

# Algorithmische Summation $q$ -hypergeometrischer Reihen

Masterarbeit am FB10 Universität Kassel  
Institut für Mathematik  
Schwabauer Arthur  
18.12.2014



# Inhaltsverzeichnis

	<b>1</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen der <math>q</math>-Theorie</b>	<b>3</b>
2.1 Grundlagen der $q$ -Theorie . . . . .	3
2.1.1 Zahlenbereiche . . . . .	3
2.1.2 Polynom und rationale Funktion . . . . .	3
2.1.3 $q$ -Analogon und $q$ -Klammer . . . . .	3
2.1.4 $q$ -Pochhammer-Symbol . . . . .	4
2.1.5 $q$ -Gammafunktion . . . . .	4
2.1.6 $q$ -Fakultät . . . . .	5
2.1.7 $q$ -Binomialkoeffizient . . . . .	5
2.1.8 $q$ -hypergeometrische Funktion . . . . .	6
2.1.9 Beispiel einer $q$ -Erweiterung: $q$ -binomischer Lehrsatz . . . . .	7
<b>3 <math>q</math>-Gosper Algorithmus</b>	<b>9</b>
<b>4 <math>q</math>-Zeilberger Algorithmus</b>	<b>17</b>
<b>5 Implementation in Mathematica</b>	<b>21</b>
5.1 Vorbereitung . . . . .	21
5.2 $q$ -Gosper Algorithmus . . . . .	25
5.3 $q$ -Zeilberger Algorithmus . . . . .	26



# Arthur Schwabauer

## 1 Einleitung

Oftmals wird in der Mathematik mit endlichen oder unendlichen Summen hantiert oder versucht Identitäten zu beweisen, welche Summen beinhalten. Das einfachste Beispiel, was gleichzeitig ein gutes Beispiel für den Einstieg in das Thema ist, ist die geometrische Reihe  $\Sigma = \sum_{k=0}^{\infty} a_0 q^k$  mit  $q \in \mathbb{R}$ , wobei  $\Sigma = a_0/(1-q)$  gilt, falls  $|q| < 1$ . Schreiben wir  $a_k = a_0 q^k$ , dann gilt  $q = a_k/a_{k-1}$ .

Im Allgemeinen nennt man eine Summe  $\sum_{k=0}^{\infty} a_k$  mit  $r = a_k/a_{k-1} \in \mathbb{K}(k)$ ,  $\mathbb{K}$  ein Körper, und  $a_0 = 1$  *hypergeometrische Funktion*. Existiert nun zu der Summe eine diskrete Stammfunktion  $s_k$ , d.h.  $s_k - s_{k-1} = a_k$ , dann wird dieser vom Gosper Algorithmus [Gos1978] bestimmt.

Ein weiterer Algorithmus bezüglich hypergeometrischen Funktionen ist der Zeilberger Algorithmus [Zeilb1990]. Betrachten wir dazu Summen der Form  $\Sigma(n) := \sum_{k=0}^n A(n, k)$ , dann bestimmt der Zeilberger Algorithmus zu einem festen  $p \in \mathbb{N}$  die Lösung  $\sigma_j(n) \in \mathbb{F}(n)$ ,  $j = 1, 2, \dots, p$ , der Rekursionsgleichung  $p$ -ter Ordnung,

$$\Sigma(n) + \sum_{k=0}^p \sigma_k(n) \Sigma(n-k),$$

wobei  $\mathbb{F}$  der Körper der rationalen Funktionen über  $\mathbb{Q}$  ist in einer beliebigen aber festen Anzahl an Unbekannten, mit Ausnahme von  $k$  und  $n$ .

Es tauchen jedoch zunehmend in verschiedenen Anwendungsgebieten Reihen auf, bei denen  $f(k) = a_k/a_{k-1} \in \mathbb{F}(q^k)$  gilt.  $q$  ist hier eine festen Unbekannte und  $\mathbb{F}$  ist diesmal der Körper der rationalen Funktion über  $\mathbb{Q}$  in einer beliebigen aber festen Anzahl an Unbekannten inklusive  $q$ , aber nicht  $k$ . Die entsprechenden Reihen nennt man  $q$ -hypergeometrische Reihen und den Term  $a_k$   $q$ -hypergeometrisch bezüglich  $k$ . Es stellt sich heraus, dass viele Eigenschaften und Algorithmen mit mehr oder wenigen Änderungen von hypergeometrischen auf  $q$ -hypergeometrische Reihen übertragen werden können.

In der folgenden Arbeit widme ich mich hauptsächlich dem Artikel von Tom H. Koornwinder [Koornw1993] zu Gosper's und Zeilberger's Algorithmus und deren  $q$ -Analoga

und der Implementation beider Algorithmen in Mathematica. Die Ansätze und Ideen in Koornwinders Artikel zur Umsetzung dieser beiden Algorithmen in der  $q$ -Theorie sollen auf den nächsten Seiten konkretisiert werden und zum Schluss mit Mathematica umgesetzt werden.

## 2 Grundlagen der $q$ -Theorie

### 2.1 Grundlagen der $q$ -Theorie

In diesem Abschnitt werden Definitionen und Notationen vorgestellt, die in dieser Arbeit verwendet werden. Insbesondere werden wir den Begriff des  $q$ -Analogons einführen und unter Anderem den Binomialkoeffizienten, das Pochhammersymbol und die hypergeometrische Funktion zusammen mit ihren  $q$ -Analoga definieren. Die Grundvoraussetzungen der nächsten Kapitel.

#### 2.1.1 Zahlenbereiche

Wir benötigen die folgenden Mengen:

Die Menge der natürlichen Zahlen  $\mathbb{N} = \{1, 2, 3, \dots\}$ ,

die Menge der nichtnegativen ganzen Zahlen  $\mathbb{N}_0 = \mathbb{N} \cup \{0\} = \{0, 1, 2, \dots\}$ ,

die Menge der ganzen Zahlen  $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$  und

die Menge der rationalen Zahlen  $\mathbb{Q}$ .

#### 2.1.2 Polynom und rationale Funktion

Sei  $\mathcal{R}$  ein Ring. Ein **Polynom**  $p$  in  $x$  mit **Koeffizienten** in  $\mathcal{R}$  ist definiert als

$$p(x) := \sum_{i=0}^n a_i x^i$$

mit  $a_i \in \mathcal{R}$  und  $n \in \mathbb{N}_0$ .  $\mathcal{R}[x]$  ist der **Ring aller Polynome** in  $x$  mit **Koeffizienten** in  $\mathcal{R}$ . Der Quotientenkörper von  $\mathcal{R}[x]$  ist der **Körper der rationalen Funktionen über  $\mathcal{R}$**  und wird geschrieben als  $\mathcal{R}(x)$ . Ein Element  $r \in \mathcal{R}(x)$  ist eine **rationale Funktion über  $\mathcal{R}$  in  $x$**  und lässt sich darstellen als Quotient von Polynomen  $r, s \in \mathcal{R}[x]$ , wobei der Nenner  $s \neq 0$  sein muss.

#### 2.1.3 $q$ -Analogon und $q$ -Klammer

Umgangssprachlich gesehen ist ein  $q$ -Analogon die Verallgemeinerung eines Ausdrucks oder einer Funktion mittels Einführung eines neuen Parameters  $q$ , wobei durch Grenzwertbildung  $q \rightarrow 1$  wieder der ursprüngliche Ausdruck bzw. die ursprüngliche Funktion entsteht. Der Begriff des  $q$ -Analogon ist allerdings nicht streng formal definiert und wird in der Literatur auch verwendet um Analogia für Sätze, Lemmata oder Algorithmen in der  $q$ -Theorie zu bezeichnen. Wie man unten im Fall des  $q$ -Pochhammer-Symbols sieht, muss die Grenzwertbildung  $q \rightarrow 1$  bei ein  $q$ -Analogon auch nicht immer den ursprünglichen Ausdruck liefern. Abweichungen von der Norm dienen in solchen Fällen praktischen Zwecken.

Das einfachste Beispiel ist das  $q$ -Analogon einer nichtnegativen ganzen Zahl  $n$ :

$$\lim_{q \rightarrow 1} \frac{1 - q^n}{1 - q} = n$$

Es ist ein häufig verwendeter Ausdruck in der  $q$ -Theorie, wir definieren uns in diesem Zusammenhang die  $q$ -Klammer

$$[n]_q := \frac{1 - q^n}{1 - q} = 1 + q + q^2 + \dots + q^{n-1}.$$

### 2.1.4 $q$ -Pochhammer-Symbol

Das  $q$ -Pochhammer Symbol ist definiert durch

$$(a; q)_k := \begin{cases} \prod_{j=0}^{k-1} (1 - aq^j) & , \text{ falls } k \geq 0 \\ \prod_{j=1}^{-k} (1 - aq^{-j})^{-1} & , \text{ falls } k < 0 \\ \prod_{j=0}^{\infty} (1 - aq^j) & , \text{ falls } k = \infty \end{cases}$$

und ist eine häufig verwendete Kurzschreibweise bei der Formulierung von  $q$ -Analoga. Man beachte, dass man für  $q \rightarrow 1$  nicht das klassische Pochhammer-Symbol

$$(a)_k := \begin{cases} a(a+1)(a+2)\dots(a+k-1) & , \text{ falls } k \in \mathbb{N} \\ 1 & , \text{ falls } k = 0 \\ \frac{1}{(a-1)(a-2)\dots(a+k)} & , \text{ falls } -k \in \mathbb{N} \text{ und } a \notin \{1, 2, \dots, -k\} \end{cases}$$

erhält. Es gilt der Zusammenhang

$$(a)_k = \lim_{q \rightarrow 1} \frac{(q^a; q)_k}{(1 - q)^k}.$$

Für ein endliches  $k$  kann das  $q$ -Pochhammer-Symbol auch durch

$$(a; q)_k = \frac{(a; q)_{\infty}}{(aq^k; q)_{\infty}}$$

ausgedrückt werden.

### 2.1.5 $q$ -Gammafunktion

Die  $q$ -Gammafunktion  $\Gamma_q(z)$  ist nach Thomae [Thomae1896] und Jackson [Jackson1904] definiert durch

$$\Gamma_q(z) := \frac{(q; q)_{\infty}}{(q^z; q)_{\infty}} (1 - q)^{1-z}, 0 < q < 1.$$

Falls  $z = n + 1$  eine positive ganze Zahl ist, dann reduziert sich der Ausdruck zu

$$\Gamma_q(n + 1) = 1(1 + q)(1 + q + q^2) \dots (1 + q + \dots + q^{n-1}).$$



Für  $q \rightarrow 1^-$  konvergiert  $\Gamma_q(n+1)$  gegen  $\Gamma(n+1) = n!$ . Bei Gasper und Rahman [GasRahm2004] wird gezeigt, dass tatsächlich

$$\lim_{q \rightarrow 1^-} \Gamma_q(z) = \Gamma(z)$$

gilt und dass  $\Gamma_q(z)$  die Funktionalgleichung

$$f(z+1) = \frac{1-q^z}{1-q} f(z), \quad \Gamma(1) = 1 \quad (1)$$

löst, welche das  $q$ -Analogon zur bekannten Funktionalgleichung der Gammafunktion,

$$\Gamma(z+1) = z\Gamma(z), \quad \Gamma(1) = 1,$$

ist.

