

Seminar
Computational Mathematics
Algorithmische Summation
“Der Gosper-Algorithmus”

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

20. Mai 2003

Inhaltsverzeichnis

3	Gosper-Algorithmus	2
3.1	Einführung in die unbestimmte Summation	2
3.2	Kurzbeschreibung des Algorithmus	4
3.3	Beweis des Gosper-Algorithmus	6
3.4	Zusammenfassung des Gosper-Algorithmus	11
3.5	Anwendungsbeispiele für den Gosper-Algorithmus	12

3 Gosper-Algorithmus

3.1 Einführung in die unbestimmte Summation

In diesem Abschnitt geht es um den Gosper-Algorithmus. Ein kurzer Blick auf die Integralrechnung zeigt interessante Parallelen zu diesem Thema auf. Der Hauptsatz der Differential- und Integralrechnung sagt aus, daß zu jeder stetigen Funktion f eine Stammfunktion F existiert, d. h. daß

$$F'(x) = f(x) \quad \text{bzw.} \quad F(x) = \int f(x)dx$$

gilt. Man sagt dann auch, daß F ein *unbestimmtes Integral* von f ist. Mit Kenntnis von F ist es einfach, durch die bekannte Gleichung

$$\int_a^b f(x)dx = F(b) - F(a)$$

jedes beliebige *bestimmte Integral* von f auszurechnen.

Gehen wir nun von dem eigentlichen Problem aus. Wir möchten auf eine effiziente Art und Weise die Summe

$$\sum_{k=m}^n a_k$$

berechnen. Wenn man nun obige Tatsachen auf das Problem der Summation überträgt, kann man durch Kenntnis einer *Antidifferenz* s_k für gegebenes a_k , also einer Folge s_k für die

$$a_k = s_{k+1} - s_k \tag{1}$$

gilt, die Summe in ähnlicher Weise wie bei Integralen berechnen. Der Ausdruck s_k soll zudem ein hypergeometrischer Term sein, d.h. es soll

$$\frac{s_{k+1}}{s_k} \in \mathbb{Q}(k) \tag{2}$$

gelten. Das Finden von s_k nennt man *unbestimmte Summation*. Hat man ein s_k mit der Eigenschaft (1) gefunden, dann gilt

$$\sum_{k=m}^n a_k = (s_{n+1} - s_n) + (s_n - s_{n-1}) + \cdots + (s_{m+1} - s_m) = s_{n+1} - s_m.$$

Der Algorithmus von Gosper berechnet bei Eingabe von a_k die hypergeometrische Antidifferenz s_k , sofern diese existiert. Ist dem so, dann nennt man a_k *Gosper-summierbar*. Wenn s_k ein hypergeometrischer Term ist, dann ist auch a_k hypergeometrisch, denn es gilt nach (1) und (2)

$$\frac{a_{k+1}}{a_k} = \frac{s_{k+2} - s_{k+1}}{s_{k+1} - s_k} = \frac{s_{k+1}}{s_k} \frac{\frac{s_{k+2}}{s_{k+1}} - 1}{\frac{s_{k+1}}{s_k} - 1} = \frac{u_k}{v_k}, \tag{3}$$

wobei u_k und v_k Polynome aus dem Polynomring über den rationalen Zahlen sind. Diese Polynome können mit dem Algorithmus 2.1 gefunden werden.

3.2 Kurzbeschreibung des Algorithmus

Die Idee des Algorithmus von Gosper wird im folgenden beschrieben.

