Sitzung 6 (Zertifikatsrekursionen)

Man kann nun den Fasenmyer-Algorithmus so modifizieren, dass dieser nicht-triviale Zertifikatsrekursionen bestimmt. Die linearen Gleichungssysteme, die bei diesem Algorithmus gelöst werden müssen sind erheblich kleiner als bei dem herkömmlichen Fasenmyer-Algorithmus, so dass der Algorithmus nicht nur wesentlich schneller ist, sondern zudem oftmals eine Rekursion für $S(n)$ mit kleinster Ordnung bestimmt, was man auch in der letzten Sitzung sehen konnte.

Literatur

- [Koe98] W. Koepf - *Hypergeometric Summation - An Algorithmic Approach to Summation and Special Function Identities* - Vieweg, Advanced Lectures in Mathematics, April 1998 5
- [Spr04] T. Sprenger - *Algorithmen für mehrfache Summen* - Diplomarbeit, Universität Kassel, Juli 2004 5
- [Weg97] K. Wegschaider - *Computer Generated Proofs of Binomial Multi-Sum Identities* - Diplomarbeit, J. Kepler Universität Linz, Mai 1997 5