### 2.1.6 $q$ -Fakultät

Die  $q$ -Fakultät ist definiert durch

$$\begin{aligned} [k]_q! &:= 1(1+q)(1+q+q^2) \dots (1+q+\dots+q^{k-1}) \\ &= \frac{(q; q)_k}{(1-q)^k}. \end{aligned}$$

So wie die Gammafunktion eine Erweiterung der Fakultät für reelle und komplexe Zahlen ist, ist die  $q$ -Gammafunktion eine Erweiterung der  $q$ -Fakultät. Denn es gilt

$$[k]_q! = \Gamma_q(k+1)$$

### 2.1.7 $q$ -Binomialkoeffizient

Analog zum Binomialkoeffizienten ergibt sich der  $q$ -Binomialkoeffizient durch

$$\binom{n}{m}_q := \begin{cases} \prod_{i=1}^m \frac{[n-i+1]_q}{[i]_q} & , \text{ falls } 0 \leq m \leq n \\ 0 & , \text{ sonst.} \end{cases}$$

Für  $0 \leq m \leq n$  gilt somit

$$\binom{n}{m}_q = \frac{(1-q^n)(1-q^{n-1}) \dots (1-q^{n-m+1})}{(1-q)(1-q^2) \dots (1-q^m)} = \frac{(q; q)_n}{(q; q)_m (q; q)_{n-m}}.$$

### 2.1.8 $q$ -hypergeometrische Funktion

Die  $q$ -hypergeometrische Funktion  ${}_r\varphi_s$  (auch basic hypergeometric function genannt) ist eine Erweiterung der hypergeometrischen Funktion

$${}_rF_s \left( \begin{matrix} a_1, \dots, a_r \\ b_1, \dots, b_s \end{matrix} \middle| x \right) := \sum_{k=0}^{\infty} \frac{(a_1)_k (a_2)_k \dots (a_r)_k}{(b_1)_k (b_2)_k \dots (b_s)_k} \frac{x^k}{k!}$$

und ist von der Form

$${}_r\varphi_s \left( \begin{matrix} a_1, \dots, a_r \\ b_1, \dots, b_s \end{matrix} \middle| q, z \right) := \sum_{k=0}^{\infty} \frac{(a_1, a_2, \dots, a_r; q)_k}{(b_1, b_2, \dots, b_s; q)_k} \frac{z^k}{(q; q)_k} \left( (-1)^k q^{\binom{k}{2}} \right)^{1+s-r}, \quad (2)$$

wobei  $(a_1, a_2, \dots, a_r; q)_k$  eine Abkürzung ist für  $\prod_{i=1}^r (a_i; q)_k$ . Es gilt hier der Grenzwert

$$\lim_{q \rightarrow 1^-} {}_r\varphi_s \left( \begin{matrix} q^{a_1}, \dots, q^{a_r} \\ q^{b_1}, \dots, q^{b_s} \end{matrix} \middle| q, (q-1)^{1+s-r} z \right) = {}_rF_s \left( \begin{matrix} a_1, \dots, a_r \\ b_1, \dots, b_s \end{matrix} \middle| z \right)$$

Der zusätzliche Faktor  $\left( (-1)^k q^{\binom{k}{2}} \right)^{1+s-r}$  war in früheren Definitionen nicht enthalten, es liefert uns die praktische Identität

$$\lim_{a_r \rightarrow \infty} {}_r\varphi_s \left( \begin{matrix} a_1, \dots, a_r \\ b_1, \dots, b_s \end{matrix} \middle| q, \frac{z}{a_r} \right) = {}_{r-1}\varphi_s \left( \begin{matrix} a_1, \dots, a_{r-1} \\ b_1, \dots, b_s \end{matrix} \middle| q, z \right),$$

welche sich aus dem Grenzwert für  $k \geq 0$  herleiten lässt:

$$\begin{aligned} \lim_{a_r \rightarrow \infty} \frac{(a_r; q)_k}{a_r^k} &= \lim_{a_r \rightarrow \infty} \prod_{j=0}^{k-1} (a_r^{-1} - q^j) \\ &= \prod_{j=0}^{k-1} (-q)^j \\ &= (-1)^k q^{k(k-1)/2} \\ &= (-1)^k q^{\binom{k}{2}} \end{aligned}$$

Schreiben wir  $f_k$  für den Summanden von (2), so folgt

$$\frac{f_{k+1}}{f_k} = \frac{(1 - a_1 q^k)(1 - a_2 q^k) \dots (1 - a_r q^k)}{(1 - b_1 q^k)(1 - b_2 q^k) \dots (1 - b_s q^k)} \frac{(-q^k)^{1+s-r} z}{1 - q^{k+1}}. \quad (3)$$

Falls dabei der Quotient  $f_{k+1}/f_k$  rational in  $q^k$  ist, heißt  $f_k$   $q$ -hypergeometrisch bzgl.  $q^k$ . Allerdings kann jede in  $q^k$  rationale Funktion in die Form (3) gebracht werden. Also ist jede Reihe  $\sum_{k=0}^{\infty} g_k$ , für die  $g_{k+1}/g_k$  rational in  $q^k$  ist und  $g_0 = 1$ , eine  $q$ -hypergeometrische Funktion.

Aufgrund der einfachen Form des Quotienten  $f_{k+1}/f_k$  lässt sich mit dem Quotientenkriterium der Konvergenzradius von  ${}_r\varphi_s$  bestimmen.

Sei  $0 < |q| < 1$ . Für  $r \leq s$  konvergiert die Reihe für alle  $z \in \mathbb{C}$ . Für  $r > s + 1$  und  $z \neq 0$  divergiert die Reihe, sofern sie nicht terminiert. Im Fall  $r = s + 1$  konvergiert die Reihe nur für alle  $z \in \mathbb{C}$  mit  $|z| < 1$ .

Sei  $|q| > 1$ . Falls  $|z| < |b_1 b_2 \dots b_s q| / |a_1 a_2 \dots a_r|$ , dann konvergiert die Reihe absolut. Falls  $|z| > |b_1 b_2 \dots b_s q| / |a_1 a_2 \dots a_r|$  und falls  ${}_r\varphi_s$  nicht terminiert, dann divergiert die Reihe.

### 2.1.9 Beispiel einer $q$ -Erweiterung: $q$ -binomischer Lehrsatz

Eine der elementarsten Beispiele für eine unbestimmte Summation ist die geometrische Reihe

$$\sum_{n=0}^{\infty} z^n = (1 - z)^{-1}$$

wobei  $z$  reell oder komplex ist und  $|z| < 1$ . Wendet man die Taylorentwicklung auf die Funktion  $f(z) = (1 - z)^{-a}$  im Bereich  $|z| < 1$  an und beachte dabei, dass für die Ableitungen

$$f^{(n)}(z) = (a)_n (1 - z)^{-(a+n)}|_{z=0} = (a)_n$$

gilt, so erhält man den binomischen Lehrsatz

$$\sum_{n=0}^{\infty} \frac{(a)_n}{n!} z^n = (1 - z)^{-a}, |z| < 1. \quad (4)$$

Setzt man  $a = -m$ ,  $m \in \mathbb{N}$ , und  $z = -x/y$ , so erhält man die bekanntere Darstellung

$$(x + y)^m = \sum_{n=0}^{\infty} \binom{m}{n} x^n y^{m-n}. \quad (5)$$

Sei  $0 < q < 1$ . Nach der Regel von L'Hospital gilt

$$\lim_{q \rightarrow 1^-} \frac{1 - q^a}{1 - q} = a$$

und damit

$$\lim_{q \rightarrow 1^-} \frac{(1 - q^a)(1 - q^{a+1} \dots (1 - q^{a+n-1}))}{(1 - q)(1 - q^2) \dots (1 - q^n)} = \lim_{q \rightarrow 1^-} \frac{(q^a; q)_n}{(q; q)_n} = \frac{(a)_n}{n!}. \quad (6)$$

Wir können nun den Koeffizienten von  $z^n$  in (4) mit dem Bruch auf der linken Seite von (6) ersetzen. Wir schreiben in verallgemeinerter Form

$$f(b, z) := \sum_{n=0}^{\infty} \frac{(b; q)_n}{(q; q)_n} z^n, \quad (7)$$

wobei  $a, q, z$  reell oder komplex sind mit  $|z| < 1$ . Wir gehen für den Moment von  $|q| < 1$  aus. Mit dieser Voraussetzung ist die unbestimmte Summe konvergent. Um nun eine

Formel für  $f(b, z)$  zu bestimmen, betrachten wir zunächst die folgende Umformung. Mit  $1 - b = (1 - bq^n) - b(1 - q^n)$  gilt

$$\begin{aligned}
f(b, z) &= 1 + \sum_{n=1}^{\infty} \frac{(b; q)_n}{(q; q)_n} z^n \\
&= 1 + \sum_{n=1}^{\infty} \frac{(bq; q)_{n-1}}{(q; q)_n} [(1 - bq^n) - b(1 - q^n)] z^n \\
&= 1 + \sum_{n=1}^{\infty} \frac{(bq; q)_n}{(q; q)_n} z^n - b \sum_{n=1}^{\infty} \frac{(bq; q)_{n-1}}{(q; q)_{n-1}} z^n \\
&= f(bq, z) - bz f(bq, z) = (1 - bz) f(bq, z). \tag{8}
\end{aligned}$$

Nach  $n$  Iterationen erhält man

$$f(b, z) = (bz; q)_n f(bq^n, z), \quad n = 1, 2, \dots,$$

und für  $n \rightarrow \infty$  folgt wegen  $q^n \rightarrow 0$

$$f(b, z) = (bz; q)_{\infty} f(0, z). \tag{9}$$

Das unendliche Produkt  $(b; q)_{\infty} = \prod_{k=0}^{\infty} (1 - bq^k)$ ,  $|q| < 1$  divergiert, wenn  $b \neq 0$  und  $|q| \geq 1$ . Daher geht man oftmals von  $|q| < 1$  aus, wenn  $(b; q)_{\infty}$  auftaucht. Nun setzen wir  $b = q$  in (9) und erhalten

$$f(0, z) = \frac{f(q, z)}{(qz; q)_{\infty}} = \frac{(1 - z)^{-1}}{(qz; q)_{\infty}} = \frac{1}{(z; q)_{\infty}},$$

das heißt, es gilt

$$f(b, z) = \frac{(bz; q)_{\infty}}{(z; q)_{\infty}}, \quad |z| < 1, |q| < 1.$$

Setzt man nun  $b = q^a$ , erhält man

$$\lim_{q \rightarrow 1^{-1}} \frac{(q^a z; q)_{\infty}}{(z; q)_{\infty}} = \lim_{q \rightarrow 1^{-1}} \sum_{n=0}^{\infty} \frac{(b; q)_n}{(q; q)_n} z^n = \sum_{n=0}^{\infty} \frac{(a)_n}{n!} z^n = (1 - z)^{-a}.$$

Dieses  $q$ -Analogon vom binomischen Lehrsatz, auch genannt  $q$ -binomischer Lehrsatz, wurde von Cauchy [Cauchy1893] und Heine [Heine1847] hergeleitet.