1. Die Eingabe besteht aus dem hypergeometrischen Term a_k . Dieser Term läßt sich auch folgendermaßen schreiben: $a_k = p_k b_k$, wobei p_k ein Polynom und b_k ein Ausdruck bestehend aus Fakultäten ist. Der Quotient b_{k+1}/b_k ist rational und wird mit Hilfe von zwei Polynomen q_k und r_k ausgedrückt: $b_{k+1}/b_k = q_{k+1}/r_{k+1}$.
2. Obig beschriebene Polynome p_k , q_k und r_k kann man immer so wählen, daß folgende Eigenschaft erfüllt ist

$$\gcd(q_k, r_{k+j}) = 1 \quad \forall j \in \mathbb{N}_0. \quad (4)$$

Es gilt

$$\frac{a_{k+1}}{a_k} = \frac{p_{k+1} q_{k+1}}{p_k r_{k+1}}. \quad (5)$$

3. Nun wird eine Funktion f_k definiert durch

$$f_k := \frac{s_{k+1} p_{k+1}}{a_{k+1} r_{k+1}} \quad \text{oder} \quad s_k = \frac{r_k}{p_k} f_{k-1} a_k. \quad (6)$$

Anhand der Gleichung

$$f_k = \frac{s_{k+1} p_{k+1}}{a_{k+1} r_{k+1}} = \frac{s_{k+1} p_{k+1}}{s_{k+2} - s_{k+1} r_{k+1}} = \frac{1}{\frac{s_{k+2}}{s_{k+1}} - 1} \frac{p_{k+1}}{r_{k+1}}$$

sieht man, daß f_k rational ist.

4. Gosper konnte aufgrund von (4) nun zeigen, daß die Funktion f_k ein Polynom ist. Nach Definition von f_k gilt

$$a_k = s_{k+1} - s_k = \frac{r_{k+1}}{p_{k+1}} f_k a_{k+1} - \frac{r_k}{p_k} f_{k-1} a_k.$$

Multipliziert man diese Gleichung mit p_k/a_k und verwendet (5), ergibt sich die lineare und inhomogene Rekurrenzgleichung

$$p_k = \frac{a_{k+1}}{a_k} \frac{p_k}{p_{k+1}} r_{k+1} f_k - r_k f_{k-1} = q_{k+1} f_k - r_k f_{k-1} \quad (7)$$

für f_k .

5. Mit Hilfe von der Rekurrenzgleichung für f_k kann man dann eine obere Schranke n für den Grad von f_k ermitteln. Man setzt nun

$$f_k = \sum_{i=0}^n c_i k^i$$

und erhält durch Einsetzen von f_k in (7) und Koeffizientenvergleich ein lineares Gleichungssystem mit den Unbekannten c_i . Durch Lösen dieses Gleichungssystems bekommt man f_k und schließlich mit (6) auch das gesuchte s_k . Wenn der Gosper-Algorithmus kein f_k findet, dann existiert auch keine hypergeometrische Antidifferenz s_k von a_k .

3.3 Beweis des Gosper-Algorithmus

Der Beweis des Algorithmus von Gosper unterteilt sich in vier Abschnitte. Zunächst gilt folgendes Lemma:

Lemma 3.1 *Die Funktionen p_k , q_k und r_k in (5) können so gewählt werden, daß*

$$\gcd(q_k, r_{k+j}) = 1 \quad \forall j \in \mathbb{N}_0. \quad (8)$$

Beweis Setze $p_k := 1$, $q_k := u_{k-1}$ und $r_k := v_{k-1}$. Wenn (8) erfüllt ist, so sind wir fertig, andernfalls existiert ein $j \in \mathbb{N}_0$, so daß

$$\gcd(q_k, r_{k+j}) = g_k \neq 1. \quad (9)$$

Sei J die endliche Menge aller $j \in \mathbb{N}_0$ für die (9) gilt. Für jedes $j \in \mathbb{N}_0$ eliminiert man den gemeinsamen Teiler g_k , indem man neue Funktionen p'_k , q'_k und r'_k wie folgt einführt

$$p'_k = p_k g_k g_{k-1} \cdots g_{k-j+1}, \quad q'_k = \frac{q_k}{g_k} \quad \text{und} \quad r'_k = \frac{r_k}{g_{k-j}}.$$