### 3 $q$ -Gosper Algorithmus

Sei  $\mathbb{F}$  der Körper der rationalen Funktionen über  $\mathbb{Q}$  in einer festen Zahl an Unbekannten inklusive der Unbekannten  $q$ , aber ohne  $k$ . Sei  $\mathbb{F}(q^k)$  der Körper der rationalen Funktionen in  $q^k$  über  $\mathbb{F}$  und  $\mathbb{F}[q^k]$  der Polynomring in  $q^k$  über  $\mathbb{Q}$ . Sei  $a(k)$ ,  $k = 0, 1, \dots$  eine Folge in  $\mathbb{F}$  mit  $a(k) \neq 0 \forall k \in \mathbb{N}_0$ , sodass  $a(k)/a(k-1) \in \mathbb{F}(q^k)$ . Wir bezeichnen die Folge  $s(k)$ ,  $k = -1, 0, 1, \dots$ , in  $\mathbb{F}$  als Antidifferenz für  $a(k)$ , wenn

$$s(n) - s(m) = \sum_{k=m+1}^n a(k), \quad n, m = -1, 0, 1, \dots, m < n$$

oder, äquivalent dazu,

$$s(k) - s(k-1) = a(k), \quad k \in \mathbb{N}_0 \quad (10)$$

gilt. Eine Antidifferenz  $s(k)$  ist bis auf einen konstanten Faktor eindeutig gegeben. Der  $q$ -Gosper Algorithmus liefert uns nun Folgendes.

1. Es bestimmt, ob eine Lösung  $s(k)$  zu (10) mit  $s(k) \neq 0 \forall k \in \mathbb{N}_0$  und  $s(k)/s(k-1) \in \mathbb{F}(q^k)$  existiert.
2. Falls eine Lösung existiert, wird die Lösung ausgegeben.

In solch einem Fall ist auch  $s(k)$  eine rationale Funktion in  $\mathbb{F}$ . Es folgen einige Lemmata, die wir benötigen werden.

**Lemma 3.1.** *Sei  $b(k)$  eine rationale Funktion in  $\mathbb{F}(q^k)$ . Dann existieren  $p(k), r_1(k)$  und  $r_2(k)$  in  $\mathbb{F}[q^k]$ , sodass*

$$b(k) = \frac{p(k)}{p(k-1)} \frac{r_1(k)}{r_2(k)}, \quad (11)$$

$$\gcd(r_1(k), p(k-1)) = 1 = \gcd(r_2(k), p(k)), \quad (12)$$

und

$$\gcd(r_1(k), r_2(k+j)) = 1 \quad (13)$$

für alle  $j \in \mathbb{N}_0$  gilt.  $\gcd$  sei hier der größte gemeinsame Teiler.

**Beweis:** Für den Existenzbeweis nehmen wir zunächst an, es gelten die Gleichungen (11), (12) sowie für ein  $i \in \mathbb{N}$  die Gleichung (13) für alle  $j \in \{0, 1, \dots, i-1\}$ . Für  $i = 1$  und  $p(k) = 1$  ist die Existenz garantiert. Wir verändern nun  $p(k), r_1(k)$  und  $r_2(k)$  über eine endliche Zahl an Iterationen, woraufhin die Gleichung (13) für alle  $j \in \mathbb{N}_0$  gelten wird. Falls  $r_1(k)$  einen Primfaktor  $\gamma(k)$  hat, sodass  $\gamma(k-i)$  ein Faktor von  $r_2(k)$  ist, dann setzen wir  $\tilde{r}_1(k) := r_1(k)/\gamma(k)$ ,  $\tilde{r}_2(k) := r_2(k)/\gamma(k-i)$  und  $\tilde{p}(k) := p(k)\gamma(k)\gamma(k-1)\dots\gamma(k-i+1)$ . Wenn man nun  $p(k), r_1(k)$  und  $r_2(k)$  durch  $\tilde{p}(k), \tilde{r}_1(k)$  und  $\tilde{r}_2(k)$  ersetzt, dann gelten weiterhin (11) und (13) für alle  $j \in \{0, 1, \dots, i-1\}$ . Der Nachweis von (12) folgt aus der folgenden Überlegung. Ein gemeinsamer Teiler von  $\tilde{p}(k)$  und  $\tilde{r}_2(k)$  kann nur  $\gamma(k-j)$  für ein  $j \in \{0, 1, \dots, i-1\}$  sein. Da aber  $\gamma(k)$  ein Teiler von  $r_1(k)$  ist, kann  $\gamma(k-j)$  kein Teiler von  $r_2(k)$  sein. Ein gemeinsamer Teiler von  $\tilde{p}(k-1)$  und  $\tilde{r}_1(k)$  kann nur  $\gamma(k-j)$  für ein  $j \in \{1, \dots, i\}$  sein. Da aber  $\gamma(k-i)$  ein Teiler von  $r_2(k)$  ist, kann  $\gamma(k-j)$  kein Teiler von  $r_1(k)$  sein.  $\square$

**Bemerkung 3.2.** Die Polynome  $p(k)$ ,  $r_1(k)$  und  $r_2(k)$ , die man in Lemma 3.1 erhält, sind eindeutig bis auf einen konstanten Faktor in  $\mathbb{F}$ .

**Lemma 3.3.** Sei  $b(k)$  eine rationale Funktion in  $\mathbb{F}(q^k)$  mit  $b(k) \neq 0 \forall k \in \mathbb{N}$ . Schreibt man  $b(k)$  in der Form (11) mit den geltenden Bedingungen (13) und

$$\gcd(r_1(k), p(k-1)) \neq 0 \neq \gcd(r_2(k), p(k)) \forall k \in \mathbb{N}. \quad (14)$$

Dann gilt  $p(k) \neq 0 \forall k \in \mathbb{N}_0$  und  $r_1(k), r_2(k) \neq 0 \forall k \in \mathbb{N}$ .

**Beweis:**  $r_1(k), r_2(k) \neq 0 \forall k \in \mathbb{N}$  folgt direkt aus dem Iterationsverfahren im Beweis von Lemma 3.1. Angenommen es gelte  $p(k) = 0$  für ein  $k \in \mathbb{N}_0$ . Sei  $j$  die größte nichtnegative ganze Zahl, sodass  $p(j) = 0$ . So ein  $j$  existiert, da  $p$  nur endlich viele Nullstellen besitzt. Dann hat  $b(k)$  eine Polstelle bei  $k = j + 1$ , was ein Widerspruch zur Annahme ist.  $\square$

Wir nehmen nun an, dass  $a(k) \neq 0$  für alle  $k \in \mathbb{N}_0$  und dass

$$\frac{a(k)}{a(k-1)} = \frac{p(k)}{p(k-1)} \frac{r_1(k)}{r_2(k)}, \quad k \in \mathbb{N} \quad (15)$$

mit der Bedingung (13) für alle  $j \in \mathbb{N}_0$  gilt. Wenn (12) oder das schwächere (14) gilt, dann folgt auch  $p(k) \neq 0$  für  $k \in \mathbb{N}_0$  und  $r_1(k), r_2(k) \neq 0$  für  $k \in \mathbb{N}$ . Aus (15) folgt

$$\frac{a(k+1)r_2(k+1)}{p(k+1)} = \frac{a(k)r_1(k+1)}{p(k)}, \quad k \in \mathbb{N}_0. \quad (16)$$

Seien nun  $s(k), f(k) \in \mathbb{F}$  definiert für  $k = -1, 0, 1, \dots$ , sodass

$$s(k) = \frac{a(k+1)r_2(k+1)}{p(k+1)} f(k), \quad k = -1, 0, 1, \dots \quad (17)$$

Wegen (16) gilt auch

$$s(k) = \frac{a(k)r_1(k+1)}{p(k)} f(k), \quad k = 0, 1, \dots \quad (18)$$

Wegen Lemma 3.3 gilt nun  $s(k) \neq 0$  genau dann, wenn  $f(k) \neq 0$ , da die anderen Faktoren ungleich 0 sind. Von nun an sei  $s(k)$  und  $f(k)$  ungleich 0 für alle  $k \in \mathbb{N}_0$ .