Dann gilt

$$\frac{p'_{k+1} q'_{k+1}}{p'_k r'_{k+1}} = \frac{p_{k+1} g_{k+1} g_k \cdots g_{k-j+2} q_{k+1} g_{k-j+1}}{p_k g_k g_{k-1} \cdots g_{k-j+1} g_{k+1} r_{k+1}} = \frac{p_{k+1} q_{k+1}}{p_k r_{k+1}}.$$

Wegen (9) ist

$$\gcd(q'_k, r'_{k+j}) = \gcd\left(\frac{q_k}{g_k}, \frac{r_{k+j}}{g_k}\right) = 1.$$

Man setzt nun $p_k = p'_k$, $q_k = q'_k$ und $r_k = r'_k$ und aktualisiert die Menge J , die nun mindestens ein Element (nämlich j) weniger enthält. Ist die Menge J nichtleer wiederholt man obige Schritte solange, bis schließlich J leer und somit (8) erfüllt ist. \square

Für zwei Polynome q_k und r_k ist die Zahl

$$\text{disp}(q_k, r_k) := \max\{j \in \mathbb{N}_0 \mid \gcd(q_k, r_{k+j}) \neq 1\}$$

die *Dispersion* von q_k und r_k . Das Maximum der leeren Menge soll $-\infty$ sein. Die *Dispensionsmenge* von q_k und r_k ist gegeben durch

$$J := \{j \in \mathbb{N}_0 \mid \gcd(q_k, r_{k+j}) \neq 1\}.$$

Wie findet man nun die Dispersionsmenge von zwei Polynomen q_k und r_k ?

Ein einfacher Algorithmus wäre der folgende:

Man berechnet die Resultante von q_k und r_{k+j} bezüglich k und erhält dann ein Polynom in j . Dessen nichtnegative, ganzzahlige Nullstellen sind die Elemente der Dispersionsmenge von q_k und r_k . Besitzt q_k den Grad n und r_k den Grad m , so besitzt das Polynom in j den Grad mn . Das Finden der nichtnegativen, ganzzahligen Nullstellen von diesem Polynom vom Grad mn kann allerdings bei großen n und m sehr lange dauern, so daß dieser Algorithmus nicht effizient ist. Wir wollen deshalb einen anderen Algorithmus studieren, der auf Faktorisierung von Polynomen beruht und wesentlich schneller ist, als der eben beschriebene Algorithmus.

Algorithmus 3.2 *Algorithmus zur Bestimmung der Dispersionsmenge von zwei Polynomen q_k und r_k :*

1. Eingabe: $q_k, r_k \in \mathbb{Q}[k]$.
 2. Faktorisiere q_k und r_k über \mathbb{Q} .
 3. Setze $J := \{\}$. Für jeden Faktor s_k von q_k und jeden Faktor t_k von r_k bestimme $D := \text{primedispersion}[s, t, k]$ durch folgende Schritte
 - (a) Wenn sich der Grad m von s_k von dem Grad n von t_k unterscheidet, setze $D := \{\}$ und gib D zurück.
 - (b) Berechne folgende Koeffizienten:

$$a := \text{Coefficient}[s, k, n]$$

$$b := \text{Coefficient}[s, k, n-1]$$

$$c := \text{Coefficient}[t, k, n]$$

$$d := \text{Coefficient}[t, k, n-1]$$
 - (c) Wenn $j := \frac{bc-ad}{acn}$ keine nichtnegative ganze Zahl ist, setze $D := \{\}$ und gib D zurück.
 - (d) Wenn $cs_k - at_{k+j} \equiv 0$ ist, dann setze $D := \{j\}$, sonst $D := \{\}$. Gib D zurück.
- $J := J \cup D$.
4. Ausgabe: J .