**Lemma 3.4.** Unter den obigen Annahmen ist

$$s(k) - s(k-1) = a(k) \quad (19)$$

für alle  $k \in \mathbb{N}_0$  äquivalent zu

$$r_1(k+1)f(k) - r_2(k)f(k-1) = p(k). \quad (20)$$

**Beweis:** (17) eingesetzt in (19) ergibt

$$\frac{a(k+1)r_2(k+1)}{p(k+1)}f(k) - \frac{a(k)r_2(k)}{p(k)}f(k-1) = a(k)$$

und da  $p(k) \neq 0$ , kann (20) umgeformt werden zu

$$\frac{a(k)r_1(k+1)}{p(k)}f(k) - \frac{a(k)r_2(k)}{p(k)}f(k-1) = a(k).$$

Mit (16) folgt die Behauptung.  $\square$

**Lemma 3.5.** *Unter den Annahmen wie in Lemma 3.4 seien die Gleichungen (19) und (20) für alle  $k \in \mathbb{N}_0$  gegeben. Dann sind folgende Aussagen äquivalent:*

1.  $s(k-1)/s(k) \in \mathbb{F}(q^k)$
2.  $f(k) \in \mathbb{F}(q^k)$
3.  $f(k) \in \mathbb{F}[q^k, q^{-k}]$ <sup>1</sup>

**Beweis:** Zunächst zeigen wir  $1. \Leftrightarrow 2.$  . Aus (18) und (20) folgt

$$1 - \frac{s(k-1)}{s(k)} = \frac{a(k)}{s(k)} = \frac{p(k)}{r_1(k+1)} \frac{1}{f(k)}.$$

$3. \Rightarrow 2.$  ist schnell einzusehen. Man bringt die Summanden in  $f(k)$  auf einen gemeinsamen Nenner. Es fehlt noch  $2. \Rightarrow 3.$  zu zeigen. Angenommen  $f(k)$  hat die Form

$$f(k) = c(k)/d(k),$$

wobei  $c(k), d(k) \in \mathbb{F}[q^k]$ ,  $d(k) \neq 0$ ,

$$\gcd(c(k), d(k)) = 1 = \gcd(c(k-1), d(k-1)) \quad (21)$$

und  $d(k)$  ist kein Monom, mit anderen Worten, der niedrigste und der höchste auftauchende Exponent stimmen nicht überein. Wir formen Gleichung (20) um zu

$$r_1(k+1)c(k)d(k-1) - r_2(k)d(k)c(k-1) = p(k)d(k)d(k-1). \quad (22)$$

Sei  $j$  die größte ganze Zahl, sodass

$$\gcd(d(k), d(k+j)) = g(k) \neq 1. \quad (23)$$

So ein  $j$  existiert und ist nichtnegativ, denn  $d(k) \neq 1$ . Weil  $j$  maximal ist und  $d(k+j)$  ein Vielfaches von  $g(k)$  ist, gilt

$$\gcd(d(k-1), d(k+j)) = 1 = \gcd(d(k-1), g(k)). \quad (24)$$

---

<sup>1</sup> $f(k)$  ist ein sogenanntes Laurent-Polynom in  $q^k$  über  $\mathbb{F}$ , d.h.  $f(k)$  ist eine Summe  $\sum_{j \in \mathbb{Z}} a_j q^{jk}$  mit  $a_j \in \mathbb{F}$ , wobei  $a_j \neq 0$  für endlich viele  $j \in \mathbb{Z}$

In Gleichung (23) verschieben wir  $k$  um  $-j - 1$  und erhalten

$$\gcd(d(k - j - 1), d(k - 1)) = g(k - j - 1) \neq 1. \quad (25)$$

Auf der linken Seite von Gleichung (24) verschieben wir  $k$  um  $-j$  und erhalten

$$\gcd(d(k - j - 1), d(k)) = 1 = \gcd(g(k - j - 1), d(k)), \quad (26)$$

denn  $d(k - j - 1)$  ist ein Vielfaches von  $g(k - j - 1)$ .

Untersuchen wir nun die Teilbarkeit beider Seiten in Gleichung (22) durch  $g(k)$  und  $g(k - j - 1)$ . Wir wissen bereits, dass  $g(k)$  ein Teiler ist von  $d(k)$ . Wegen Gleichung (21) und (24) ist  $g(k)$  teilerfremd zu  $c(k)$  und  $d(k - 1)$ .  $g(k)$  in Gleichung (22) ist ein Teiler von  $r_1(k + 1)$ , d.h.  $g(k - 1)$  ist ein Teiler von  $r_1(k)$ . Analog dazu, wegen Gleichung (25) ist  $g(k - j - 1)$  ein Teiler von  $d(k - 1)$  und wegen Gleichung (21) und (26) ist  $g(k - j - 1)$  teilerfremd zu  $c(k - 1)$  und  $d(k)$ . Also ist  $g(k - j - 1)$  ein Teiler von  $r_2(k)$ , d.h.  $g(k - 1)$  ist ein Teiler von  $r_2(k + j)$ .

$r_1(k)$  und  $r_2(k + j)$  haben somit den gemeinsamen Teiler  $g(k - 1)$ , was ein Widerspruch ist zu (13). Somit ist  $f(k)$  ein Laurent-Polynom in  $\mathbb{F}[q^k, q^{-k}]$ .

□

Für das nächste Lemma sei  $\deg(g(k))$  definiert als der höchste auftauchende Exponent  $m_2$  und  $\text{ldeg}(g(k))$  definiert als der niedrigste auftauchende Exponent  $m_1$  in einem Laurent-Polynom  $g(k) = \sum_{j=m_1}^{m_2} a_j q^{jk}$  mit  $m_1, m_2 \in \mathbb{Z}$ ,  $m_1 \leq m_2$ ,  $a_{m_1} \neq 0 \neq a_{m_2}$ .

**Lemma 3.6.** *Es seien dieselben Voraussetzungen gegeben wie in Lemma 3.5 und es sei  $0 \neq f(k) \in \mathbb{F}[q^k, q^{-k}]$  eine Lösung von (20). Dann gelten die folgenden Aussagen.*

(a) Falls  $\text{ldeg}(r_1(k)) \neq \text{ldeg}(r_2(k))$ , dann gilt

$$\text{ldeg}(f(k)) = \text{ldeg}(p(k)) - \min\{\text{ldeg}(r_1(k)), \text{ldeg}(r_2(k))\}.$$

(b) Falls  $l := \text{ldeg}(r_1(k)) = \text{ldeg}(r_2(k))$ , dann sei  $d_l$  bzw.  $e_l$  der Koeffizient von  $q^{lk}$  in  $r_1(k)$  bzw.  $r_2(k)$  und es gelten die 2 folgenden Aussagen.

(b1) Falls  $\log_q(e_l/d_l) \notin \mathbb{Z}$ , dann gilt  $\text{ldeg}(f(k)) = \text{ldeg}(p(k)) - l$ .

(b2) Falls  $\log_q(e_l/d_l) \in \mathbb{Z}$ , dann gilt  $\text{ldeg}(f(k)) \geq \min\{\log_q(e_l/d_l), \text{ldeg}(p(k))\} - l$ .

(a') Falls  $\deg(r_1(k)) \neq \deg(r_2(k))$ , dann gilt

$$\deg(f(k)) = \deg(p(k)) - \max\{\deg(r_1(k)), \deg(r_2(k))\}.$$

(b') Falls  $l := \deg(r_1(k)) = \deg(r_2(k))$ , dann sei  $d_l$  bzw.  $e_l$  der Koeffizient von  $q^{lk}$  in  $r_1(k)$  bzw.  $r_2(k)$  und es gelten die 2 folgenden Aussagen.

(b1') Falls  $\log_q(e_l/d_l) \notin \mathbb{Z}$ , dann gilt  $\deg(f(k)) = \deg(p(k)) - l$ .

(b2') Falls  $\log_q(e_l/d_l) \in \mathbb{Z}$ , dann gilt  $\deg(f(k)) \leq \max\{\log_q(e_l/d_l), \deg(p(k))\} - l$ .



**Beweis:**

(a) Falls  $\text{ldeg}(r_1(k)) \neq \text{ldeg}(r_2(k))$ , dann folgt aus (3.5)

$$\begin{aligned} \min\{\text{ldeg}(r_1(k)f(k)), \text{ldeg}(r_2(k)f(k))\} &= \\ \min\{\text{ldeg}(r_1(k)), \text{ldeg}(r_2(k))\} + \text{ldeg}(f(k)) &= \text{ldeg}(p(k)) \end{aligned}$$

und daraus die Behauptung.

(b1) Sei  $\log_q(\frac{e_l}{d_l}) \notin \mathbb{Z}$  und sei  $m := \text{ldeg}(f(k))$ , dann gilt  $m + l \neq \log_q(\frac{e_l}{d_l})$  und

$$m + l \neq \log_q\left(\frac{e_l}{d_l}\right) \Leftrightarrow q^{m+l} \neq \frac{e_l}{d_l} \Leftrightarrow q^l d_l - \frac{e_l}{q^m} \neq 0.$$

Damit verschwindet in (3.5) der Koeffizient von  $p(k)$  zum Grad  $\text{ldeg}(f(k)) + l$  nicht und es folgt die Behauptung.

(b2) Sei  $\log_q(\frac{e_l}{d_l}) \in \mathbb{Z}$  und sei  $c_m$  der Koeffizient von  $f(k)$  zum Grad  $m := \text{ldeg}(f(k))$ . Im Falle, dass der Koeffizient von  $p(k)$  zum Grad  $m + l$  verschwindet, gilt

$$q^l d_l - \frac{e_l}{q^m} = 0 \Rightarrow m + l = \log_q\left(\frac{e_l}{d_l}\right)$$

und es folgt die Behauptung.

Im Falle, dass der Koeffizient von  $p(k)$  zum Grad  $m + l$  nicht verschwindet, gilt

$$\text{ldeg}(f(k)) = \text{ldeg}(p(k)) - l$$

und es folgt die Behauptung.

(a'), (b1') und (b') lassen sich mit denselben Überlegungen beweisen. □

**Bemerkung 3.7.** Wenn wir die Voraussetzung an  $p(k)$  in Lemma 3.4 abschwächen und lediglich voraussetzen, dass  $p(k) \in \mathbb{F}[q^k, q^{-k}]$  anstatt  $p(k) \in \mathbb{F}[q^k]$ , dann lässt sich ein  $\tilde{f}(k) \in \mathbb{F}[q^k]$  finden, welches (20) löst. Sei  $m$  eine in Lemma 3.6 bestimmte untere Schranke für  $\text{ldeg}(f(k))$ . Wir setzen

$$\begin{aligned} \tilde{f}(k) &:= q^{-mk} f(k), & \tilde{p}(k) &:= q^{-mk} p(k), \\ \tilde{r}_1(k) &:= r_1(k), & \tilde{r}_2(k) &:= q^{-m} r_2(k). \end{aligned}$$

$\tilde{f}(k)$  ist in  $\mathbb{F}[q^k]$  und  $\tilde{f}(k)$ ,  $\tilde{p}(k)$ ,  $\tilde{r}_1(k)$  und  $\tilde{r}_2(k)$  erfüllen ebenso die Gleichungen (15) und (20) wie auch  $f(k)$ ,  $p(k)$ ,  $r_1(k)$  und  $r_2(k)$ , aber  $\tilde{p}(k)$  könnte in  $\mathbb{F}[q^k, q^{-k}] \setminus \mathbb{F}[q^k]$  liegen. In diesem Fall hat (20) keine Lösung.

**Algorithmus 3.8. ( $q$ -Gosper Algorithmus)**

**Eingabe:** Ein  $q$ -hypergeometrischer Term  $a(k)$  mit  $k \in \mathbb{N}_0$ .

**Ausgabe:** Eine Antidifferenz  $s(k)$  zu  $a(k)$ <sup>2</sup> mit  $s(k) \neq 0 \forall k \in \mathbb{N}_0$  und  $s(k)/s(k-1) \in \mathbb{F}(q^k)$ .