Beweis Wir zeigen zunächst, daß die Dispersion von zwei irreduziblen Polynomen q_k und r_k durch $\text{primedispersion}[q, r, k]$ gegeben ist. Angenommen q_k und r_k haben die Dispersion $j \geq 0$ und sind irreduzibel. Dann gilt

$$\gcd(q_k, r_{k+j}) = g_k \neq 1.$$

Da aber q_k und r_k irreduzibel sind, unterscheiden sich g_k von q_k , sowie g_k von r_{k+j} nur durch eine Einheit. Dementsprechend hat q_k denselben Grad n wie r_k und es gilt (a). Folglich haben die zwei Polynome

$$q_k = ak^n + bk^{n-1} + \dots$$

und

$$r_k = ck^n + dk^{n-1} + \dots$$

eine Dispersion $j \in \mathbb{N}_0$ genau dann, wenn

$$\frac{c}{a}q_k \equiv r_{k+j} = c(k+j)^n + d(k+j)^{n-1} + \dots = ck^n + (cnj + d)k^{n-1} + \dots \quad (10)$$

(binomische Lehrsatz). Dazu müssen aber die Koeffizienten von k^{n-1} auf beiden Seiten übereinstimmen. Also

$$\frac{bc}{a} = cnj + d \quad \text{oder} \quad j = \frac{bc - ad}{acn}. \quad (11)$$

Dieses j muß nichtnegativ und ganzzahlig sein. Ist dem so, dann kann man r_{k+j} ausrechnen und (10) überprüfen.

Um die gesamte Dispersionsmenge zu bekommen, wendet man die Funktion **primedispersion** auf alle irreduziblen Faktoren von q_k und r_k an und sammelt die Dispersionen in der Menge J . \square

Als nächstes ist zu zeigen, daß f_k definiert durch (6) ein Polynom ist.

Lemma 3.3 *Die rationale Funktion f_k definiert durch (6), (7) und (8) ist ein Polynom.*

Beweis Da die Funktion f_k rational ist, gilt

$$f_k = \frac{c_k}{d_k}$$

mit Polynomen c_k und d_k . Annahme: f_k ist kein Polynom. Dann hat d_k positiven Grad und es gilt

$$\gcd(c_k, d_k) = 1 = \gcd(c_{k-1}, d_{k-1}). \quad (12)$$

Multipliziert man die Rekurrenzgleichung (7) mit $d_k d_{k-1}$, erhält man

$$d_k d_{k-1} p_k = d_k d_{k-1} (q_{k+1} f_k - r_k f_{k-1}) = c_k d_{k-1} q_{k+1} - c_{k-1} d_k r_k. \quad (13)$$

Sei nun $j \geq 0$ die Dispersion von d_k mit sich selbst, dann ist j die größte ganze Zahl, für die

$$\gcd(d_k, d_{k+j}) = g_k \neq 1 \quad (14)$$

gilt. Da j maximal und d_{k+j} ein Vielfaches von g_k ist, ist

$$\gcd(d_{k-1}, d_{k+j}) = 1 = \gcd(d_{k-1}, g_k). \quad (15)$$

Wenn man den Index k in (14) um $-(j+1)$ verschiebt, bekommt man

$$\gcd(d_{k-(j+1)}, d_{k-1}) = g_{k-(j+1)} \neq 1 \quad (16)$$

und durch Verschiebung von k um $-j$ in (15) und unter Berücksichtigung der Tatsache, daß $d_{k-(j+1)}$ ein Vielfaches von $g_{k-(j+1)}$ ist, gilt

$$\gcd(d_{k-(j+1)}, d_k) = 1 = \gcd(g_{k-(j+1)}, d_k). \quad (17)$$

Dividiert man die Gleichung (13) durch g_k so erhält man

$$\frac{d_k d_{k-1} p_k}{g_k} = \frac{c_k d_{k-1} q_{k+1}}{g_k} - \frac{c_{k-1} d_k r_k}{g_k}. \quad (18)$$

Anhand dieser Gleichung kann man folgendes feststellen:

g_k ist teilbar durch d_k wegen (14) und somit ist die linke Seite von (18) ein Polynom. Der rechte Term der rechten Seite ist demnach auch ein Polynom. Betrachten wir nun den linken Term der rechten Seite. Wegen (15) sind d_{k-1} und g_k relativ prim. Gleiches gilt für c_k und g_k , denn g_k ist Teiler von d_k und es gilt (12). Also folgt, daß q_{k+1} teilbar ist durch g_k , bzw. q_k teilbar ist durch g_{k-1} . Im nächsten Schritt dividieren wir (13) durch $g_{k-(j+1)}$ und erhalten

$$\frac{d_k d_{k-1} p_k}{g_{k-(j+1)}} = \frac{c_k d_{k-1} q_{k+1}}{g_{k-(j+1)}} - \frac{c_{k-1} d_k r_k}{g_{k-(j+1)}}. \quad (19)$$

Anhand dieser Gleichung kann man folgendes feststellen:

$g_{k-(j+1)}$ ist teilbar durch d_{k-1} wegen (16) und somit ist die linke Seite von (19) ein Polynom. Der linke Term der rechten Seite ist demnach auch ein Polynom. Betrachten wir nun den rechten Term der rechten Seite. Wegen (17) sind d_k und $g_{k-(j+1)}$ relativ prim. Gleiches gilt für c_{k-1} und $g_{k-(j+1)}$, denn $g_{k-(j+1)}$ ist Teiler von d_{k-1} und es gilt (12). Also folgt, daß r_k teilbar ist durch $g_{k-(j+1)}$, bzw. r_{k+j} teilbar ist durch g_{k-1} .

Die Polynome q_k und r_{k+j} sind also beide teilbar durch g_{k-1} . Das ist aber ein Widerspruch zu der Bedingung (8) und demnach ist f_k ein Polynom. \square

Der vierte und letzte Teil des Beweises von dem Algorithmus von Gosper dient dazu eine obere Grenze n für den Grad des Polynoms f_k zu bekommen. Diese obere Grenze benötigt man, um ein allgemeines Polynom vom Grad n in die Gleichung (7) einzusetzen und durch Koeffizientenvergleich f_k zu berechnen.

Lemma 3.4 *Eine obere Grenze für den Grad von f_k ist gegeben durch den folgenden Algorithmus.*

1. Wenn $\deg(q_{k+1} + r_k) \leq \deg(q_{k+1} - r_k)$, dann

$$\deg f_k = \deg p_k - \deg(q_{k+1} - r_k)$$

2. Wenn $n := \deg(q_{k+1} + r_k) > \deg(q_{k+1} - r_k)$, dann sei a der Koeffizient von k^n von dem Polynom $q_{k+1} + r_k$, und b der Koeffizient von k^{n-1} von dem Polynom $q_{k+1} - r_k$.

(a) Wenn $-2b/a$ keine nichtnegative ganze Zahl ist, dann

$$\deg f_k = \deg p_k - n + 1$$

(b) Wenn $-2b/a$ nichtnegativ und ganzzahlig ist, dann

$$\deg f_k \leq \max\{-2b/a, \deg p_k - n + 1\}.$$

Beweis Wir schreiben die Gleichung (7) um, so daß

$$p_k = (q_{k+1} - r_k) \frac{f_k + f_{k-1}}{2} + (q_{k+1} + r_k) \frac{f_k - f_{k-1}}{2}. \quad (20)$$

Für jedes Polynom das von dem Nullpolynom verschieden ist, gilt

$$\deg(f_k - f_{k-1}) = \deg(f_k + f_{k-1}) - 1, \quad (21)$$

da sich die höchsten Koeffizienten in $f_k - f_{k-1}$ wegheben, wohingegen der höchste Koeffizient in $f_k + f_{k-1}$ nicht verschwindet. Der Grad von dem Nullpolynom sei -1 . Wenn $\deg(q_{k+1} + r_k) \leq \deg(q_{k+1} - r_k)$ ist, dann besitzt der zweite Summand in (20) kleineren Grad als der erste Summand. Das liefert (1.). Sei m der Grad von f_k und $f_k = ck^m + \dots$. Wenn $\deg(q_{k+1} + r_k) > \deg(q_{k+1} - r_k)$ ist, dann gilt wegen (20)