<sup>2</sup>d.h.  $s(k) - s(k-1) = a(k)$ ,  $k \in \mathbb{N}_0$

1. Bestimme  $p(k)$ ,  $r_1(k)$ ,  $r_2(k)$  in (15) gemäß dem Beweis zu Lemma 3.1.
2. Bestimme  $\text{ldeg}(f(k))$  bzw. eine untere Schranke für  $\text{ldeg}(f(k))$  gemäß Lemma 3.6, ersetze  $f(k)$ ,  $p(k)$ ,  $r_1(k)$  und  $r_2(k)$  wie in Bemerkung 3.7 und bestimme dann  $d = \text{deg}(f(k))$  bzw. eine obere Schranke  $d$  für  $\text{deg}(f(k))$ . Falls  $\text{ldeg}(\tilde{p}(k)) < 0$  oder falls  $d < 0$ , dann existiert keine Antidifferenz  $s(k)$ ,  $k \in \{-1, 0, 1, \dots\}$ , für  $a(k)$  und der Algorithmus bricht ab.
3. Setze

$$f(k) := \sum_{i=0}^d f_i q^{ik},$$

wobei  $f_i \in \mathbb{F}$  noch unbekannt sind. Bestimme den Lösungsraum des Linearen Gleichungssystems, welches man erhält, indem man die Koeffizienten von

$$r_1(k+1)f(k) - r_2(k)f(k-1) - p(k) \in \mathbb{F}[q^k]$$

mit 0 gleichsetzt. Falls keine Lösung existiert, dann existiert keine Antidifferenz  $s(k)$ ,  $k \in \{-1, 0, 1, \dots\}$ , für  $a(k)$  und der Algorithmus bricht ab. Andernfalls hat der Lösungsraum Dimension 0 oder 1.

4. Falls der Lösungsraum Dimension 0 hat, prüfe ob  $f(k) \neq 0$  für alle  $k \in \mathbb{N}_0$ . Wenn das nicht der Fall ist, dann existiert keine Antidifferenz  $s(k)$ ,  $k \in \{-1, 0, 1, \dots\}$ , für  $a(k)$  und der Algorithmus bricht ab.
5. Bestimme die Antidifferenz(en)  $s(k)$ ,  $k \in \{-1, 0, 1, \dots\}$ , für  $a(k)$  mit (17) (oder (18)).

Probieren wir den Algorithmus an einem Beispiel aus.

**Beispiel:**

Gegeben ist die Reihe

$$\sum_{k=0}^{\infty} a(k) = \sum_{k=0}^{\infty} (-1)^k q^{\binom{k(k-1)}{2}} \binom{n}{k}_q, \quad n \in \mathbb{N}.$$

Gesucht ist eine Antidifferenz  $s(k)$ ,  $k \in \{-1, 0, 1, \dots\}$ , für  $a(k)$ .

Aus

$$\frac{\binom{n}{k}_q}{\binom{n}{k-1}_q} = \frac{(q; q)_{k-1} (q; q)_{n-k+1}}{(q; q)_k (q; q)_{n-k}} = \frac{1 - q^{n-k+1}}{1 - q^k}$$

folgt

$$\frac{a(k)}{a(k-1)} = -q^{k-1} \frac{\binom{n}{k}_q}{\binom{n}{k-1}_q} = -q^{k-1} \frac{1 - q^{n-k+1}}{1 - q^k} = \frac{q^k - q^{n+1}}{q(q^k - 1)},$$

also ist  $a(k)$  in  $\mathbb{F}(q^k)$  und damit ein  $q$ -hypergeometrischer Term. Die Voraussetzungen für den  $q$ -Gosper Algorithmus sind gegeben. Weiterhin lässt sich erkennen, dass

$\gcd(a(k), a(k+j)) = 1$  für alle  $j \in \mathbb{N}_0$  gilt, damit sind  $p(k) = 1$ ,  $r_1(k) = a(k)$  und  $r_2(k) = a(k-1)$  in Schritt 1. gefunden.

Nun wenden wir Lemma 3.6 zur Bestimmung von  $\text{ldeg}(f(k))$  an. Wegen  $\text{ldeg}(r_1(k)) = 0 = \text{ldeg}(r_2(k))$  tritt Fall (b) ein und wir erhalten mit  $d_0 = -q^{(n+1)}$  und  $e_0 = -q$  den Fall (b2), denn  $\log_q \frac{1}{q^n} = -n \in \mathbb{Z}$ . Es gilt

$$\text{ldeg}(f) \geq \min\{-n, 0\} = -n$$

und damit ist  $-n$  eine untere Gradschranke für  $f(k)$ . Wir führen nun die Ersetzungen gemäß Bemerkung 3.7 mit  $m = -n$  durch und erhalten

$$\tilde{p}(k) = q^{nk}, \tilde{r}_1(k) = q^k - q^{n+1}, \tilde{r}_2(k) = q^{n+1}(q^k - 1) \text{ und } \tilde{f}(k) = q^{nk} f(k).$$

Im Folgenden schreiben wir weiterhin  $p(k), r_1(k), r_2(k)$  und  $f(k)$  für  $\tilde{p}(k), \tilde{r}_1(k), \tilde{r}_2(k)$  und  $\tilde{f}(k)$ .

Schließlich bestimmen wir noch eine obere Gradschranke für das neue  $f(k)$ . Wegen  $\deg(r_1(k)) = 1 = \deg(r_2(k))$  tritt Fall (b') ein und mit  $d_1 = 1$ ,  $e_1 = q^{n+1}$  tritt Fall (b2') ein. Es gilt

$$\deg(f(k)) \leq \max\{\log(q^{n+1}), n\} - 1 = n =: d$$

für Schritt 3. setzen wir

$$f(k) := \sum_{i=0}^n f_i q^{ik}$$

und bestimmen  $f_0, \dots, f_n$  mittels Koeffizientenvergleich in

$$r_1(k+1)f(k) - r_2(k)f(k-1) - p(k) = 0.$$

Dazu:

$$\begin{aligned} & r_1(k+1)f(k) - r_2(k)f(k-1) - p(k) \\ = & (q^{k+1} - q^{n+1})f(k) - (q^{n+1}(q^k - 1))f(k-1) - q^{nk} \\ = & \left( \sum_{i=0}^n q f_i q^{(i+1)k} \right) - \left( \sum_{i=0}^n q^{n+1} f_i q^{ik} \right) - \left( \sum_{i=0}^n q^{n+1} f_i q^{(i+1)k-i} \right) + \left( \sum_{i=0}^n q^{n+1} f_i q^{ik-i} \right) \\ & - q^{nk} \\ = & \left( \sum_{i=0}^n \left( q - \frac{q^{n+1}}{q^i} \right) f_i q^{(i+1)k} \right) + \left( \sum_{i=0}^n \left( \frac{q^{n+1}}{q^i} - q^{n+1} \right) f_i q^{ik} \right) - q^{nk} \\ = & (q^{n+1} - q^{n+1})f_0 + \left( \sum_{i=1}^n \left( f_{i-1} \left( q - \frac{q^{n+1}}{q^{i-1}} \right) + f_i \left( \frac{q^{n+1}}{q^i} - q^{n+1} \right) \right) q^{ik} \right) \\ & + \underbrace{\left( q - \frac{q^{n+1}}{q^n} \right)}_{=0} f_n q^{(n+1)k} - q^{nk} \end{aligned}$$

In der letzten Zeile lässt sich die Lösung herauslesen. Nehmen wir testweise an, es gelte  $f_0 = 0$ , dann folgt durch die Rekursionsgleichungen, welche man aus dem Koeffizientenvergleich bekommt, iterativ  $f_1 = f_2 = \dots = f_{n-1} = 0$ . Die letzte Gleichung des Linearen Gleichungssystems lautet

$$f_n(q - q^{n+1}) - 1 = 0$$

und liefert uns eine mögliche Lösung

$$f(k) = \frac{q^{nk}}{q - q^{n+1}}.$$

Der Lösungsraum hat Dimension 1, denn das Gleichungssystem besitzt einen Freiheitsgrad.

Die Bedingung  $f(k) \neq 0$  für alle  $k \in \mathbb{N}_0$  in Schritt 4. trifft zu. Aus (18) ergibt sich letztendlich die Antidifferenz

$$s(k) = \frac{r_1(k+1)a(k)}{p(k)}f(k) = \frac{(q^{k+1} - q^{n+1})a(k)q^{nk}}{q^{nk}(q - q^{n+1})} = a(k)\frac{q^k - q^n}{1 - q^n}.$$

## 4 $q$ -Zeilberger Algorithmus

Sei  $\mathbb{F}$  der Körper der rationalen Funktionen über  $\mathbb{Q}$  in einer festen Zahl an Unbekannten inklusive der Unbekannten  $q$ , aber ohne  $k$  und  $n$ . Sei  $A(n, k)$  ein Term mit den Eigenschaften

- (i)  $A(n, k) \in \mathbb{F}$  für  $n, k \in \mathbb{N}_0$ ,
- (ii)  $0 \neq A(n, k) \in \mathbb{F}(q^n)$  für  $k \in \mathbb{N}_0$ ,
- (iii)  $A(n, k) = 0$ , falls  $0 \leq n < k$ ,
- (iv)  $\frac{A(n, k)}{A(n, k-1)} \in \mathbb{F}(q^n, q^k)$
- (v)  $\frac{A(n, k)}{A(n-1, k)} \in \mathbb{F}(q^n, q^k)$

Wir setzen

$$\Sigma(n) := \sum_{k=0}^n A(n, k), n \in \mathbb{N}_0. \quad (27)$$

Für ein festes  $p \in \mathbb{N}$  bestimmt der  $q$ -Zeilberger-Algorithmus, falls existent, eine Rekursionsgleichung  $p$ -ter Ordnung

$$\Sigma(n) + \sum_{j=1}^p \sigma_j(n) \Sigma(n-j) = 0, n \in \mathbb{N}, n \geq p \quad (28)$$

mit den Unbekannten  $\sigma_j(n) \in \mathbb{F}(q^n)$ ,  $j = 1, \dots, p$ . Nach Bestimmung einer solchen Rekursionsgleichung lässt sich  $\Sigma(n)$  für alle  $n \in \mathbb{N}$  iterativ bestimmen, wenn die ersten  $p$  Anfangswerte  $\Sigma(0), \dots, \Sigma(p-1)$  bekannt sind.