$$p_k = (b + a \frac{m}{2}) ck^{n+m-1} + \text{Terme niedrigeren Grades.}$$

Der Koeffizient von k^{n+m-1} ist nur dann ungleich 0, wenn $m \neq -2b/a$. Das ist der Fall, wenn $-2b/a \notin \mathbb{N}_0$, was uns (2.a) bringt. Ist $-2b/a \in \mathbb{N}_0$, so könnte auch $m = -2b/a$ gelten, was schließlich zu (2.b) führt. \square

Mit diesem Lemma endet der Beweis von dem Gosper-Algorithmus.

3.4 Zusammenfassung des Gosper-Algorithmus

Hier ist der Gosper-Algorithmus nochmal Schritt für Schritt zusammengefaßt:

1. Die Eingabe ist ein hypergeometrischer Term $a_k \neq 0$.
2. Algorithmus (2.1) berechnet die Polynome u_k und v_k , für die

$$\frac{a_{k+1}}{a_k} = \frac{u_k}{v_k}$$

gilt.

3. Man bildet die Polynome p_k , q_k und r_k gemäß Lemma 3.1.
4. Mit Hilfe von Lemma 3.4 berechnet man die obere Grenze m des Grades von f_k . Wenn $m < 0$, dann existiert keine hypergeometrische Antidifferenz s_k .
5. Setze das Polynom

$$f_k = \sum_{j=0}^m c_j k^j$$

in die Rekurrenzgleichung

$$p_k = q_{k+1}f_k - r_k f_{k-1}$$

für f_k ein. Durch Koeffizientenvergleich erhält man ein lineares Gleichungssystem für die Unbekannten c_j ($j = 0, 1, \dots, m$), daß man dann löst. Existiert keine Lösung, so existiert auch keine hypergeometrische Antidifferenz s_k .

6. Die Ausgabe ist schließlich $s_k = \frac{r_k}{p_k} f_{k-1} a_k$.

3.5 Anwendungsbeispiele für den Gosper-Algorithmus

Beispiel 3.1 (Polynome) Jedes Polynom a_k ist Gosper-summierbar, da jedes Polynom a_k eine polynomiale Antidifferenz s_k besitzt (Beweis durch Induktion). Angenommen, wir kürzen keine gemeinsamen Faktoren bei dem Term a_{k+1}/a_k , dann initialisiert man bei Anwendung des Gosper-Algorithmus die Polynome p_k , q_k und r_k mit $p_k = 1$, $q_k = a_k$ und $r_k = a_{k-1}$. Die Dispersionsmenge enthält somit die Zahl 1, so daß $p_k = a_k$ und $q_k = r_k = 1$ (nach Lemma 3.1) gewählt werden müssen, so daß (8) gilt. Folglich entsteht die einfache Rekurrenzgleichung

$$a_k = f_k - f_{k-1} \quad (22)$$

für f_k und es gilt nach (6) $f_k = s_{k+1}$. Nach (21) folgt, daß $\deg s_k = \deg f_k = \deg a_k + 1$. Schließlich muß man nun noch die Koeffizienten von s_k bestimmen, indem man das zugehörige lineare Gleichungssystem mit $\deg a_k + 1$ Unbekannten löst.

Für Polynome gibt es allerdings schnellere Algorithmen als der Gosper-Algorithmus, um zu einer Antidifferenz zu gelangen.