Der  $q$ -Zeilberger Algorithmus verwendet den  $q$ -Gosper Algorithmus zur Bestimmung der Lösung. Wir setzen dazu

$$a(k) := A(n, k) + \sum_{j=1}^p \sigma_j(n) A(n-j, k), \quad (29)$$

wobei die  $\sigma_j(n) \in \mathbb{F}(q^n)$ ,  $j = 1, \dots, p$ , noch unbekannt sind.  $a(k)$  ist damit für alle  $k \in \mathbb{N}_0$  in  $\mathbb{F}(q^n)$ . Angenommen die  $\sigma_j(n)$ ,  $j = 1, \dots, p$ , sind so gewählt, dass  $a(k) \neq 0$  und  $a(k) \in \mathbb{F}(q^n)$ . Aus (29) folgt

$$\frac{a(k)}{a(k-1)} = \frac{1 + \sum_{j=1}^p \sigma_j(n) \frac{A(n-j, k)}{A(n, k)}}{1 + \sum_{j=1}^p \sigma_j(n) \frac{A(n-j, k-1)}{A(n, k-1)}} \frac{A(n, k)}{A(n, k-1)}. \quad (30)$$

Aus Voraussetzung (v) folgt

$$\frac{A(n-j, k)}{A(n, k)} \in \mathbb{F}(q^n, q^k) \text{ und } \frac{A(n-j, k-1)}{A(n, k-1)} \in \mathbb{F}(q^n, q^k)$$

und damit ist  $a(k)/a(k-1)$  in  $\mathbb{F}(q^n)(q^k)$ . Anwenden des  $q$ -Gosper Algorithmus liefert uns die  $\sigma_j(n)$ ,  $j = 1, \dots, p$  und die Lösung  $s(k) = S(n, k)$  für (10), wobei  $s(k) \in \mathbb{F}(q^n)$

für alle  $k = -1, 0, 1, \dots$  und  $s(k) \neq 0$  für alle  $k \in \mathbb{N}_0$ . Wie die  $\sigma_j(n)$ ,  $j = 1, \dots, p$  genau bestimmt werden, folgt weiter unten. Angenommen es gilt  $s(-1) = 0$ . Dann folgt

$$S(n, m) = s(m) = \sum_{k=0}^m a(k) = \sum_{k=0}^m A(n, k) + \sum_{j=1}^p \sigma_j(n) \sum_{k=0}^m A(n-j, k), \quad (31)$$

für alle  $m \in \mathbb{N}_0$ . Angenommen es gilt auch  $S(n, n) = 0$ , dann folgt mit  $m = n$  aus (31) durch Anwenden von (27) die Rekursionsgleichung (28). Nun wenden wir den  $q$ -Gosper Algorithmus in etwas abgeänderter Form an. Wir setzen

$$\frac{A(n, k)}{A(n-1, k)} = \frac{B(n, k)}{C(n, k)} \quad (32)$$

so, dass  $B(n, k)$  und  $C(n, k)$  teilerfremd in  $\mathbb{F}[q^n, q^k]$  sind, und

$$\frac{A(n, k)}{A(n, k-1)} = \frac{D(n, k)}{E(n, k)} \quad (33)$$

so, dass  $D(n, k)$  und  $E(n, k)$  teilerfremd in  $\mathbb{F}[q^n, q^k]$  sind. Dann folgt  $B(n, k)$ ,  $C(n, k)$ ,  $D(n, k)$ ,  $E(n, k) \neq 0$  in  $\mathbb{F}(q^n)$  für alle  $k \in \mathbb{N}_0$ . (30) kann umgeschrieben werden zu

$$\frac{a(k)}{a(k-1)} = \frac{1 + \sum_{j=1}^p \sigma_j(n) \prod_{i=0}^{j-1} \frac{C(n-i, k)}{B(n-i, k)}}{1 + \sum_{j=1}^p \sigma_j(n) \prod_{i=0}^{j-1} \frac{C(n-i, k-1)}{B(n-i, k-1)}} \frac{D(n, k)}{E(n, k)}.$$

Wir setzen

$$\frac{a(k)}{a(k-1)} = \frac{p_0(k)}{p_0(k-1)} \frac{r_{10}(k)}{r_{20}(k)} \quad (34)$$

mit

$$p_0(k) := \prod_{i=0}^{p-1} B(n-i, k) + \sum_{j=1}^p \sigma_j(n) \prod_{i=0}^{j-1} C(n-i, k) \prod_{i=j}^{p-1} B(n-i, k), \quad (35)$$

$$r_{10}(k) := D(n, k) \prod_{i=0}^{p-1} B(n-i, k-1), \quad (36)$$

$$r_{20}(k) := E(n, k) \prod_{i=0}^{p-1} B(n-i, k). \quad (37)$$

Es gilt  $r_{10}(k), r_{20} \in \mathbb{F}(q^n)[q^k]$  und für alle  $k \in \mathbb{N}$   $r_{10}(k), r_{20}(k) \neq 0$  in  $\mathbb{F}(q^n)$ . Nun konstruieren wir uns gemäß dem Beweis von Lemma 3.1

$$\frac{r_{10}(k)}{r_{20}(k)} = \frac{p_1(k)}{p_1(k-1)} \frac{r_1(k)}{r_2(k)}, \quad (38)$$

wobei  $p_1(k), r_1(k), r_2(k) \in \mathbb{F}(q^n)[q^k]$ ,

$$\gcd(r_1(k), r_2(k+j)) = 1$$

für alle  $j \in \mathbb{N}_0$  und

$$\gcd(r_1(k), p_1(k-1)) = 1 = \gcd(r_2(k), p_1(k)).$$

Schließlich setzen wir noch

$$p(k) := p_0(k)p_1(k) \tag{39}$$

und erhalten

$$\frac{a(k)}{a(k-1)} = \frac{p(k)}{p(k-1)} \frac{r_1(k)}{r_2(k)}. \tag{40}$$

Zur Anmerkung, aus (35) und (29) folgt die Identität

$$p_0(k) = \frac{a(k)}{A(n, k)} \prod_{i=0}^{p-1} B(n-i, k).$$

Zusammen mit (17) und (39) folgt

$$S(n, k) = s(k) = \frac{r_2(k+1)A(n, k+1)}{p_1(k+1) \prod_{i=0}^{p-1} B(n-i, k+1)} f(k), \quad k = -1, 0, 1, \dots$$

#### Algorithmus 4.1. ( $q$ -Zeilberger Algorithmus)

**Eingabe:** Ein Term  $A(n, k)$  und ein  $p \in \mathbb{N}$ .

**Ausgabe:** Eine Rekursionsgleichung  $p$ -ter Ordnung der Form (28) zur Summation (27) mit  $\sigma_j(n) \in \mathbb{F}(q^n)$ ,  $j \in \{1, \dots, p\}$ , falls existent. Falls der Algorithmus vorher abbricht, wird ausgegeben: "Es existiert keine Rekursionsgleichung  $p$ -ter Ordnung."

1. Überprüfe die Voraussetzungen (i)-(v) am Anfang des Kapitels. Wenn die Voraussetzungen nicht gelten, breche den Algorithmus ab, ansonsten schreibe  $A(n, k)/A(n-1, k)$  und  $A(n, k)/A(n, k-1)$  wie in (32) und (33).
2. Bestimme  $p_1(k)$ ,  $r_1(k)$  und  $r_2(k)$  in (38) gemäß dem Beweis von Lemma 3.1. Falls  $r_2(0) = 0$ , breche den Algorithmus ab. Bestimme  $p(k)$  nach (39) und (35) (mit den noch Unbestimmten  $\sigma_j(n)$ ).
3. Bestimme, wie auch in Schritt 2. des  $q$ -Gosper Algorithmus, eine untere Schranke für  $\text{ldeg}(f(k))$  zur Lösung  $f(k) \in \mathbb{F}(q^n)[q^k]$  der Rekursionsgleichung

$$r_1(k+1)f(k) - r_2(k)f(k-1) = p(k)$$

gemäß Lemma 3.6. Führe die Ersetzungen wie in Bemerkung 3.7 durch (mit  $\tilde{p}_1(k) := q^{-mk}p_1(k)$  anstelle von  $\tilde{p}(k) := q^{-mk}p(k)$ ,  $p_0(k)$  bleibt unverändert) und bestimme dann eine obere Schranke  $d$  für  $\text{deg}(f(k))$ . Falls  $\text{ldeg}(\tilde{p}_1) < 0$  oder falls  $d < 0$ , dann breche den Algorithmus ab. (Beachte in diesem Schritt bei der Bestimmung der Schranken von  $\text{ldeg}(f(k))$  und  $\text{deg}(f(k))$ , dass aufgrund der Unbestimmten  $\sigma_j(n)$  der Grad von  $\tilde{p}_1$  auch kleiner sein kann.)

4. Ersetze  $f(k) := \sum_{i=0}^d f_i q^{ik}$  in der Rekursionsgleichung in Schritt 2. und setze die Koeffizienten zu den verschiedenen Exponenten von  $q^k$  gleich 0. Löse das lineare Gleichungssystem über  $\mathbb{F}(q^n)$  mit den Unbekannten  $f_i, i \in \{0, \dots, d\}$ , und  $\sigma_j(n), j \in \{1, \dots, p\}$ . Falls keine Lösung existiert, breche den Algorithmus ab, falls die Dimension des Lösungsraums größer als 0 ist, erweitere  $\mathbb{F}$  um die zusätzlich erhaltenen Parameter im Lösungsraum.
5. Nach dem Einsetzen der Lösungen in Schritt 3. müssen einige Bedingungen geprüft werden um zu versichern, dass die Schritte des  $q$ -Gosper Algorithmus noch erlaubt sind. Prüfe, ob  $p(k) \neq 0$  in  $\mathbb{F}(q^n)$  für alle  $k \in \mathbb{N}_0$  und ob  $a(0) \neq 0$  in  $\mathbb{F}(q^n)$ . Falls nicht, breche den Algorithmus ab. Nach (40) ist dann  $a(k) \neq 0$  in  $\mathbb{F}(q^n)$  für alle  $k \in \mathbb{N}_0$ . Prüfe, ob  $f(k) \neq 0$  für alle  $k \in \mathbb{N}_0$ . Falls nicht, breche den Algorithmus ab.
6. Bestimme die Antidifferenz  $s(k) = S(n, k)$  mit (17). Wegen  $r_2(0) = 0$  gilt dann  $s(-1) = 0$ . Prüfe, ob eine der Lösungen  $f_i(n)$  und  $\sigma_j(n)$  einen Pol bei  $n = p, p+1, \dots$  hat. Sei  $n_0 \in \mathbb{N}$  so gewählt, dass es keine Pole für  $n \geq n_0 \geq p$  gibt. Es gelte nun (31) für  $n \geq n_0$ . Betrachte (17). Wegen (29) und Voraussetzung (iii) gilt dann  $a(k+1)|_{k=n} = 0$  für  $n \geq n_0$ . Nun muss noch geprüft werden, ob  $r_2(k+1)|_{k=n}$  einen Pol hat oder ob  $p(k+1)|_{k=n} = 0$  für  $n \geq n_0$ . Falls dies der Fall ist, dann existiert ein  $n_1 \geq n_0$ , sodass  $S(n, n) = 0$  für  $n \geq n_1$ . Definiere  $\Sigma(n)$  wie am Anfang des Kapitels Dann gilt die Rekursionsgleichung (28) für  $n > n_1$ .