Beispiel 3.2 (Rationale Funktionen) Nicht jede rationale Funktion ist Gosper-summierbar. Nimmt man beispielsweise den Term $a_k = 1/k$, so beweist der Gosper-Algorithmus, daß keine hypergeometrische Antidifferenz existiert. Das sieht man so ein: $p_k = 1$, $q_k = k - 1$ und $r_k = k$ erfüllen die Bedingungen von Lemma 3.1 und nach Lemma 3.4 ist die obere Grenze des Grades von f_k gegeben durch 0. Die Rekurrenzgleichung sieht dann folgendermaßen aus (mit $f_k = c$)

$$1 = ck - ck = 0,$$

so daß offensichtlich keine Lösung existiert. Deswegen bilden die *harmonischen Zahlen*

$$H_k = \sum_{j=1}^k \frac{1}{j}$$

keinen hypergeometrischen Term.

Wenn aber ein rationales a_k eine rationale Antidifferenz s_k besitzt, dann ist a_k Gosper-summierbar und wir können s_k mit Hilfe von Gosper bestimmen. Nehmen wir zum Beispiel

$$a_k = \frac{1}{k} - \frac{1}{k+1} = \frac{1}{k(k+1)}.$$

Offensichtlich ist $s_k = -1/k$. Der Algorithmus von Gosper läuft so ab:

$p_k = 1$, $q_k = k - 1$ und $r_k = k + 1$ lautet die Initialisierung nach Lemma 3.1 und

ist gleichzeitig auch die endgültige Wahl dieser Polynome, da (8) schon erfüllt ist. Der Fall (2b) von Lemma 3.4 bringt uns auf die Gradgrenze 1 für f_k . Sei nun $f_k = a + bk$, so erhalten wir

$$1 = k(a + bk) - (k + 1)(a + b(k - 1)) = b - a.$$

Setzen wir $b = 0$ (den Grad von f_k wählen wir so gering wie möglich), bekommen wir $a = -1$. Also ist $f_k = -1$ und nach (6)

$$s_k = \frac{r_k}{p_k} f_{k-1} a_k = -(k + 1) \frac{1}{k(k + 1)} = -\frac{1}{k}.$$

Beispiel 3.3 (Binomialkoeffizient) Wir stellen uns jetzt die Frage, ob der Ausdruck $a_k := \binom{n}{k}$ Gosper-summierbar ist. Wenn dem so ist, dann kann man die Summe

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

für beliebiges m einfach ausrechnen. Ausserdem könnte man dann die binomische Identität

$$\sum_{k=0}^m \binom{n}{k} = 2^n$$

erweitern, die ein Spezialfall ($n = m$) von der entstehenden Formel darstellen würde. Gosper bringt zunächst

$$\frac{a_{k+1}}{a_k} = \frac{n - k}{k + 1}$$

und $p_k = 1$, $q_k = n - k + 1$ und $r_k = k$ sind die Polynome, die (8) erfüllen. Der Fall (1) von Lemma 3.4 gibt aber -1 als obere Grenze des Grades von f_k aus. Folglich existiert kein Polynom f_k , daß die Rekurrenzgleichung erfüllt und demnach ist a_k nicht Gosper-summierbar.

Nun betrachten wir die alternierende Summe mit $a_k := (-1)^k \binom{n}{k}$. Wir kennen bereits die Formel (Beispiel 3.4)

$$\sum_{k=0}^m a_k = \sum_{k=0}^m (-1)^k \binom{n}{k} = (-1)^m \frac{n - m}{n} \binom{n}{m},$$

die zeigt, daß a_k tatsächlich Gosper-summierbar ist. Es gilt

$$\frac{a_{k+1}}{a_k} = \frac{k - n}{k + 1}$$

und $p_k = 1$, $q_k = k - n - 1$ und $r_k = k$ sind die Polynome, die (8) erfüllen. Lemma 3.4 führt zu der oberen Gradgrenze 0 für f_k . Also sei $f_k = c$ und die Rekurrenzgleichung lautet dann

$$1 = (k - n)c - kc = -nc$$

mit der Lösung $c = -1/n$, so daß $f_k = -1/n$. Schließlich erhalten wir durch (6)

$$s_k = \frac{r_k}{p_k} f_{k-1} a_k = -\frac{k}{n} a_k = -\frac{k}{n} (-1)^k \binom{n}{k}.$$