## 5 Implementation in Mathematica

### 5.1 Vorbereitung

Bevor wir den eigentlichen  $q$ -Gosper Algorithmus und im Anschluss den  $q$ -Zeilberger Algorithmus in Mathematica implementieren, benötigen wir einige Hilfsfunktionen, die in diesem Abschnitt zusammengestellt und betrachtet werden. Die Benennung der Funktionen orientiert sich am Mathematica Paket `SpecialFunctions.m` von Wolfram Koepf, Axel Rennoch und Gregor Stölting erhältlich auf der Seite

<http://www.mathematik.uni-kassel.de/~koepf/Publikationen/>

Das Paket enthält den klassischen Gosper und Zeilberger Algorithmus und hat für mich als struktureller Ansatz für die weiteren Implementationen gedient. Zur Unterscheidung kennzeichne ich in meinen Implementationen alle Funktionen, welche  $q$  als Eingabeparameter besitzen, mit dem Präfix `q`.

Zum Erkennen von Polynomen in  $\mathbb{F}[q^k]$  reicht die in Mathematica eingebaute Funktion `PolynomialQ` nicht aus. Der Aufruf `PolynomialQ[f, qk]` liefert nur dann ein korrektes Ergebnis, wenn unser Körper  $\mathbb{F}$  nicht die Unbekannte  $q$  beinhaltet. Demzufolge implementieren wir eine neue Funktion, welche zunächst alle Ausdrücke  $q^k$  in einem Polynom  $f \in \mathbb{F}[q^k]$  durch eine neue Variable ersetzt. Anschließend kann die Funktion `PolynomialQ` angewendet werden.

```

qPolynomialQ[f_, q_, k_]:=Module[{ff, xx, w, s},
ff = ExpandAll[f];
ff = ff/.q^(k) -> xx;
ff = ff/.q^(w_*k) -> xx^w;
ff = ff/.q^(k+s_) -> (xx*q^s);
ff = ff/.q^(w_*k+s_) -> (xx^w*q^s);
If[!FreeQ[ff, k],
Return[False];
];
PolynomialQ[ff, xx]
]

```

Als Nächstes benötigen wir eine Funktion zum Vereinfachen von Ausdrücken mit  $q$ -Pochhammer-Symbolen,  $q$ -Binomialkoeffizienten,  $q$ -Fakultäten und  $q$ -Gamma Funktionen:

```

qSimplifyCombinatorial[term1_, q_, k_]:=
qSimplifyGamma[qToGamma[term1, q, k], q, k];

```

Zunächst werden in `qToGamma` alle  $q$ -Pochhammer-Symbole,  $q$ -Binomialkoeffizienten

und  $q$ -Fakultäten mithilfe der Identitäten

$$\begin{aligned} \binom{n}{m}_q &= \frac{(q; q)_n}{(q; q)_m (q; q)_{n-m}}, 0 \leq m \leq n, n \geq 0, \\ [n]_q! &= \Gamma_q[n+1], n \geq 0, \\ (n; q)_m &= (q^{\log_q(n)}; q)_m = \frac{\Gamma_q(\log_q(n)) \prod_{j=0}^{m-1} (1 - q^{\log_q(n)+j})}{\Gamma_q(\log_q(n))} \\ &= (1 - q)^m \frac{\Gamma_q(\log_q(n) + m)}{\Gamma_q(\log_q(n))}, n, m \geq 0 \end{aligned}$$

zu  $q$ -Gamma Funktionen umgewandelt:

```

qToGamma[term2_, q_]:=Module[term1, n, m],
term1 = term2;
term1 = (term1/.QBinomial[n_, m_, q] →
QPochhammer[q, q, n]/(QPochhammer[q, q, m] * QPochhammer[q, q, n - m]));
term1 = (term1/.QPochhammer[n_, q, 0] → 1);
term1 = (term1/.QPochhammer[0, q, m_] → 1);
term1 = (term1/.QPochhammer[n_, q, m_] →
(1 - q)^m QGamma[Log[q, n] + m, q]/QGamma[Log[q, n], q]);
term1 = (term1/.QFactorial[n_, q]->QGamma[n + 1, q]);
Return[term1]
]

```

Der resultierende Ausdruck kann nun weiter vereinfacht werden, indem alle  $q$ -Gamma Ausdrücke, dessen Funktionsargumente einen ganzzahligen Abstand zueinander haben, zu einem  $q$ -Gamma Ausdruck zusammengefasst werden. Hierbei wird die uns bereits bekannte Rekursionsgleichung

$$\Gamma_q(z+1) = \frac{1 - q^z}{1 - q} \Gamma_q$$

bzw.

$$\Gamma_q(z+j) = \frac{(1 - q^z)(1 - q^{z+1}) \dots (1 - q^{z+j-1})}{(1 - q)^j} \Gamma_q(z), j \in \mathbb{N} \quad (41)$$

angewendet:

```

qSimplifyGamma[term1_, q_, k_]:=
Module[{term2, highl, lhighl, i, n, m},
term2 = term1/.QGamma[m_, q] :=> QGamma[ExpandAll[m], q];
highl = qGammaArgs[term2];
highl = RemoveIntegerDifferenceElements[highl];
highl = Map[ExpandAll, highl];
lhighl = Length[highl];
For[i = 1, i <= lhighl, i++,

```

```

n = highl[[i]];
term2 = (term2/.QGamma[m_, q]->Hold[m, n, q]);
term2 = (term2/.Hold -> qShiftGamma);
term2 = Evaluate[term2];
];
Return[Factor[term2]]
]

```

Die Funktion **qGammaArgs** listet alle Funktionsargumente der  $q$ -Gamma Ausdrücke auf.

```

qGammaArgs[term1.] := Module[{term2, argl, pos},
If[FreeQ[term1, QGamma], Return[{}]];
pos = Position[term1, QGamma];
argl := Map[term1][[ArgumentPosition[#]]]&, pos];
Return[argl]
]

```

```

ArgumentPosition[list1.] := Sequence[
Delete[list1, Length[list1]]/.List->Sequence, 1]

```

Die Funktion **RemoveIntegerDifferenceElements** entfernt iterativ alle diejenigen Elemente in der Liste **highl**, welche ein weiteres Element in der Liste mit einem ganzzahligen Abstand dazu besitzen. D.h. es bleiben nur noch Elemente übrig, die paarweise untereinander keinen ganzzahligen Abstand zueinander besitzen.

```

RemoveIntegerDifferenceElements[{}]:={}
RemoveIntegerDifferenceElements[list.] :=
Module[{length, last, nolast},
length = Length[list];
last = list[[length]];
nolast = Delete[list, length];
If[Superficious[nolast, last],
RemoveIntegerDifferenceElements[nolast],
Append[RemoveIntegerDifferenceElements[nolast], last]]
]

```

```

Superficious[list_, element.] :=
Or[Evaluate[Map[IntegerQ[# - element]&, list]]/.List->Sequence]]

```

Die Rekursionsgleichung (41) wird dann in der Funktion **qShiftGamma** zur Anwendung gebracht.

```

qShiftGamma[xx_, nn_, q_] := Module[{j, x, n},
x = Expand[xx];
n = Expand[nn];
If[!IntegerQ[x - n] || (x - n == 0), Return[QGamma[x, q]];
If[x - n > 0,
QGamma[n, q] * Product[(1 - q^(n + j))/(1 - q), {j, 0, x - n - 1}],
QGamma[n, q]/Product[(1 - q^(n + j))/(1 - q), {j, x - n, -1}]]

```

```
]
]
```

Bevor wir zur eigentlichen Implementation übergehen, führen wir eine weitere Funktion **qDispersionsMenge** ein, die in Schritt 1. des  $q$ -Gosper Algorithmus benötigt wird. Die Funktion bestimmt zu den Termen  $r_1(k), r_2(k) \in \mathbb{F}[q^k]$  die Menge aller  $j \in \mathbb{N}_0$ , sodass  $\gcd(r_1(k), r_2(k+j)) \neq 0$ .

```
qDispersionsMenge[r1_, r2_, q_, k_]:=
Module[{f, g, m1, m2, i, j, result, tmp, op1, op2},
f = Factor[r1];
g = Factor[r2];
If[Not[Head[f]===Times], m1 = 1, m1 = Length[f]];
If[Not[Head[g]===Times], m2 = 1, m2 = Length[g]];
result = {};
Do[
If[Head[f]===Times, op1 = f[[i]], op1 = f];
Do[
If[Head[g]===Times, op2 = g[[j]], op2 = g];
tmp = qPrimDispersion[op1, op2, q, k];
If[Not[tmp==={}], AppendTo[result, tmp[[1]]],
{j, 1, m2}],
{i, 1, m1}];
Union[result]
]
```

Es werden paarweise die Primfaktoren von  $r_1$  und  $r_2$  mit der Funktion **qPrimDispersion** verglichen. Sei  $r_1'$  ein Primfaktor von  $r_1$  und  $r_2'$  ein Primfaktor von  $r_2$ , dann gibt **qPrimDispersion[r1', r2', q, k]** eine Liste  $\{j\}$  zurück, falls  $r_1'(k+j)$  und  $r_2'(k)$  bis auf eine Konstante in  $\mathbb{F}$  für ein  $j \in \mathbb{N}_0$  übereinstimmen, ansonsten wird  $\{\}$  zurückgegeben.

```
qPrimDispersion[r1_, r2_, q_, k_]:=
Module[{s, t, tshifted, n, z1, z2, a1, a2, equal, j, rules, w, z},
s = Collect[ExpandAll[r1], q^k];
t = Collect[ExpandAll[r2], q^k];
n = Exponent[s, q^k];
If[n===0||Not[n==Exponent[t, q^k]], Return[{}]];
z1 = Coefficient[s, q^k, 0];
z2 = Coefficient[t, q^k, 0];
If[z1 == 0||z2 == 0, Return[{}]];
s = ExpandAll[s/z1];
t = ExpandAll[t/z2];
a1 = Coefficient[s, q^k, n];
a2 = Coefficient[t, q^k, n];
j = Simplify[Log[q, (a1)/(a2)]/n, Element[q, Reals]&&Positive[q]];
If[Not[IntegerQ[j]&&j >= 0], Return[{}]];
tshifted = ExpandAll[t/.k -> k + j];
equal = True;
```

```

Do[
equal = Coefficient[s, q^k, i] == Coefficient[tshifted, q^k, i];
If[equal == False, Return[{}]; ];
, {i, 1, n - 1}];
Return[{j}]

```

## 5.2 $q$ -Gosper Algorithmus

Mithilfe der vorangegangenen Funktionen kann der  $q$ -Gosper Algorithmus nun implementiert werden.

```

qDiskreteStammfunktion[a_, q_, k_, assumptions_ : True]:=
Module[{rat, u, v, p, r1, r2, y, dmenge, i, f, l, c1, c2,
ldeg, deg, log, alpha, A, B, C, rec, sol, w},
rat = qSimplifyCombinatorial[ExpandAll[a/(a/(k ->k - 1))], q];
u = Numerator[rat];
v = Denominator[rat];
If[Not[qPolynomialQ[u, q, k]&&qPolynomialQ[v, q, k]],
Return[„Eingabe ist kein q-hypergeometrischer Term“];

```

(\* Schritt 1. \*)

```

r1 = u; r2 = v; p = 1;
While[Not[(dmenge = qDispersionsMenge[r1, r2, q, k])==={}],
i = dmenge[[1]];
y = PolynomialGCD[r1, r2/.q^k -> q^(k + i)];
y = FactorList[y][[2, 1]];
r1 = r1/y;
r2 = r2/(y/.q^k -> q^(k - i));
p = p * Product[y/.q^k -> q^(k - j), {j, 0, i - 1}];
];

```

(\* Schritt 2. \*)

```

If[(l = Exponent[r1, q^k, Min]) != Exponent[r2, q^k, Min],
ldeg = Exponent[p, q^k, Min] -
Min[{Exponent[r1, q^k, Min], Exponent[r2, q^k, Min]}],
c1 = Coefficient[r1, q^k, l];
c2 = Coefficient[r2, q^k, l];
log = Simplify[Log[q, c2/c1],
Element[q, Reals]&&Positive[q]&&assumptions];
If[FreeQ[log, Log]&&Simplify[log ∈ Integers, assumptions],
ldeg = Simplify[Min[{log, Exponent[p, q^k, Min]}] - l,
assumptions];,
ldeg = Exponent[p, q^k, Min] - l;
];
];

```

```

If[Simplify[ldeg < 0, assumptions],
p = p * q^(-ldegk);
r2 = r2 * q^(-ldeg);
];
If[Simplify[Exponent[p, q^k, Min], assumptions] < 0,
Return[„Es gibt keine q-hypergeometrische Stammfunktion“]];
If[(l = Exponent[r1, q^k, Max]) ≠ Exponent[r2, q^k, Max],
deg = Exponent[p, q^k, Max] -
Max[{Exponent[r1, q^k, Max], Exponent[r2, q^k, Max]}],
c1 = Coefficient[r1, q^(k), l];
c2 = Coefficient[r2, q^(k), l];
log = Simplify[Log[q, c2/c1],
Element[q, Reals]&&Positive[q]&&assumptions];
If[FreeQ[log, Log]&&Simplify[log ∈ Integers, assumptions],
deg = Simplify[Max[{log, Exponent[p, q^k, Max]}] - l, assumptions];,
deg = Exponent[p, q^k, Max] - l;
];
];
If[Simplify[deg < 0, assumptions],
Return[„Es gibt keine q-hypergeometrische Stammfunktion“]];

(* Schritt 3. *)
A = (r1/.(k → k + 1));
B = -r2;
C = p;
f = Sum[α[j] * q^(jk), {j, 0, deg}];
rec = Collect[A * f + B * (f/.k → k - 1) - C, q^k];
sol = Assuming[assumptions,
Solve[CoefficientList[rec, q^k] == 0, Table[α[j], {j, 0, deg}]]];
If[sol==={},
Return[„Es gibt keine hypergeometrische Stammfunktion“]];
f = f/.sol[[1]];

(*Schritt4. + 5.*)
If[FreeQ[f, α[-]]&&Not[qDispersionsMenge[(q^k - 1), f, q, k]==={}],
Return[„Es gibt keine hypergeometrische Stammfunktion“]];
f = f/.α[-] → 0;
(r1/.(k → k + 1))af/p
]

```

### 5.3 q-Zeilberger Algorithmus

```

qSumRekursion[F-, q-, k-, S-[n-], assumptions- : True]:=
Module[{σ, ratk, ratn, rat, j, rec, sol, B, C, D, E, r10, r20,

```

```

r1, r2, p, l, ldeg, deg, log, f, dmenge, y,  $\alpha$ , g, RE},
RE = {};
ratn = qSimplifyCombinatorial[F/(F/.n  $\rightarrow$  n - 1), q];
ratk = qSimplifyCombinatorial[F/(F/.k  $\rightarrow$  k - 1), q];
B = Numerator[ratn];
C = Denominator[ratn];
D = Numerator[ratk];
E = Denominator[ratk];
If[Not[qPolynomialQ[B, q, k]&&qPolynomialQ[C, q, k]&&
qPolynomialQ[D, q, k]&&qPolynomialQ[E, q, k]], Return[]];

```

```
Do[
```

```
(*Schritt2.*)
```

```

r10 = D * Product[B/.{n  $\rightarrow$  n - j, k  $\rightarrow$  k - 1}, {j, 0, J - 1}];
r20 = E * Product[B/.{n  $\rightarrow$  n - j}, {j, 0, J - 1}];
rat = qSimplifyCombinatorial[ExpandAll[r10/r20], q];
r1 = Numerator[rat];
r2 = Denominator[rat];
p = 1;
While[Not[(dmenge = qDispersionsMenge[r1, r2, q, k])==={}],
j = dmenge[[1]];
y = PolynomialGCD[r1, r2/.q^k  $\rightarrow$  q^(k + j)];
y = FactorList[y][[2, 1]];
r1 = r1/y;
r2 = r2/(y/.q^k  $\rightarrow$  q^(k - j));
p = p * Product[y/.q^k  $\rightarrow$  q^(k - s), {s, 0, j - 1}];
];
If[Not[(r2/.k  $\rightarrow$  0) == 0], Continue[]];
p = p * Product[B/.n  $\rightarrow$  n - i, {i, 0, J - 1}] +
Sum[ $\sigma$ [j] * Product[C/.n  $\rightarrow$  n - i, {i, 0, j - 1}] *
Product[B/.n  $\rightarrow$  n - i, {i, j, J - 1}], {j, 1, J}];

```

```
(*Schritt 3.*)
```

```

If[(l = Exponent[r1, q^k, Min])  $\neq$  Exponent[r2, q^k, Min],
ldeg = Exponent[p, q^k, Min] -
Min[{Exponent[r1, q^k, Min], Exponent[r2, q^k, Min]}],
log =
Simplify[Log[q, Coefficient[r2, q^k, l]/
Coefficient[r1, q^k, l]],
Element[q, Reals]&&Positive[q]&&assumptions];
If[FreeQ[log, Log]&&Simplify[log  $\in$  Integers, assumptions],
ldeg = Simplify[Min[{log, Exponent[p, q^k, Min]}] - l,
assumptions];,
ldeg = Exponent[p, q^k, Min] - l;

```

```

];
];
If[Simplify[ldeg < 0, assumptions],
p = p * q^(-ldegk);
r2 = r2 * q^(-ldeg);
];
If[Simplify[Exponent[p, q^k, Min], assumptions] < 0, Continue[]];
If[(l = Exponent[r1, q^k, Max]) ≠ Exponent[r2, q^k, Max],
deg = Exponent[p, q^k, Max] -
Max[{Exponent[r1, q^k, Max], Exponent[r2, q^k, Max]}],
log =
Simplify[Log[q, Coefficient[r2, q^(k), l]/
Coefficient[r1, q^(k), l]],
Element[q, Reals]&&Positive[q]&&assumptions];
If[FreeQ[log, Log]&&Simplify[log ∈ Integers, assumptions],
deg = Simplify[Max[{log, Exponent[p, q^k, Max]}] - l,
assumptions]; ,
deg = Exponent[p, q^k, Max] - l;
];
];
If[Simplify[deg < 0, assumptions], Continue[]];

(*Schritt 4.*)
f = Sum[α[j] * q^(jk), {j, 0, deg}];
rec = Collect[(r1/(k → k + 1)) * f - r2 * (f/.k → k - 1) - p, q^k];
sol = Assuming[assumptions,
Solve[CoefficientList[rec, q^k] == 0,
Union[Table[α[j], {j, 0, deg}], Table[σ[j], {j, 1, J}]]]];
If[Not[sol==={}],

(*Schritt 5.*)
If[Not[qDispersionsMenge[(q^k - 1), p/.sol[[1]], q, k]==={}]&&
Not[(F + Sum[σ[j] * (F/.{n → n - j}), {j, 1, J}]/.{k → 0})===0],
Continue[]];
RE = S[n] + Sum[σ[j] * S[n - j]/.sol[[1]], {j, 1, J}];
RE = Numerator[Together[RE]];
RE = Collect[RE, S[___]];
RE = Map[Factor, RE];
Break[];
]
, {J, 1, 5}];
If[RE==={}, „Es gibt keine Rekursion der Ordnung 5“, RE == 0]
]

```



---

## Literatur

- [Koor1993] Koornwinder, T.H., On Zeilberger's algorithm and its q-analogue(1993), Journal of Computational and Applied Mathematics 48, 91-111, Netherlands
- [Gos1978] Gosper Jr, R. W., Decision procedure for indefinite hypergeometric summation, Proc. Natl. Acad. Sci. USA 75, 1978, 40-42.
- [Zeil1990] Zeilberger, D., A fast algorithm for proving terminating hypergeometric identities, Discrete Math. 80, 1990, 207-211.
- [Cauchy1893] Cauchy, A.-L., Mémoire sur les fonctions dont plusieurs valeurs sont liées entre elles par une équation linéaire, et sur diverses transformations de produits composés d'un nombre indéfini de facteurs (1843), C. R. Acad. Sci. Paris, T. XVII, p. 523, Oeuvres de Cauchy, 1<sup>re</sup> série, T. VIII, Gauthier-Villars, Paris, 1893, S. 42-50.
- [Heine1847] Heine, E., Untersuchungen über die Reihe ... (1847), J. reine angew. Math., 34, S. 285-328.
- [Jackson1904] Jackson, F.H., A generalization of the functions  $\Gamma(n)$  and  $x^n$ , 1904, Proc. Roy. Soc. London, 74, S. 64-72
- [Thomae1896] Thomae, J., Beiträge zur Theorie der durch die Heinesche Reihe ... , 1896, J. reine angew. Math., 70, S. 258-281
- [GasRahm2004] Gasper, G. and Rahman, M., Basic Hypergeometric Series, 2004, Encyclopedia of Mathematics and Its Applications, 35, Cambridge University Press, Cambridge and New York.



**Eigenständigkeitserklärung**

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbstständig verfasst habe.

Ich versichere, dass ich keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe, und dass die eingereichte Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist.

